

1С:Предприятие 8.3.6

Документация

Руководство администратора 83.003.10 от 17-03-2015

Руководство администратора Клиент-серверный вариант 83.005.09 от 12-02-2015

Интерфейс «Такси» 83.000.03 от 05-02-2015

Руководство пользователя 83.004.06 от 12-02-2015

Руководство разработчика 83.002.11 от 30.06.2015

1С:Предприятие 8.3.6

Руководство администратора

83.003.10 от 17-03-2015

Руководство администратора (02)

Введение (06)

Глава 1. Требования к аппаратуре и программному обеспечению (06)

Глава 2. Установка системы «1С_Предприятие» (15)

Глава 3. Установка конфигураций (04)

Глава 4. Запуск компонентов системы (12)

Глава 5. Ведение списка информационных баз (22)

Глава 6. Администрирование информационной базы (24)

Глава 7. Настройка веб-серверов для работы с «1С_Предприятием» (17)

Глава 8. Настройка веб-браузеров для работы в веб-клиенте (06)

Глава 9. Защита от несанкционированного использования настройка (26)

Глава 10. Обновление системы (01).pdf Глава 11. Удаление системы (01)

Глава 12. Администрирование мобильной платформы (03).pdf

Глава 13. Утилита контроля целостности (04)

Приложение 1. Структура каталога установки и назначение каталогов и файлов

Приложение 2. Описание элементов журнала регистрации (03)

Приложение 3. Описание и расположение служебных файлов (41)

Приложение 4. Утилита тестирования chdbfl (02)

Приложение 5. Обработка ошибок (02)

Приложение 6. Интернет-сервисы получения списка общих ИБ и дистрибутива

Приложение 7. Параметры командной строки запуска «1С_Предприятия» (16)

Приложение 8. Используемые компоненты и материалы (03)

Руководство администратора

ПРАВО ТИРАЖИРОВАНИЯ
ПРОГРАММНЫХ СРЕДСТВ И ДОКУМЕНТАЦИИ
ПРИНАДЛЕЖИТ ФИРМЕ «1С»

Приобретая систему «1С:Предприятие», вы тем самым даете согласие не допускать копирования программ и документации без письменного разрешения фирмы «1С».

© ООО «1С», 1996 – 2015

Фирма «1С», Москва, 123056, а/я 64

Отдел продаж: Селезневская ул., 21,

телефон: +7 (495) 737-92-57,

факс: +7 (495) 681-44-07,

e-mail: 1c@1c.ru

URL: <http://www.1c.ru>, <http://www.v8.1c.ru>, <http://users.v8.1c.ru>

Группа разработки программ – А. Алексеев, Р. Алейников, А. Безбородов, Д. Бескоровайнов, А. Бушнев, П. Василец, А. Виноградов, Я. Вирковский, А. Волков, И. Гольштейн, Е. Горностаев, И. Гусаров, Г. Дамье, А. Даровских, О. Дерут, Н. Евграфов, Б. Евтифеев, Д. Зарецкий, Д. Ивашов, И. Коваленко, С. Копиенко, Н. Корсаков, С. Кравченко, В. Кудрявцев, П. Кукушкин, А. Кунченко, А. Лакутин, М. Лейбович, Г. Леонтьев, А. Лехан, А. Макеев, А. Малышенок, А. Медведев, Е. Митрошкин, С. Мурзин, С. Нуралиев, Л. Онучин, М. Отставнов, Д. Павленко, А. Плякин, П. Романов, А. Рукин, Д. Русанов, М. Саблин, Е. Силин, Д. Службин, А. Смирнов, В. Соколов, П. Солодкий, А. Соляник, В. Сосновский, Д. Сысоенков, Д. Тишков, А. Топорков, В. Тунегов, А. Трубкин, А. Тюшкин, В. Филиппов, А. Хасанов, А. Цилябин, В. Ческис, В. Черемисинов, П. Чиков, А. Чичерин, А. Шевченко, А. Щербинин, В. Шульга.

Документация – В. Байдаков, В. Дранищев, Е. Королькова, А. Краюшкин, И. Кузнецов, М. Лавров, А. Моничев, А. Плякин, М. Радченко.

Группа консультационной поддержки – О. Акулова, С. Алексеева, О. Багрова, О. Баклушина, Э. Гарифуллина, В. Давыдова, О. Дмитренко, Л. Ермакова, М. Ершова, Ю. Жестков, О. Завальская, Н. Заявлиная, М. Звонилов, М. Иванова, Г. Коробка, Ю. Лаврова, С. Лепешкина, С. Мазурин, С. Марков, Ю. Мисан, А. Павликов, И. Панин, О. Пехтерева, С. Постнова, А. Прокуровский, Е. Романова, Г. Степаненко, Н. Степанов, Т. Токарева, Е. Широкова.

Группа тестирования – С. Баталин, А. Беляков, Б. Зиатдинов, А. Лапин, Е. Медведев, С. Потапкин.

Корректурa – Е. Семененко.

Верстка – О. Шестакова.

Наименование книги:	1С:Предприятие 8.3. Руководство администратора.
Номер издания:	83.003.10
Дата выхода:	17 марта 2015 года

ЛИНИЯ КОНСУЛЬТАЦИЙ

Для пользователей системы «1С:Предприятие 8» услуги линии консультаций предоставляются по линии информационно-технологического сопровождения (ИТС).

Получение консультаций возможно только после регистрации программного продукта (для чего необходимо заполнить регистрационную анкету и выслать ее в фирму «1С») и оформления подписки на ИТС.

В стоимость комплекта поставки программного продукта включено обслуживание по линии ИТС в течение периода, указанного в вашей регистрационной анкете. Купон на бесплатное обслуживание по линии ИТС, а также конверт с маркой и адресом входят в состав приобретенного вами комплекта программ.

По истечении периода бесплатного обслуживания получение консультаций возможно только по договору на ИТС. Для заключения договора обращайтесь к партнерам фирмы «1С».

Телефон и адрес электронной почты для консультаций указаны на диске ИТС.

Линия консультаций работает с 9:30 до 17:30 по московскому времени, кроме суббот, воскресений и праздничных дней (по расписанию московских предприятий).

Обращаясь к линии консультаций, следует находиться рядом со своим компьютером, иметь под рукой настоящее руководство и свою половину регистрационной анкеты. Желательно заранее уточнить типы используемого компьютера и принтера.

Набрав телефон линии консультаций, вы должны услышать ответ дежурного сотрудника. После этого сообщите наименование вашей организации, номер вашего экземпляра системы (он обозначен на CD-ROM комплекта поставки и на вашей половине регистрационной анкеты) и, возможно, другую регистрационную информацию по запросу сотрудника линии консультаций. Названная вами информация будет проверена по данным, указанным в отосланной в фирму «1С» половине регистрационной анкеты.

Отвечая на возникшие у вас вопросы, дежурный сотрудник может воспроизвести возникшую ситуацию на своем компьютере. Он даст вам рекомендации сразу же или после обсуждения с разработчиками. Вам не нужно просить к телефону конкретных специалистов: мы отвечаем за работу всего персонала. Работа линии консультаций регистрируется, поэтому при повторных обращениях по сложным проблемам вы можете сослаться на дату и время предыдущего разговора.

МЫ ВСЕГДА РАДЫ ВАМ ПОМОЧЬ!

Введение

Настоящая книга является руководством по администрированию системы «1С:Предприятие».

Структура Руководства

[Глава 1](#) содержит требования к аппаратуре и программному обеспечению для установки и работы «1С:Предприятия».

[Глава 2](#) посвящена описанию процесса установки системы программ «1С:Предприятие».

[Глава 3](#) содержит описание установки конфигураций.

[Глава 4](#) описывает запуск системы и настройку параметров окна запуска.

[Глава 5](#) содержит информацию по управлению списком информационных баз.

[Глава 6](#) посвящена возможностям администрирования в системе «1С:Предприятие».

[Глава 7](#) описывает настройку веб-серверов для работы с системой программ «1С:Предприятие».

[Глава 8](#) описывает настройку веб-браузеров для работы с веб-клиентом.

[Глава 9](#) посвящена описанию системы защиты от несанкционированного копирования и настройке ее параметров.

[Глава 10](#) описывает процесс обновления системы.

[Глава 11](#) описывает процесс удаления системы с компьютера.

[Глава 12](#) описывает администрирование мобильного приложения.

[Глава 13](#) описывает утилиту контроля целостности данных.

В приложениях содержится различная вспомогательная информация:

- Описание структуры каталогов, которая будет создана после установки, а также описание некоторых файлов и каталогов.
- Описание и расположение служебных файлов.
- Описание работы с утилитой тестирования и исправления информационных баз.
- Описание командной строки запуска клиентских приложений «1С:Предприятия» и Конфигуратора.
- Используемые компоненты и лицензии.

ВНИМАНИЕ! Информация, касающаяся администрирования клиент-серверного варианта работы системы «1С:Предприятие», приводится в книге «1С:Предприятие 8.3. Клиент-серверный вариант. Руководство администратора». Данная книга входит в состав поставки программных продуктов, содержащих серверную часть системы «1С:Предприятие».

Что вы должны знать

Характер изложения материала данного Руководства предполагает, что вы знакомы с операционной системой компьютера, на котором работает система «1С:Предприятие» (ОС семейства Microsoft Windows и ОС семейства Linux, подробнее перечень поддерживаемых ОС см. на сайте «1С:Предприятия»: <http://v8.1c.ru/requirements/>), и владеете базовыми навыками работы в ней.

Кроме того, необходимо обладать навыками администрирования используемой операционной системы.

При выполнении некоторых процедур администрирования могут потребоваться административные права доступа, а также дистрибутив используемой операционной системы.

Книги документации

В состав документации входят книги по технологической платформе «1С:Предприятие»:

- «1С:Предприятие 8.3. Руководство пользователя». Книга содержит описание общих приемов работы с программными продуктами, созданными на базе платформы «1С:Предприятие», и может не поставляться отдельно. Данная книга не поставляется в составе продукта «1С:Предприятие 8. Версия для обучения программированию».
- «1С:Предприятие 8.3. Руководство разработчика». Книга необходима для изменения и настройки конфигурации под особенности учета конкретной организации, а также для разработки новых конфигураций.
- «1С:Предприятие 8.3. Руководство администратора». Книга описывает администрирование системы «1С:Предприятие», включая информацию об особенностях построения клиент-серверных систем. Данная книга не поставляется в составе продукта «1С:Предприятие 8. Версия для обучения программированию».
- «1С:Предприятие 8.3. Клиент-серверный вариант. Руководство администратора». Книга описывает особенности работы системы «1С:Предприятие» с информационными базами в варианте клиент-сервер в части установки и эксплуатации. Данная книга не поставляется в составе продукта «1С:Предприятие 8. Версия для обучения программированию».
- Синтаксис встроенного языка и языка запросов представлен в книге «1С:Предприятие 8.3. Руководство разработчика». Описание объектной модели полностью включено в поставку в электронном виде (в разделах справки конфигуратора и синтакс-помощнике). Описание объектной модели также содержится в книге «1С:Предприятие 8.3. Описание встроенного языка», которая распространяется отдельно.

ВНИМАНИЕ! Комплект поставки конкретного продукта может включать лишь некоторые из перечисленных книг документации.

Методические материалы и дополнительные возможности

Фирма «1С» осуществляет методическую поддержку освоения и внедрения системы программ «1С:Предприятие». Методическая поддержка включает в себя разнообразные формы предоставления информации, необходимой для грамотной и эффективной разработки и использования прикладных решений.

Сопроводительные файлы комплекта поставки «1С:Предприятия»

В процессе установки платформы «1С:Предприятие» выполняется копирование на жесткий диск ряда сопроводительных файлов, содержащих описание изменений, реализованных в данной версии платформы, и инструкции по переходу с предыдущих версий.

Все сопроводительные файлы располагаются в каталоге установочных файлов конкретной версии системы «1С:Предприятие», в каталоге [\docs\ru](#).

- [V8Update.htm](#) – в этом файле содержатся отличия текущей версии платформы от предыдущих версий и особенности перехода на новую версию.

ИТС – информационно-технологическое сопровождение

Фирма «1С» осуществляет платную методическую поддержку пользователей в рамках информационно-технологического сопровождения (ИТС) программ системы «1С:Предприятие».

Ежемесячные выпуски ИТС содержат большое количество постоянно обновляемой информации, позволяющей более эффективно использовать продукты системы «1С:Предприятие». Отметим наиболее важные составляющие ИТС:

- обновления технологической платформы «1С:Предприятие» и прикладных решений;
- методические материалы по технологической платформе «1С:Предприятие»;
- методические материалы по типовым прикладным решениям фирмы «1С»;
- конфигурация «Конвертация данных» для настройки правил обмена между информационными базами «1С:Предприятия», имеющими различную конфигурацию;
- система стандартов и методик разработки конфигураций для платформы «1С:Предприятие» (предназначена для ознакомления партнеров и пользователей фирмы «1С» с техническими и проектными решениями, используемыми при разработке типовых конфигураций на платформе «1С:Предприятие»);
- советы линии консультаций, помогающие пользователям «1С:Предприятия» получить ответы на наиболее часто встречающиеся вопросы и избежать типичных ошибок;
- информация по обучению работе с платформой «1С:Предприятие» и прикладными решениями фирмы «1С»;
- демонстрационные ролики программных продуктов, позволяющие получить первое представление о возможностях прикладных решений;
- справочники по заполнению деклараций по налогам (налог на прибыль, НДС, налог на имущество, ЕСН, взносы в ПФР);
- обширная подборка бухгалтерской периодики, включая текущие выпуски журналов и архивы;
- база аналитических обзоров законодательства и арбитражной практики;
- рекомендации по составлению квартальной и годовой отчетности в «1С:Бухгалтерии 8»;
- справочник типовых хозяйственных операций хозрасчетного предприятия;
- справочник по оформлению расчетов с персоналом по оплате труда и правовым аспектам трудовых отношений;
- справочная правовая система «Гарант» – полный набор нормативных документов законодательства Российской Федерации, в том числе по бухгалтерскому учету, налогам и предпринимательству.

Начиная с 2005 года, стандартный сервис по поддержке пользователей – подписчиков ИТС включает доступ к сайту поддержки пользователей системы «1С:Предприятие».

Более подробно о проекте ИТС можно прочитать на сайте фирмы «1С»:

<http://www.1c.ru/rus/support/its/its.htm>.

Оставить заявку на демонстрацию дисков и приобретение подписки на ИТС можно на сайте фирмы «1С»:

<http://www.1c.ru/rus/support/its/zajavka.jsp>.

Оформить подписку на ИТС можно у партнеров фирмы «1С». Список партнеров, имеющих опыт массового обслуживания пользователей в рамках проектов ИТС, опубликован на сайте фирмы «1С»:

<http://www.1c.ru/rus/partners/service.jsp>.

Информация по «1С:Предприятию 8»

Адрес сайта: <http://v8.1c.ru/AllInfo>.

Данный сайт представляет собой набор ссылок на часто используемую информацию для следующих категорий пользователей:

- пользователи прикладных решений,

- разработчики прикладных решений,
- партнеры фирмы «1С».

Сайт системы программ «1С:Предприятие 8»

Адрес сайта: <http://v8.1c.ru>.

Сайт содержит информацию по технологической платформе системы «1С:Предприятие» и по типовым прикладным решениям, выпущенным фирмой «1С» на ее основе.

Пользовательский сайт

Адрес сайта: <http://users.v8.1c.ru>.

На сайте поддержки пользователей системы «1С:Предприятие» представлена информация о номерах версий платформы и конфигураций, дате их выхода, выпусках ИТС, на которых опубликовано обновление. По каждой версии представлена следующая информация:

- для технологической платформы:
 - отличия данной версии от предыдущих версий и особенности перехода;
 - ошибки, исправленные при выпуске данной версии;
 - дистрибутив обновления;
 - файл [readme.htm](#);
- для прикладных решений:
 - новое в релизе;
 - полный список изменений;
 - список основных изменений;
 - порядок обновления;
 - дистрибутив обновления;
 - номер версии платформы, необходимой для использования релиза конфигурации.

Также на пользовательском сайте публикуются рекомендации по администрированию системы «1С:Предприятие».

Основное преимущество, которое дает пользователям «1С:Предприятия» поддержка на данном сайте, – это возможность обновления технологической платформы и прикладных решений через Интернет до получения дисков ИТС.

Также на сайте публикуются дополнительные компоненты, используемые системой «1С:Предприятие» (например, СУБД PostgreSQL), а также тестовые версии платформы и прикладных решений.

ВНИМАНИЕ! К сайту поддержки пользователей имеют право доступа зарегистрированные пользователи программных продуктов системы «1С:Предприятие». При этом пользователи продуктов, для которых предусмотрено обслуживание по линии информационно-технологического сопровождения (ИТС), должны иметь действующую подписку на ИТС.

Зарегистрироваться на сайте поддержки пользователей можно самостоятельно либо обратиться к партнеру фирмы «1С».

Для самостоятельной регистрации необходимо с главной страницы сайта перейти по ссылке [Самостоятельная регистрация пользователей по PIN-коду](#) и выполнить указанные на сайте действия.

В процессе регистрации потребуется PIN-код, входящий в комплект поставки программного продукта.

Если в комплекте поставки отсутствует PIN-код для регистрации на сайте, следует обратиться к партнеру фирмы «1С».

Для регистрации программного продукта необходимо заполнить регистрационную анкету на программный продукт (она является частью регистрационной карточки) и отправить ее в фирму «1С» по почте или факсу.

Для оформления подписки на ИТС можно обратиться к любому из сервис-партнеров фирмы «1С». Список партнеров, имеющих опыт массового обслуживания пользователей в рамках проектов ИТС, опубликован на сайте фирмы «1С»: <http://www.1c.ru/rus/partners/service.jsp>.

Принятые обозначения

Для лучшего понимания излагаемого материала в настоящем Руководстве приняты некоторые общие приемы выделения отдельных элементов текста. Соглашение о таких приемах приведено ниже.

Обозначения клавиш. Клавиши, такие как [Enter](#), [Esc](#), [Del](#) и подобные, будут обозначаться, как показано выше, без кавычек.

Для ссылок на клавиши управления курсором (клавиши со стрелками) будет использоваться фраза [клавиши управления курсором](#), когда необходимо сослаться сразу на все эти клавиши. Если необходимо упомянуть эти клавиши по отдельности, будут использоваться выражения [Стрелка вверх](#), [Стрелка вниз](#), [Стрелка вправо](#) и [Стрелка влево](#).

Комбинации клавиш. Когда для выполнения какой-либо команды необходимо нажать комбинацию из двух клавиш, она дается в виде [Ctrl + F3](#).

Обозначения кнопок. Наименования кнопок в форме будут даваться их названиями без кавычек, например, [OK](#), [Отмена](#), [Удалить](#) и так далее.

Ключевые слова встроенного языка. Ключевые слова встроенного языка системы «1С:Предприятие» выделяются шрифтом и пишутся так, как в программных модулях: [РабочаяДата](#). В тексте также будут встречаться ссылки на описания разделов или элементов встроенного языка (свойства, методы и т. д.). С данными описаниями можно ознакомиться в справке (ветвь [Встроенный язык](#)).

Описание действия с помощью меню. Для описания выбора пункта меню используется следующая конструкция: [Меню – Подменю – Подменю – ... – Пункт](#). Например: «Для выбора масштаба изображения используется пункт [Таблица – Вид – Масштаб](#), что эквивалентно тексту: «Для выбора масштаба изображения используется пункт [Масштаб](#) подменю [Вид](#) меню [Таблица](#) главного меню программы». Если выбор осуществляется не из главного меню программы, то это указывается дополнительно.

Режимы работы системы «1С:Предприятие». Система «1С:Предприятие» работает в двух режимах: настройки и проверки конфигурации (далее в Руководстве – режим Конфигуратор или конфигурирование), когда в Руководстве описывается работа по созданию или изменению конфигурации) и исполнения конфигурации (далее в Руководстве – режим 1С:Предприятие).

В данном Руководстве пользователем будет называться специалист, выполняющий разработку или сопровождение конфигурации.

Выражение %USERPROFILE% означает переменную окружения ОС Windows, которая содержит путь к каталогу профиля текущего пользователя. В случае стандартной установки (и для пользователя [Ivanov](#)) этот путь выглядит следующим образом:

C:\Documents and Settings\Ivanov

[Копировать в буфер обмена](#)

Для ОС Windows Vista и выше, этот путь выглядит следующим образом:

C:\Users\Ivanov

Выражение %APPDATA% означает переменную окружения ОС Windows, которая содержит путь к каталогу (в профиле пользователя), где приложения хранят свои данные. В случае стандартной установки (и для пользователя **Ivanov**) этот путь выглядит следующим образом:

[Копировать в буфер обмена](#)

C:\Documents and Settings\Ivanov\Application Data

Для ОС Windows Vista и выше, этот путь выглядит следующим образом:

[Копировать в буфер обмена](#)

C:\Users\Ivanov\AppData\Roaming

Выражение %LOCALAPPDATA% означает переменную окружения ОС Windows Vista и старше, содержащую путь к каталогу (в профиле пользователя), в котором находятся данные приложения, специфичные для пользователя. В случае стандартной установки (и для пользователя **Ivanov**) этот путь выглядит следующим образом:

[Копировать в буфер обмена](#)

C:\Users\Ivanov\AppData\Local

Выражение %ALLUSERSPROFILE% означает переменную окружения ОС Windows, которая содержит путь к каталогу, доступному всем пользователям системы. В случае стандартной установки этот путь выглядит следующим образом:

[Копировать в буфер обмена](#)

C:\Documents and Settings\All Users

Для ОС Windows Vista и выше, этот путь выглядит следующим образом:

[Копировать в буфер обмена](#)

C:\ProgramData

Исполняемые файлы, разработанные одновременно для ОС Windows и для ОС Linux, обозначаются одинаково, вне зависимости от используемой ОС, и без указания расширения (если таковое присутствует). Так, исполняемый файл **1cv8.exe** будет упоминаться как **1cv8**. При этом для ОС Windows необходимо добавить расширение **.exe** (регистр файл не имеет значения), а для ОС Linux ничего добавлять не надо (регистр файлов имеет значение).

Глава 1. Требования к аппаратуре и программному обеспечению

Актуальные системные требования системы «1С:Предприятие» с текущими уточнениями опубликованы на сайте: <http://www.v8.1c.ru/requirements/>.

1.1. Тонкий клиент

Компьютер конечного пользователя:

- операционная система Windows XP Service Pack 2, Windows Server 2003, Windows Vista, Windows 7, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Fedora 17, Mint 12, Ubuntu 12.04 LTS, Alt Linux СРТ 6.0;
- процессор Intel Pentium/Celeron 1800 МГц и выше;
- оперативная память 1 Гбайт и выше;
- жесткий диск (при установке используется около 70 Мбайт);
- устройство чтения компакт-дисков;
- USB-порт;
- SVGA-дисплей.

1.2. Толстый клиент

Компьютер конечного пользователя:

- операционная система Windows XP Service Pack 2, Windows Server 2003, Windows Vista, Windows 7, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Fedora 17, Mint 12, Ubuntu 12.04 LTS, Alt Linux СРТ 6.0;
- процессор Intel Pentium/Celeron 1800 МГц и выше;
- оперативная память 1 Гбайт и выше;
- жесткий диск (при установке используется около 300 Мбайт);
- устройство чтения компакт-дисков;
- USB-порт;
- SVGA-дисплей.

Компьютер, используемый для разработки конфигураций:

- операционная система Windows XP Service Pack 2, Windows Server 2003, Windows Vista, Windows 7, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Fedora 17, Mint 12, Ubuntu 12.04 LTS;
- процессор Intel Pentium/Celeron 2400 МГц и выше;
- оперативная память 2 Гбайт и выше (рекомендуется 4 Гбайт);
- жесткий диск (при установке используется около 300 Мбайт);
- устройство чтения компакт-дисков;

- USB-порт;
- SVGA-дисплей.

На требования, предъявляемые к оперативной памяти при работе толстого клиента и разработке конфигураций, оказывает влияние функциональная наполненность конфигурации.

1.3. Веб-клиент

Компьютер конечного пользователя:

- Требования для работы веб-клиента в основном определяются используемым веб-браузером. Особенности работы веб-клиента описаны в книге «1С:Предприятие 8.3. Руководство разработчика».

Список поддерживаемых веб-браузеров:

- для ОС Windows:
 - Mozilla Firefox 17 и выше;
 - Microsoft Internet Explorer 8.0 – 11.0;
 - Google Chrome 4 и выше.
- для ОС Linux:
 - Mozilla Firefox 17 и выше.
- для ОС MacOS X:
 - Safari 4.0.5 и выше (для MacOS X версии 10.5 и выше);
- процессор Intel Pentium/Celeron 1800 МГц и выше;
- оперативная память 1 Гбайт и выше;
- жесткий диск (при установке используется около 250 Мбайт);
- SVGA-дисплей.

СОВЕТ. Для компьютеров с небольшим объемом памяти и не производительным процессором, для браузера Microsoft Internet Explorer рекомендуется использовать версию 9.0 или использовать другие браузеры, поддерживаемые 1С:Предприятием.

1.4. Мобильная платформа

Для устройств на базе ОС iOS:

- iOS – версия 5.1 и старше;
- iPhone – 3GS и старше;
- iPod Touch – 3 поколения и старше;
- iPad – все версии;
- iPad mini.

Для устройств на базе ОС Android:

- Android – версия 2.3 и старше;
- Процессор:
 - с архитектурой ARMv5TE и выше (ARMv6, ARMv7, ARMv8);
 - с архитектурой Intel x86.
- Оперативная память – не менее 256Мб;
- Сенсорный экран.

Рекомендуется выполнять обновление до последней актуальной версии операционной системы на используемом мобильном устройстве.

1.5. Использование энергосберегающих режимов работы компьютера

Если на компьютере запущено «1С:Предприятие», то использование энергосберегающих режимов будет возможно только при соблюдении всех нижеперечисленных условий:

- ключ защиты установлен локально;
- используется файл-серверный вариант;
- файл базы данных расположен на локальном диске.

В противном случае переход в энергосберегающий режим будет запрещен независимо от вызвавшей его причины.

1.6. Поддерживаемые веб-серверы

Система «1С:Предприятие» поддерживает работу со следующими веб-серверами:

- **Microsoft Internet Information Services (IIS) версий 5.1, 6.0, 7.0, 7.5, 8.0, 8.5.**

Документация на веб-сервер:

- версия IIS 6.0:
<http://technet.microsoft.com/ru-ru/library/cc785089.aspx>;
- версия IIS 7.0:
<http://technet.microsoft.com/ru-ru/library/cc732976.aspx>;
- версия IIS 7.5:
<http://technet.microsoft.com/ru-ru/library/cc754281.aspx>;
- версия IIS 8.0, 8.5:
<http://technet.microsoft.com/ru-ru/library/hh831725.aspx>.

- **Apache HTTP Server версий 2.0, 2.2.**

Актуальные версии веб-сервера можно получить по адресу: <http://httpd.apache.org/download.cgi>.
Документация на веб-сервер:

- версия Apache 2.0:
<http://httpd.apache.org/docs/2.0/> (на английском языке);
- версия Apache 2.2:
<http://httpd.apache.org/docs/2.2/> (на английском языке).

ПРИМЕЧАНИЕ. Текущий список поддерживаемых веб-серверов опубликован на сайте <http://www.v8.1c.ru/requirements/>.

Для использования внешних источников данных в случае файлового варианта информационной базы, работа с которым выполняется через веб-сервер, работающий под управлением ОС Linux, на компьютере с веб-сервером необходима библиотека [unixOdbc](#) версии 2.2.11 и выше.

Если веб-сервер используется для доступа к файловому варианту информационной базы, то к компьютеру, на котором работает веб-сервер и расширение веб-сервера, предъявляются следующие требования:

- процессор Intel Pentium/Celeron 1800 МГц и выше;
- оперативная память 1 Гбайт и выше (рекомендуется 2 Гбайт);
- жесткий диск (при установке используется около 300 Мбайт);
- устройство чтения компакт-дисков;
- USB-порт;
- SVGA-дисплей.

1.7. Прочие требования

1.7.1. Для ОС Windows

На клиентском компьютере должен быть установлен Microsoft Internet Explorer версии 7.0 и выше.

Пользователь, от чьего имени выполняется клиентское приложение под управлением ОС Windows, должен иметь право **Список содержимого папки** на каталог временных файлов.

Для функционирования системы программного лицензирования (см. [здесь](#)) необходимо, чтобы на компьютере была запущена служба WMI (Windows Management Instrumentation, <http://msdn.microsoft.com/en-us/library/aa394582.aspx>).

1.7.2. Для ОС Linux

Для работы под управлением ОС Linux, необходимо наличие установленных библиотек:

- [webkitgtk](#) версии 1.4.3 и выше.

Для использования некоторых возможностей системы, может потребоваться наличие следующих библиотек:

- **ImageMagick:**
 - Имя загружаемой библиотеки: [libMagickWand](#), [libWand](#) или [libMagickWand-6.Q16](#).
 - Используемая версия:
 - версии [6.2.8](#) и выше, если не планируется работать с географической схемой;
 - версии [6.3.2](#) и выше, если планируется работать с географической схемой.
 - Назначение:
 - Для работы сервера «1С:Предприятия» в управляемом режиме;
 - При использовании на сервере объектов [Диаграмма](#), [ГрафическаяСхема](#), [ТабличныйДокумент](#);

- При использовании на сервере метода `ПолучитьКартинку()` объектов `Диаграмма`, `ДиаграммаГанта`, `Дендрограмма`, `СводнаяДиаграмма`.

- **FreeType:**

- Имя загружаемой библиотеки: `libfreetype`.
- Используемая версия: 2.1.9 и выше.
- Назначение:
 - Для работы сервера «1С:Предприятия» в управляемом режиме;
 - При использовании на сервере объектов `Диаграмма`, `ГрафическаяСхема`, `ТабличныйДокумент`;
 - при использовании метода `ПолучитьКартинку()` объектов `Диаграмма`, `ДиаграммаГанта`, `Дендрограмма`, `СводнаяДиаграмма`;
 - При выполнении операции сохранения в PDF.

- **Libgsf;**

- Имя загружаемой библиотеки: `libgsf-1`.
- Используемая версия: 1.10.1 и выше.
- Назначение: экспорт/импорт документов в формате XLS.

- **Glib;**

- Имя загружаемой библиотеки: `libglib-2.0`.
- Используемая версия: 2.12.4 и выше.
- Назначение: экспорт/импорт документов в формате XLS.

- **unixOdbc:**

- Имя загружаемой библиотеки: `libodbc`.
- Используемая версия: 2.2.11 и выше.
- Назначение: работа с внешними источниками данных.

- **Kerberos:**

- Имя загружаемой библиотеки: `libkrb5`.
- Используемая версия: 1.4.2 и выше.
- Назначение: Аутентификация средствами ОС.

- **GSS-API Kerberos:**

- Имя загружаемой библиотеки: `libgssapi_krb5`.
- Используемая версия: 1.4.2 и выше.
- Назначение: Аутентификация средствами ОС.

- **Microsoft Core Fonts.**

«1С:Предприятие» загружает библиотеку, указывая имя в виде [имя загружаемой библиотеки.so.X.Y](#), где:

- [имя загружаемой библиотеки](#) – значение из вышеприведенного списка;
- [so](#) – признак файл с библиотекой;
- [X.Y](#) – цифры суффикса присутствующей в системе библиотеки.

Используются только библиотеки, зарегистрированные в системном кеше динамического линковщика времени выполнения (данную информацию можно получить, запустив команду `ldconfig -p`). Если в системе имеется несколько версий одной библиотеки, будет загружена самая старшая версия.

1.8. Ограничения

Совместная работа с файловой информационной базой возможна только клиентскими приложениями одной версии. Архитектура (32– или 64-разрядное приложение) и используемая ОС (**Windows** или **Linux**) клиентского приложения, выполняющего доступ к файловой информационной базе, не ограничивается.

Максимальное количество одновременных соединений с одной файловой информационной базы равно 1024.

Совместная работа с файловой информационной базой по сети поддерживается только для сетевых ресурсов, доступ к которым осуществляется по протоколу **SMB (CIFS)**. Такие ресурсы могут располагаться как на компьютерах с ОС **Windows**, так и на компьютерах с ОС **Linux**.

Глава 2. Установка системы «1С:Предприятие»

Система «1С:Предприятие» представляет собой совокупность программных модулей, предназначенных для разработки и использования решений (конфигураций) по ведению учета и автоматизации хозяйственной деятельности предприятий, и конфигурации или совокупности конфигураций.

Программные модули системы «1С:Предприятие» универсальны и способны работать с любой конфигурацией (в рамках имеющегося Лицензионного соглашения).

Одновременно с установкой «1С:Предприятия» также производится установка драйвера защиты от несанкционированного использования.

Программа установки позволяет устанавливать на один компьютер несколько версий «1С:Предприятия», выбирать устанавливаемые компоненты системы, выбирать вариант установки сервера «1С:Предприятия».

Программа запуска, входящая в состав системы «1С:Предприятие», позволяет работать с единым списком информационных баз всех версий системы программ «1С:Предприятие» (версий 8.0, 8.1, 8.2 и 8.3).

2.1. Общие сведения о процессе установки

Установка системы «1С:Предприятие» существенно отличается для операционных систем семейства Microsoft Windows (далее ОС Windows) и для операционных систем семейства Linux (далее ОС Linux).

Для ОС Windows установка выполняется с помощью специальной программы установки (см. [здесь](#)).

Для ОС Linux такая программа отсутствует, поэтому в соответствующих разделах будут описаны конкретные действия для выполнения той или иной операции (см. [здесь](#)).

Перед тем как приступить к установке, следует убедиться, что компьютер не заражен вирусами, а жесткий диск не содержит ошибок и имеется достаточно свободного места для выполнения установки.

ПРИМЕЧАНИЕ. Для установки могут понадобиться дистрибутивные носители операционной системы, установленной на компьютере, а также **права** локального или сетевого **администратора**.

2.2. Программа установки для ОС Windows

2.2.1. Доступные программы установки

Доступны следующие программы установки:

- **1С:Предприятие 8** – позволяет выполнить установку любого компонента системы, кроме 64-разрядного сервера «1С:Предприятия».
- **1С:Предприятие 8 Тонкий клиент** – позволяет выполнить установку только компонентов, необходимых для доступа к серверу «1С:Предприятия» и собственно тонкий клиент.
- **1С:Предприятие 8 (x86-64)** – позволяет выполнить установку 64-разрядного сервера «1С:Предприятия».

Принципы работы с каждой программой установки одинаковы, поэтому будут представлены общие сведения для программы установки **1С:Предприятие 8**.

2.2.2. Общие сведения о программе установки

Установку выполняет специальный помощник, переход по страницам которого выполняется с помощью кнопки **Далее >>**. Для старта помощника следует запустить программу **setup.exe** из каталога выбранного дистрибутива. На каждой странице нужно указать какую-либо необходимую информацию, которая в дальнейшем будет использована для установки системы «1С:Предприятие».

Установка будет выполнена в следующих случаях:

- Пользователь, который запускает программу установки, входит в группу локальных администраторов.
- Пользователь, который запускает программу установки, не входит в группу локальных администраторов, но для пользователя и компьютера разрешена установка программ (ключ реестра **AlwaysInstallElevated**).

Далее будет приведено краткое описание каждого шага помощника.

ПРИМЕЧАНИЕ. Если запустить программу **setup.exe** с использованием ключа **/S**, то установка будет выполнена в «бесшумном» режиме, по настройкам из файла **1cestart.cfg** (см. [здесь](#)), а если файл не существует, будет выполнена установка по умолчанию.

При запуске программы установки можно указать параметр **USEHWLICENSES**, с помощью которого можно в момент установки указать необходимость поиска ключа защиты при запуске клиентского приложения. Например, приведенная ниже командная строка выполняет запись параметра **UseHwLicenses=0** в файл **1cestart.cfg** (см. [здесь](#)) пользователя, который выполняет установку «1С:Предприятия»:

```
setup.exe USEHWLICENSES=0
```

[Копировать в буфер обмена](#)

ПРИМЕЧАНИЕ. Параметр **USEHWLICENSES** также можно указать в файле **setup.ini**, в ключе **CmdLine** группы **Startup**.

2.2.2.1. Приветствие

С этого окна начинается работа помощника установки системы «1С:Предприятие».

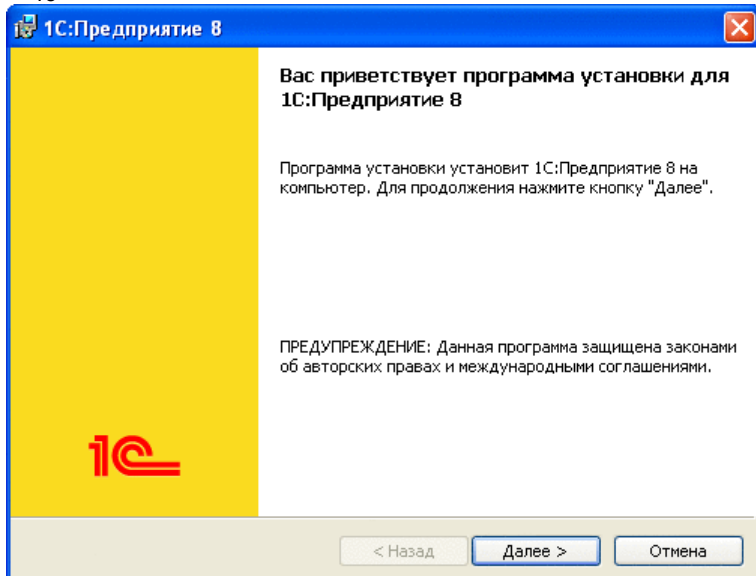


Рис. 1. Приветствие

2.2.2.2. Выбор компонентов

На этой странице необходимо выбрать те компоненты, которые будут устанавливаться, и каталог для установки. Перечень компонентов зависит от того, что необходимо установить. Некоторые типовые сценарии установки будут рассмотрены ниже (см. [здесь](#)).

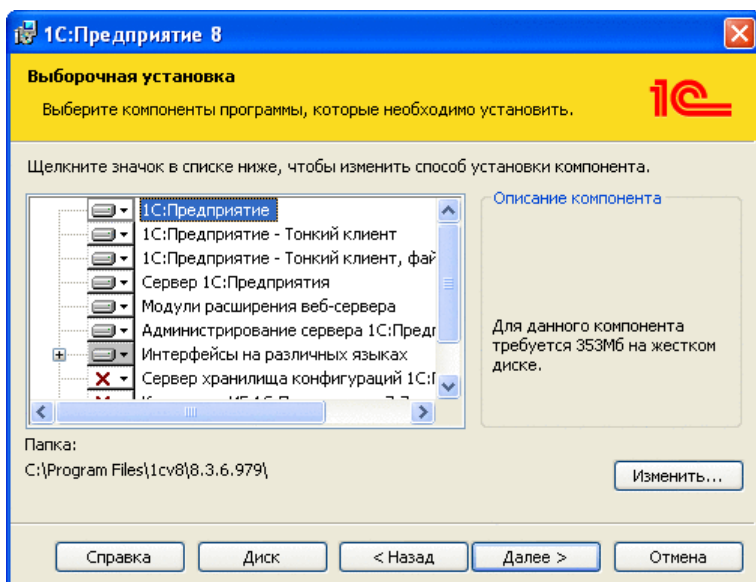


Рис. 2. Выбор компонентов

Если необходимо установить какой-либо компонент, нужно отметить его. Если какой-либо компонент устанавливать нет необходимости, следует запретить его. Для выбора компонента необходимо нажать мышью пиктограмму слева от имени компонента (или клавишу [Пробел](#)). В появившемся меню выбрать необходимый пункт (см. [рис. 3](#)).

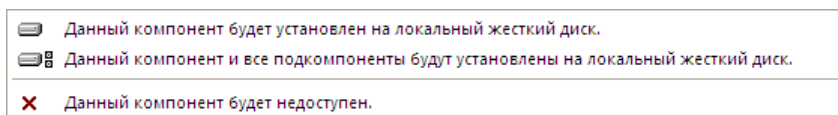


Рис. 3. Меню установки компонента

Устанавливаемые и запрещенные компоненты выглядят на экране так, как показано на [рис. 4](#).

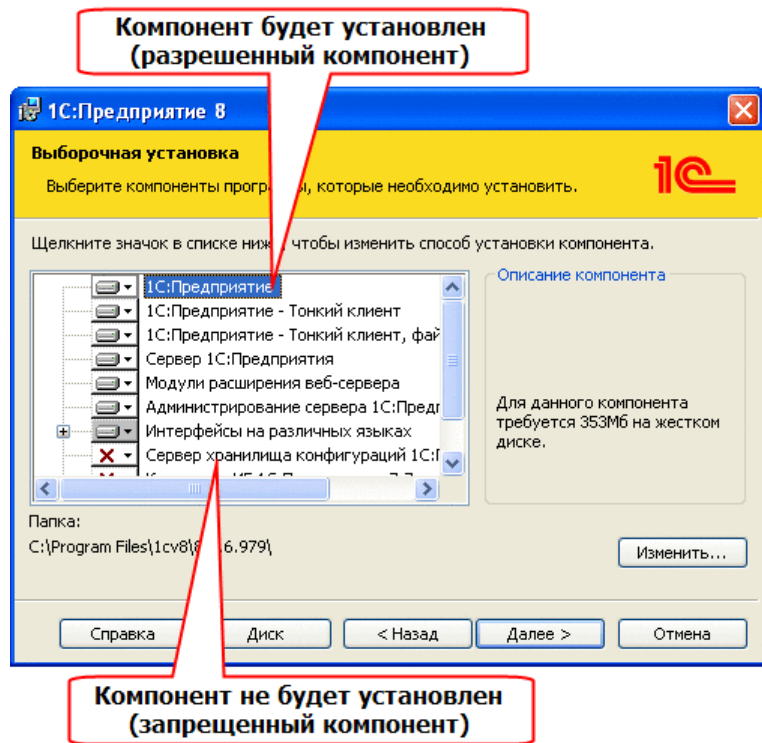


Рис. 4. Разрешенные и запрещенные компоненты

Далее приводится краткое описание устанавливаемых компонентов.

Компонент	Краткое описание
1С:Предприятие	Основные компоненты «1С:Предприятия», включая компоненты для администрирования, конфигурирования, толстый и тонкий клиент
1С:Предприятие – тонкий клиент	Компоненты тонкого клиента только для работы в клиент-серверном варианте
1С:Предприятие – тонкий клиент, файловый вариант	Компоненты тонкого клиента, включая компоненты для работы с файловым вариантом информационной базы
Сервер 1С:Предприятия 8	Компоненты сервера «1С:Предприятия», включая сервер администрирования и утилиту администрирования
Модули расширения веб-сервера	Модули расширения веб-серверов, необходимые для работы веб-клиента и Web-сервисов
Администрирование сервера 1С:Предприятия 8	Дополнительные компоненты для администрирования кластера серверов «1С:Предприятия»
Интерфейсы на разных языках	Пользовательские интерфейсы на различных языках
Сервер хранилища конфигураций 1С:Предприятия 8	Компоненты сервера хранилища конфигураций «1С:Предприятия»
Конвертор ИБ 1С:Предприятия 7.7	Конвертер информационных баз «1С:Предприятия 7.7»
Контроль целостности	Утилита контроля целостности данных (см. здесь)

Независимо от того, в какой каталог будет устанавливаться «1С:Предприятие» (поле **Папка:** и кнопка **Изменить**), некоторые каталоги установленной системы будут располагаться в фиксированных местах. Подробнее об этом см. [здесь](#).

После успешного завершения процесса установки будет сформирован локальный конфигурационный файл (подробнее см. [здесь](#)) для всех пользователей, в котором будут установлены два параметра: **InstalledLocation** и **InstallComponents**. Значения этих параметров будут установлены в соответствии с настройками, выполненными в процессе установки системы.

Утилита контроля целостности (см. [здесь](#)) устанавливается (по умолчанию) в тот же каталог, что и «1С:Предприятие». При необходимости можно изменить каталог установки утилиты. Для этого следует выделить компонент **Контроль целостности** и затем воспользоваться кнопкой **Изменить...** для смены каталога установки.

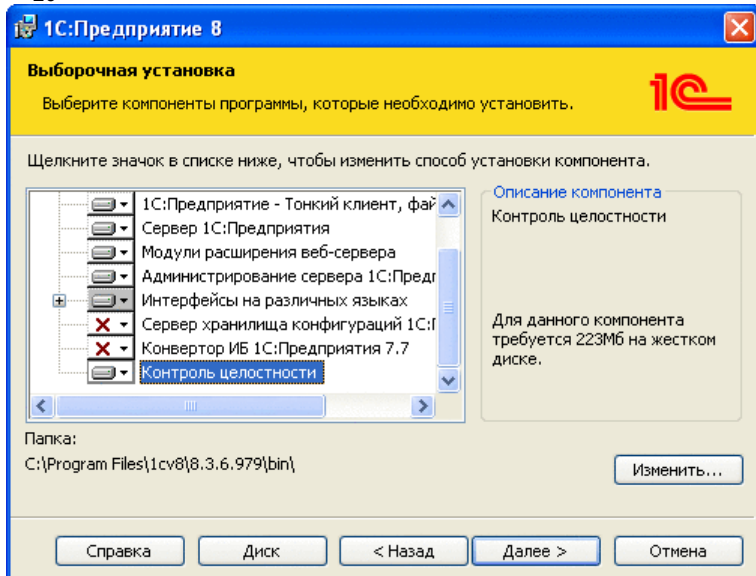


Рис. 5. Установка утилиты контроля целостности

Возможность смены каталога установки для утилиты контроля целостности недоступны, если компонент запрещен.

Факт установки утилиты контроля целостности не запоминается в локальном конфигурационном файле. При установке каждой версии «1С:Предприятия» необходимость установки этой утилиты указывается вручную.

2.2.2.3. Выбор языка интерфейса по умолчанию

На следующем шаге программа установки предложит выбрать язык интерфейса по умолчанию.

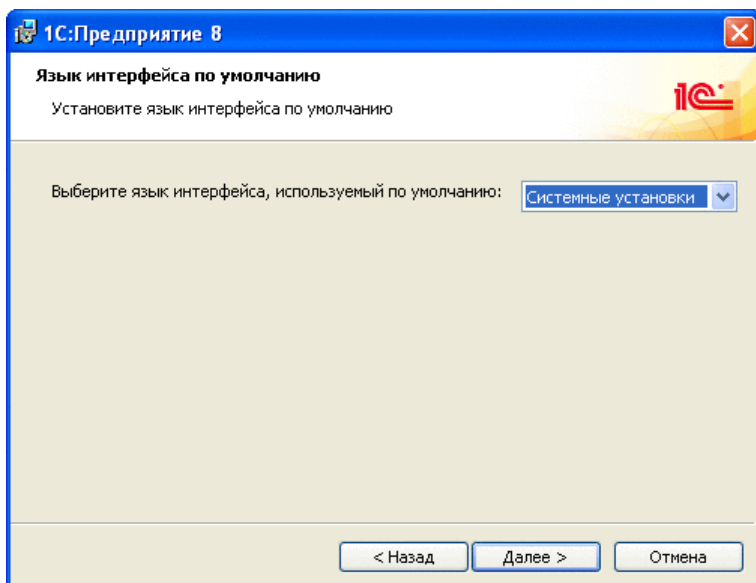


Рис. 6. Выбор языка интерфейса

Следует указать один из языков интерфейса в качестве языка интерфейса по умолчанию.

После завершения работы программы установки в каталоге `C:\Program Files\1Cv8\conf` будет создан файл `conf.cfg` (см. [здесь](#)), описывающий язык интерфейса по умолчанию.

Если в процессе работы необходимо использование языка интерфейса, отличного от установленного по умолчанию, нужно указать его с помощью ключа командной строки `/L`.

Язык интерфейса	Код языка
Азербайджанский	az
Английский	en
Болгарский	bg
Вьетнамский	vi
Грузинский	kk
Казахский	ka
Китайский	zh
Латышский	lv

Литовский	Lt
Немецкий	de
Польский	pl
Румынский	ro
Русский	ru
Турецкий	tr
Украинский	uk

2.2.2.4. Установка сервера «1С:Предприятия»

Если для установки выбран компонент [Сервер 1С:Предприятия 8](#), то будет доступна страница помощника, на которой следует выбрать режим установки сервера «1С:Предприятия» и пользователя, под которым будет работать сервер в случае его установки в режиме сервиса ОС Windows.

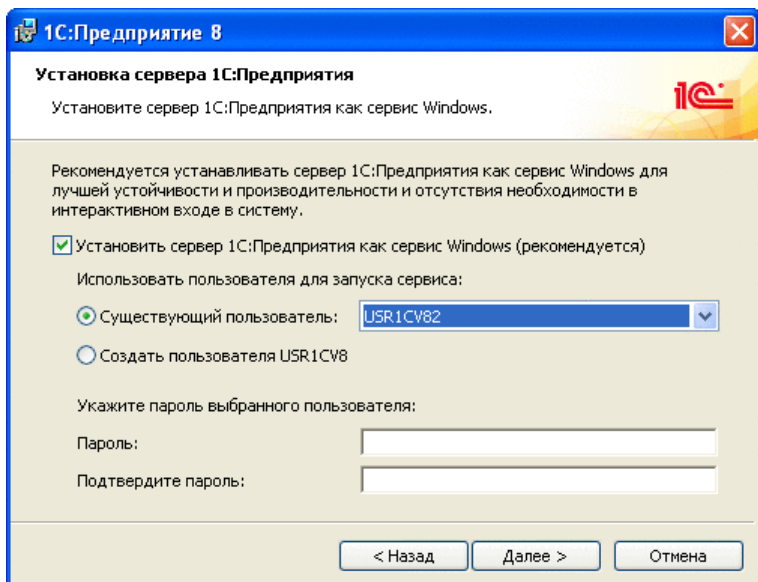


Рис. 7. Режим установки сервера «1С:Предприятия»

ПРИМЕЧАНИЕ. Если выбран режим установки сервера как сервиса, то необходимо указать пароль для выбранного пользователя, иначе программа установки не сможет запустить сервер.

Если на компьютере существует установленная версия «1С:Предприятия» с сервером, установленным в качестве сервиса ОС Windows, то программа установки переустановит сервис.

2.2.2.5. Начало установки

После нажатия кнопки [Установить](#) начнется процесс установки:

- создаются необходимые каталоги;
- копируются файлы выбранных компонентов;
- создаются конфигурационные файлы;
- выполняется регистрация программных компонентов системы;
- на рабочем столе создается ярлык для запуска «1С:Предприятия»;
- выполняется запуск сервера «1С:Предприятия», если выбрана установка сервера как сервиса ОС Windows.

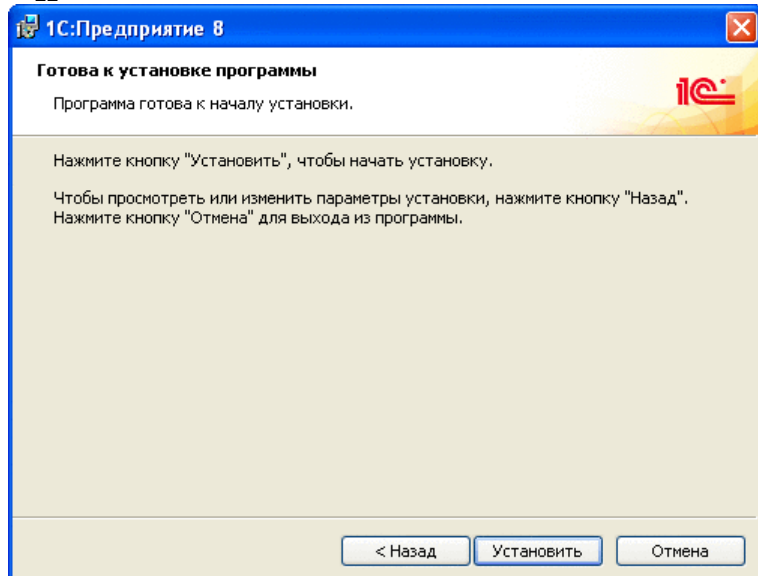


Рис. 8. Начать установку

При этом для каждой версии в компоненте [Установка и удаление программ](#) панели управления Windows будет своя запись вида [1С:Предприятие 8 \(8.3.3.100\)](#).

2.2.2.6. Установка драйвера защиты

После окончания процесса установки помощник установки предлагает произвести установку драйвера защиты от несанкционированного использования – HASP Device Driver.

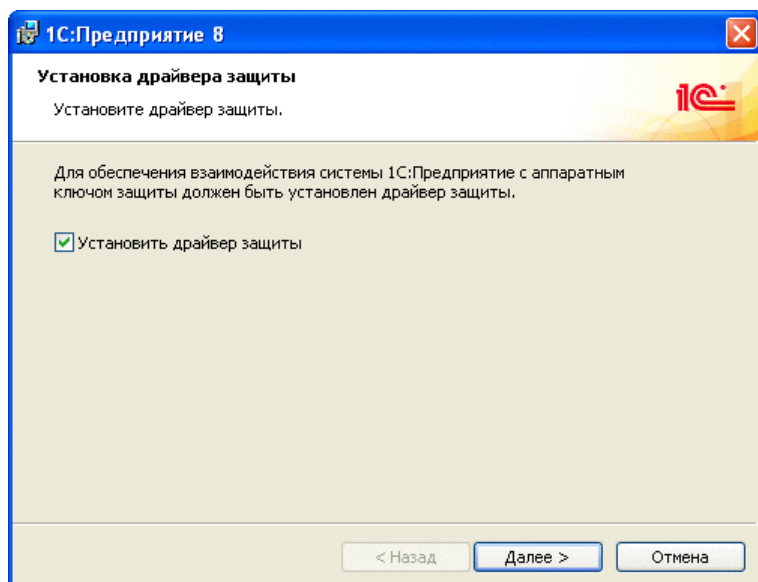


Рис. 9. Установка драйвера защиты

Установка драйвера требуется, если к USB-порту данного компьютера будет присоединяться аппаратный ключ защиты:

- пользователь владеет [Лицензионным соглашением на использование системы "1С:Предприятие" на одном рабочем месте](#);
- пользователь владеет дополнительным [Лицензионным соглашением на использование системы "1С:Предприятие" на одном дополнительном рабочем месте](#);
- пользователь владеет [Лицензионным соглашением на использование сервера "1С:Предприятия"](#).

ПРИМЕЧАНИЕ. Рекомендуется выполнять **установку** драйвера защиты **до того**, как ключ защиты будет присоединен к USB-порту компьютера.

Также установку драйвера можно выполнить, используя пункт меню [Пуск – Программы – 1С Предприятие 8 – Установка HASP Device Driver](#).

2.2.2.7. Завершение установки

Если установка завершена успешно, откроется завершающая страница помощника установки. После нажатия кнопки [Готов](#) установка будет завершена.

Если отмечен флажок [Открыть файл Readme](#), то будет открыт файл с информацией, которую рекомендуется прочитать перед использованием данной версии системы.

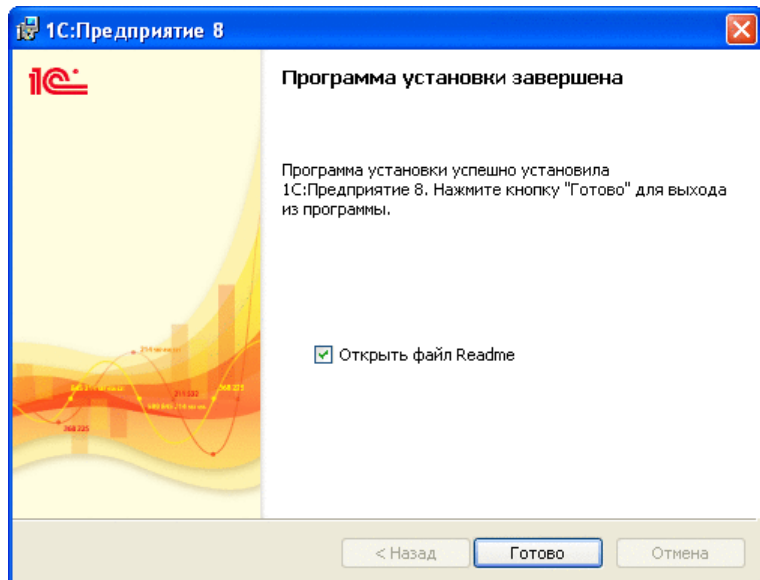


Рис. 10. Завершение установки

2.3. Типовые сценарии установки «1С:Предприятия»

В данном разделе приведены рекомендации по выполнению некоторых наиболее распространенных вариантов установки «1С:Предприятия».

2.3.1. Для ОС Windows

Данный раздел содержит типовые примеры установки компонентов системы программ «1С:Предприятия» для операционной системы Windows.

Для каждого варианта установки будет указан перечень устанавливаемых компонентов и особенности, которые необходимо учесть при установке (если они есть).

2.3.1.1. Тонкий и толстый клиент

Для выполнения данного варианта установки «1С:Предприятия» необходимо разрешить установку следующих компонентов:

- 1С:Предприятие;
- 1С:Предприятие – Тонкий клиент, файловый вариант.

Если используется локальный клиентский ключ, необходимо установить драйвер ключа защиты HASP Device Driver (см. [здесь](#)). Установку драйвера необходимо производить до того, как ключ будет вставлен в USB-разъем компьютера. Если используется сетевой ключ защиты программы, то устанавливать драйвер ключа защиты HASP Device Driver нет необходимости. Подробнее о настройке доступа к ключу защиты см. [здесь](#).

Возможен запуск:

- конфигуратора,
- тонкого клиента,
- толстого клиента.

Возможно использование следующих информационных баз:

- файловая информационная база, локальный вариант;
- файловая информационная база, сетевой вариант;
- клиент-серверный вариант;
- любая информационная база, доступ через веб-сервер.

2.3.1.2. Тонкий клиент

Для выполнения данного варианта установки «1С:Предприятия» необходимо разрешить установку компонента [1С:Предприятие – Тонкий клиент, файловый вариант](#).

Если используется локальный клиентский ключ, необходимо установить драйвер ключа защиты HASP Device Driver (см. [здесь](#)). Установку драйвера необходимо производить до того, как ключ будет вставлен в USB-разъем компьютера. Если используется сетевой ключ защиты программы, то устанавливать драйвер ключа защиты HASP Device Driver нет необходимости. Подробнее о настройке доступа к ключу защиты см. [здесь](#).

Возможен запуск тонкого клиента.

Возможно использование следующих информационных баз:

- файловая информационная база, локальный вариант;
- файловая информационная база, сетевой вариант;
- клиент-серверный вариант;
- любая информационная база, доступ через веб-сервер.

ПРИМЕЧАНИЕ. Разработка конфигураций при такой установке будет недоступна.

2.3.1.3. Тонкий клиент – клиент-серверный вариант

Для выполнения данного варианта установки «1С:Предприятия» необходимо разрешить установку компонента [1С:Предприятие – Тонкий клиент](#).

Если используется локальный клиентский ключ, необходимо установить драйвер ключа защиты HASP Device Driver (см. [здесь](#)). Установку драйвера необходимо производить до того, как ключ будет вставлен в USB-разъем компьютера. Если используется сетевой ключ защиты программы, то устанавливать драйвер ключа защиты HASP Device Driver нет необходимости. Подробнее о настройке доступа к ключу защиты см. [здесь](#).

Возможен запуск тонкого клиента.

Возможно использование следующих информационных баз:

- клиент-серверный вариант;
- любая информационная база, доступ через веб-сервер.

ПРИМЕЧАНИЕ. Разработка конфигураций при такой установке будет недоступна.

2.3.1.4. Толстый клиент

Для выполнения данного варианта установки «1С:Предприятия» необходимо разрешить установку компонента [1С:Предприятие](#).

Если используется локальный клиентский ключ, следует установить драйвер ключа защиты HASP Device Driver (см. [здесь](#)). Установку драйвера нужно производить до того, как ключ будет вставлен в USB-разъем компьютера. Если используется сетевой ключ защиты программы, то устанавливать драйвер ключа защиты HASP Device Driver нет необходимости. Подробнее о настройке доступа к ключу защиты см. [здесь](#).

Возможен запуск:

- configurатора,
- толстого клиента.

Возможно использование следующих информационных баз:

- файловая информационная база, локальный вариант;
- файловая информационная база, сетевой вариант;
- клиент-серверный вариант.

2.3.1.5. Сервер хранилища конфигураций (протокол TCP)

Для того чтобы установить на компьютер сервер хранилища конфигураций «1С:Предприятия» для работы по протоколу TCP (см. книгу «1С:Предприятие 8.3. Руководство разработчика»), необходимо разрешить установку компонента [Сервер хранилища конфигураций 1С:Предприятия](#).

ПРИМЕЧАНИЕ. Сервер хранилища конфигурации является 32-разрядным приложением.

2.3.1.6. Сервер хранилища конфигураций (протокол HTTP)

Для того чтобы установить на компьютер сервер хранилища конфигураций «1С:Предприятия» для работы по протоколу HTTP (см. книгу «1С:Предприятие 8.3. Руководство разработчика»), необходимо разрешить установку следующих компонентов:

- [Модули расширения веб-сервера](#),
- [Сервер хранилища конфигураций 1С:Предприятия](#).

ПРИМЕЧАНИЕ. Сервер хранилища конфигурации является 32-разрядным приложением.

2.3.1.7. Установка возможности публикации веб-клиента или Web-сервиса

Для того чтобы на компьютере, на котором выполняется установка, можно было опубликовать веб-клиент, необходимо к выбранным компонентам добавить компонент [Модули расширения веб-сервера](#) (если он еще не выбран).

Для публикации веб-клиента или веб-сервиса следует воспользоваться диалогом публикации на веб-сервере или утилитой [webinst](#) (только веб-клиент). Подробное описание этих инструментов см. [здесь](#).

2.3.1.8. Установка возможности использования configurатора

Для того чтобы иметь возможность использовать configurator, нужно к набору компонентов, который необходим в случае конкретной установки, добавить компонент [1С:Предприятие](#) (если он не выбран).

2.3.1.9. Установка с использованием административных средств ОС Windows

2.3.1.9.1. Установка с помощью групповых политик

При установке через групповые политики для указания языка установки нужно указывать соответствующий языковой файл трансформации. Имена файлов соответствуют десятичному представлению LCID Microsoft Windows (с расширением **.mst**). Файл трансформации для русского языка называется [1049.mst](#).

Кроме этого, дополнительно нужно указать файл трансформации [admininstallrestart.mst](#). В этом случае система «1С:Предприятие» при несовпадении версий клиента и сервера будет предлагать перезагрузку компьютера для установки новой версии. Администратор должен позаботиться, чтобы новый дистрибутив уже был добавлен в групповых политиках.

С использованием групповых политик можно устанавливать несколько версий «1С:Предприятия».

Для установки новой версии необходимо создать новую установку в групповых политиках.

2.3.1.9.2. Установка с помощью logon-скрипта

Установку можно производить из скрипта, выполняемого при входе пользователя в домен. Задание соответствующего скрипта выполняет администратор домена.

Если пользователь не имеет прав на установку программного обеспечения, то администратор должен указать выполнение скрипта установки от имени пользователя, который может выполнять установку. Пример такого скрипта см. [здесь](#).

С использованием скрипта можно устанавливать и удалять несколько версий «1С:Предприятия». Для этого нужно вызвать процедуру `installOrUninstall` с необходимыми параметрами (пример скрипта см. [здесь](#)).

Для установки новой версии администратору достаточно скорректировать пути общих сетевых ресурсов и код продукта, который нужно взять из файла [setup.ini](#).

Дополнительно нужно указать файл трансформации [admininstallrelogon.mst](#). В этом случае система «1С:Предприятие» при несовпадении версий клиента и сервера будет предлагать завершить текущий сеанс пользователя для установки новой версии. Администратор должен позаботиться, чтобы скрипт был актуализирован и дистрибутив с новой версии был доступен на сетевом ресурсе.

2.3.1.9.3. Обновление версии

При установке платформы с использованием административных средств в каталоге конфигурационных файлов (см. [здесь](#)) создается файл [admininstall.cfg](#) (см. [здесь](#)).

Если при запуске информационной базы требуемая версия «1С:Предприятия» не обнаружена на компьютере и если у пользователя недостаточно прав для установки требуемой версии, пользователю будет предложено выполнить действие, указанное в файле [admininstall.cfg](#): перезагрузка компьютера или повторный вход в систему (logon).

2.3.1.10. Установка утилиты контроля целостности

Для выполнения данного варианта установки «1С:Предприятия» необходимо разрешить установку компонента [Контроль целостности](#). При необходимости следует изменить каталог установки утилиты.

Утилита доступна в 32- и 64-разрядном вариантах.

2.3.2. Для ОС Linux

Данный раздел содержит типовые примеры установки сервера хранилища конфигураций «1С:Предприятия» для операционной системы Linux.

В дальнейшем вся информация будет приводиться для 32-разрядного RPM-варианта установочных пакетов. Установка осуществляется с помощью менеджера пакетов используемой операционной системы.

Дистрибутив «1С:Предприятия» для ОС Linux представлен в виде нескольких пакетов. Эти пакеты используются как для установки клиентских приложений, так и для установки сервера:

- `1c_enterprise83-client-8.3.<X>-<Y>.i386.rpm;`
- `1c_enterprise83-client-nls-8.3.<X>-<Y>.i386.rpm;`
- `1c_enterprise83-common-8.3.<X>-<Y>.i386.rpm;`
- `1c_enterprise83-common-nls-8.3.<X>-<Y>.i386.rpm;`
- `1c_enterprise83-crs-8.3.<X>-<Y>.i386.rpm` (только для архитектуры i386);
- `1c_enterprise83-server-8.3.<X>-<Y>.i386.rpm;`
- `1c_enterprise83-server-nls-8.3.<X>-<Y>.i386.rpm;`
- `1c_enterprise83-thin-client-8.3.<X>-<Y>.i386.rpm;`
- `1c_enterprise83-thin-client-nls-8.3.<X>-<Y>.i386.rpm;`
- `1c_enterprise83-ws-8.3.<X>-<Y>.i386.rpm;`

- [1c_enterprise83-ws-nls-8.3.<X>-<Y>.i386.rpm](#).

Пакеты содержат в себе следующие компоненты:

- [1c_enterprise83-client](#) – клиентские приложения (толстый клиент и тонкий клиент) «1С:Предприятия»;
- [1c_enterprise83-thin-client](#) – тонкий клиент «1С:Предприятия» (не поддерживается работа с файловым вариантом информационной базы);
- [1c_enterprise83-common](#) – общие компоненты «1С:Предприятия»;
- [1c_enterprise83-server](#) – компоненты сервера «1С:Предприятия» и утилита контроля целостности (см. [здесь](#));
- [1c_enterprise83-ws](#) – адаптер для публикации Web-сервисов «1С:Предприятия» на веб-сервере на основе Apache HTTP Server 2.0 или Apache HTTP Server 2.2;
- [1c_enterprise83-crs](#) – компоненты сервера хранилища конфигурации «1С:Предприятия» (сервер хранилища является 32-разрядным приложением).

Пакеты, содержащие в названии суффикс «-nls», – это дополнительные национальные ресурсы (кроме русского и английского языков) для соответствующего пакета. Например, пакет [1c_enterprise83-server-nls](#) содержит дополнительные национальные ресурсы (кроме русского и английского языков) для компонентов сервера «1С:Предприятия».

ПРИМЕЧАНИЕ 1. Для 64-разрядной RPM-версии «1С:Предприятия» в именах файлов вместо строки [.i386](#). должна указываться строка [.x86_64](#). Так, файл [1c_enterprise83-ws-8.3.3-100.i386.rpm](#) будет иметь имя [1c_enterprise83-ws-8.3.3-100.x86_64.rpm](#).

ПРИМЕЧАНИЕ 2. Для 64-разрядной версии «1С:Предприятия» (для ОС Linux семейства Debian) в именах файлов вместо строки [_i386](#) должна указываться строка [_amd64](#). Так, файл [1c-Enterprise83-ws_8.3.3-100_i386.deb](#) будет иметь имя [1c-Enterprise83-ws_8.3.3-100_amd64.deb](#).

ПРИМЕЧАНИЕ 3. Файлы установки для DEB-систем будут иметь имя, которое отличается от своей версии для RPM-систем только расширением – [.deb](#). Так, файл [1c_enterprise83-ws-8.3.3-100.i386.rpm](#) будет иметь имя [1c_enterprise83-ws-8.3.3-100.i386.deb](#).

При установке нужно учитывать следующие зависимости между пакетами:

- [1c_enterprise83-common](#) не имеет зависимостей;
- [1c_enterprise83-server](#) зависит от [1c_enterprise83-common](#);
- [1c_enterprise83-ws](#) зависит от [1c_enterprise83-common](#);
- [1c_enterprise83-crs](#) зависит от [1c_enterprise83-common](#), [1c_enterprise83-server](#) и [1c_enterprise83-ws](#);
- [1c_enterprise83-client](#) – зависит от [1c_enterprise83-server](#);
- [1c_enterprise83-thin-client](#) – не имеет зависимостей. Для работы тонкого клиента не требуется установки других пакетов из состава «1С:Предприятия». Конфликтует с пакетом [1c_enterprise83-common](#). Может быть установлен или пакет [1c_enterprise83-thin-client](#) или остальные пакеты.
- пакеты национальных ресурсов зависят от соответствующего компонента.

Поэтому для того, чтобы успешно установить пакет, предварительно нужно установить все пакеты, от которых он зависит. Например, для установки компонентов сервера «1С:Предприятия» необходимо сначала установить пакет [1c_enterprise83-common](#) и после этого – [1c_enterprise83-server](#).

После установки клиентских приложений, в меню приложений среды рабочего стола, добавляются ярлыки на запуск программы запуска ([1cstart](#)), тонкого клиента ([1cv8c](#)) и толстого клиента ([1cv8](#)). Создаются ярлыки только на реально установленные приложения. Ярлыки относятся к подкатегории [Finance](#) категории [Office](#).

Установка должна выполняться от лица пользователя [root](#).

2.3.2.1. Тонкий и толстый клиент

Для выполнения данного варианта установки «1С:Предприятия» необходимо установить следующие пакеты:

- [1c_enterprise83-common-8.3.<X>-<Y>.i386.rpm](#);
- [1c_enterprise83-common-nls-8.3.<X>-<Y>.i386.rpm](#);
- [1c_enterprise83-server-8.3.<X>-<Y>.i386.rpm](#);
- [1c_enterprise83-server-nls-8.3.<X>-<Y>.i386.rpm](#);
- [1c_enterprise83-client-8.3.<X>-<Y>.i386.rpm](#);
- [1c_enterprise83-client-nls-8.3.<X>-<Y>.i386.rpm](#).

Если используется локальный клиентский ключ, необходимо установить драйвер ключа защиты HASP Device Driver (см. [здесь](#)). Установку драйвера необходимо производить до того, как ключ будет вставлен в USB-разъем компьютера. Если используется сетевой ключ защиты программы, то устанавливать драйвер ключа защиты HASP Device Driver нет необходимости. Подробнее о настройке доступа к ключу защиты см. [здесь](#).

Возможен запуск:

- конфигуратора,
- тонкого клиента,
- толстого клиента.

Возможно использование следующих информационных баз:

- файловая информационная база, локальный вариант;
- файловая информационная база, сетевой вариант;
- клиент-серверный вариант;
- любая информационная база, доступ через веб-сервер.

2.3.2.2. Тонкий клиент

Для выполнения данного варианта установки «1С:Предприятия» необходимо установить следующие пакеты:

- [1c_enterprise83-thin-client-8.3.<X>-<Y>.i386.rpm](#);
- [1c_enterprise83-thin-client-nls-8.3.<X>-<Y>.i386.rpm](#).

Если используется локальный клиентский ключ, необходимо установить драйвер ключа защиты HASP Device Driver (см. [здесь](#)). Установку драйвера необходимо производить до того, как ключ будет вставлен в USB-разъем компьютера. Если используется сетевой ключ защиты программы, то устанавливать драйвер ключа защиты HASP Device Driver нет необходимости. Подробнее о настройке доступа к ключу защиты см. [здесь](#).

Возможен запуск тонкого клиента.

Возможно использование следующих информационных баз:

- файловая информационная база, локальный вариант;
- файловая информационная база, сетевой вариант;
- клиент-серверный вариант;
- любая информационная база, доступ через веб-сервер.

ПРИМЕЧАНИЕ. Разработка конфигураций при такой установке будет недоступна.

2.3.2.3. Сервер хранилища конфигураций (протокол TCP)

Для того чтобы установить на компьютер сервер хранилища конфигураций «1С:Предприятия» для работы по протоколу TCP (см. книгу «1С:Предприятие 8.3. Руководство разработчика»), необходимо установить следующие компоненты (конкретной версии):

- [1c_enterprise83-common-8.3.<X>-<Y>.i386.rpm](#),
- [1c_enterprise83-common-nls-8.3.<X>-<Y>.i386.rpm](#),
- [1c_enterprise83-server-8.3.<X>-<Y>.i386.rpm](#),
- [1c_enterprise83-server-nls-8.3.<X>-<Y>.i386.rpm](#),
- [1c_enterprise83-crs-8.3.<X>-<Y>.i386.rpm](#).

ПРИМЕЧАНИЕ. Сервер хранилища конфигурации является 32-разрядным приложением.

2.3.2.4. Сервер хранилища конфигураций (протокол HTTP)

Для того чтобы установить на компьютер сервер хранилища конфигураций «1С:Предприятия» для работы по протоколу HTTP (см. книгу «1С:Предприятие 8.3. Руководство разработчика»), необходимо установить следующие компоненты (конкретной версии):

- [1c_enterprise83-common-8.3.<X>-<Y>.i386.rpm](#),
- [1c_enterprise83-common-nls-8.3.<X>-<Y>.i386.rpm](#),
- [1c_enterprise83-server-8.3.<X>-<Y>.i386.rpm](#),
- [1c_enterprise83-server-nls-8.3.<X>-<Y>.i386.rpm](#),
- [1c_enterprise83-ws-8.3.<X>-<Y>.i386.rpm](#),
- [1c_enterprise83-ws-nls-8.3.<X>-<Y>.i386.rpm](#),
- [1c_enterprise83-crs-8.3.<X>-<Y>.i386.rpm](#).

ПРИМЕЧАНИЕ. Сервер хранилища конфигурации является 32-разрядным приложением.

2.3.2.5. Работа веб-клиента

Для того чтобы иметь возможность пользоваться веб-клиентом, следует на компьютер с веб-сервером установить следующие пакеты:

- `1c_enterprise83-common-8.3.<X>-<Y>.i386.rpm`,
- `1c_enterprise83-common-nls-8.3.<X>-<Y>.i386.rpm`,
- `1c_enterprise83-server-8.3.<X>-<Y>.i386.rpm`,
- `1c_enterprise83-server-nls-8.3.<X>-<Y>.i386.rpm`,
- `1c_enterprise83-ws-8.3.<X>-<Y>.i386.rpm`,
- `1c_enterprise83-ws-nls-8.3.<X>-<Y>.i386.rpm`,
- `1c_enterprise83-crs-8.3.<X>-<Y>.i386.rpm`.

Публикацию веб-клиента следует выполнять с помощью утилиты командной строки `webinst` (см. [здесь](#)).

примечание. Пакет `1c_enterprise83-crs-8.3.<X>-<Y>` предназначен только для архитектуры i386.

2.3.2.6. Утилита контроля целостности

Для выполнения данного варианта установки «1С:Предприятия» необходимо установить следующие пакеты:

- `1c_enterprise83-server-8.3.<X>-<Y>.i386.rpm`,
- `1c_enterprise83-server-nls-8.3.<X>-<Y>.i386.rpm`.

Для установки утилиты в другой каталог (отличный от стандартного каталога установки сервер «1С:Предприятие») следует воспользоваться средствами пакетного менеджера вашей версии ОС Linux.

2.4. Рекомендации по развертыванию системы

ПРИМЕЧАНИЕ. Данная рекомендация применима только в том случае, если на клиентских компьютерах установлена ОС Windows.

Для облегчения автоматической установки новых версий «1С:Предприятия» на компьютер пользователя (включая первоначальную установку системы) рекомендуется следующее расположение файлов в сетевом каталоге:

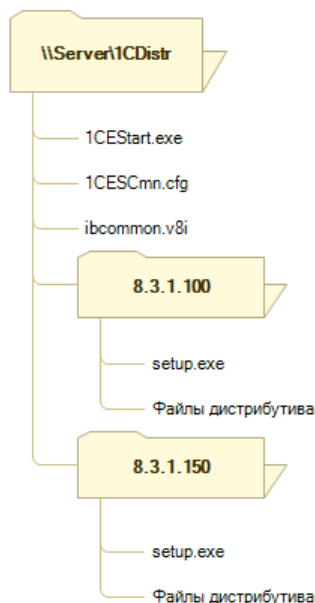


Рис. 11. Структура каталога

В вышеприведенной схеме:

- `\\Server\1CEDistr` – каталог на сервере `Server`, в котором расположены файлы, необходимые для развертывания системы.
- `1cestart` – программа запуска. Для начальной установки достаточно запустить программу запуска из этого сетевого каталога. Эту программу рекомендуется взять из самой новой версии «1С:Предприятия» версии 8.3, планируемой к установке.
- `ibcommon.v8i` – список общих информационных баз, если он существует (имя условное и не является обязательным). Описание формата файла см. [здесь](#).
- `1cescmn.cfg` – общий конфигурационный файл. Рекомендуется указать в нем следующие параметры:
 - `CommonInfoBases=ИмяФайлаСоСпискомОбщихИнформационныхБаз.v8i` – если нужно, чтобы у пользователя при запуске системы был

необходимый список информационных баз. Описание этого параметра см. [здесь](#).

- **InstallComponents** – указать те компоненты, которые необходимы для установки на компьютеры пользователей. Описание этого параметра см. [здесь](#).
- **DistributiveLocation** – указать каталог расположения дистрибутивов «1С:Предприятия». Описание этого параметра см. [здесь](#).
- **8.3.3.100** и **8.3.3.150** – каталоги с дистрибутивами соответствующих версий «1С:Предприятия».
- **Setup.exe** – программа запуска установки системы программ «1С:Предприятие».

В данном примере будет использоваться общий конфигурационный файл следующего содержания (будут устанавливаться все компоненты и языки: русский и английский):

Содержимое файла `1cescmn.cfg`:

```
CommonInfoBases=ibcommon.v8i
DistributiveLocation=\\Server\1CEDistr
InstallComponents=DESIGNERALLCLIENTS=1 SERVER=1 WEBSERVEREXT=1 CONFREPOSSERVER=1 SERVERCLIENT=1 CONVERTER77=1 LANGUAGES=ru
```

[Копировать в буфер обмена](#)

ВНИМАНИЕ! Общий конфигурационный `1cescmn.cfg` файл не должен располагаться на компьютере пользователя.

После выхода новой версии системы «1С:Предприятие» (например, 8.3.3.200) необходимо будет только скопировать файлы дистрибутива в каталог `\\Server\1CDistr\8.3.3.200`. Остальное система выполнит автоматически при запуске «1С:Предприятия» пользователем.

При использовании такой схемы развертывания нужно помнить о следующих особенностях:

- Установка «1С:Предприятия», выполняемая с помощью программы запуска, **всегда** выполняется в каталог по умолчанию. Для изменения этого каталога следует вручную запускать программу установки (`setup.exe`) соответствующей версии.
- Из общего конфигурационного файла при установке используется **только** параметр `InstallComponents`. Другие параметры не влияют на процесс установки и не переносятся из общего в локальный конфигурационный файл. В примере, рассмотренном выше, будут использованы следующие компоненты:

```
InstallComponents=DESIGNERALLCLIENTS=1 SERVER=1 WEBSERVEREXT=1 CONFREPOSSERVER=1 SERVERCLIENT=1 CONVERTER77=1 LANGUAGES=ru
```

[Копировать в буфер обмена](#)

- Во время установки в локальный конфигурационный файл записывается параметр `CommonCfgLocation`, значением которого является путь к общему конфигурационному файлу, который расположен в каталоге развертывания. В примере, рассмотренном выше, путь к этому файлу будет следующим: `\\server\1cdistr\1cescmn.cfg`. При дальнейшей работе параметры, указанные в этом файле, будут использованы программами запуска и в диалоге запуска клиентского приложения (см. [здесь](#)).

2.5. Установка и настройка дополнительного программного обеспечения

2.5.1. Для ОС Windows

2.5.1.1. Аутентификация операционной системы при использовании веб-сервера Apache

Существует возможность настройки поддержки аутентификации операционной системы для тонкого и веб-клиентов при использовании веб-сервера Apache. В данном разделе предполагается, что веб-сервер Apache уже установлен и настроен для обеспечения доступа с помощью веб-клиента.

ВНИМАНИЕ! Для настройки аутентификации операционной системы в сети должен быть развернут PDC под управлением Windows 2000 и выше.

Для настройки нужно выполнить следующее:

- Необходимо установить модуль аутентификации `mod_auth_sspi`. Модуль расположен по адресу: <http://sourceforge.net/projects/mod-auth-sspi>. Версия модуля должна соответствовать используемой версии Apache.
- Скопировать файл `mod_auth_sspi.so` в подкаталог `modules` установочного каталога Apache.
- Добавить в конфигурационный файл `httpd.conf` веб-сервера Apache строку:

```
LoadModule sspi_auth_module modules/mod_auth_sspi.so
```

[Копировать в буфер обмена](#)

- Добавить в секцию, описывающую нужный виртуальный каталог, следующие строки (выделены полужирным шрифтом):

```
<Directory "c:/www/MyApp">
  AllowOverride None
  Options None
  Order allow,deny
  Allow from all
  SetHandler lc-application
  ManagedApplicationDescriptor c:/www/MyApp/default.vrd
  AuthName "1C:Enterprise web client"
  AuthType SSPI
  SSPIAuth On
  SSPIAuthoritative On
  SSPIPackage Negotiate
  SSPIOfferBasic Off
```

[Копировать в буфер обмена](#)

```
Require valid-user
</Directory>
```

- В свойствах учетной записи компьютера, на котором запущен веб-сервер, установить флажок [Доверять компьютеру делегирование \(Trust computer for delegation\)](#).
- Перезапустить веб-сервер.

2.5.2. Для ОС Linux

2.5.2.1. Особенности настройки для работы с файловым вариантом информационной базы

При работе с файловым вариантом информационной базы, на компьютере, работающем под управлением ОС Linux, следует помнить о следующих особенностях:

- При подключении сетевого ресурса в ОС Linux с помощью команды `mount.cifs` не следует использовать ключ `nobrl` (<http://www.samba.org/samba/docs/man/manpages-3/mount.cifs.8.html>, на английском языке).
- При предоставлении доступа к каталогу с информационной базой с помощью системы Samba, не следует использовать параметр `locking=no` применительно к публикуемому ресурсу, в файле `smb.conf` (<http://www.samba.org/samba/docs/man/manpages-3/smb.conf.5.html>, на английском языке).
- Если планируется работа с файловой информационной базой нескольких различных пользователей одного компьютера, следует учитывать следующую особенность: в ОС Linux для вновь создаваемых файлов в качестве владельца устанавливается пользователь, от имени которого запущен процесс, создающий файл и основная группа этого пользователя. В результате, при попытке совместной работы разных пользователей в одной файловой информационной базе второй (и последующие) пользователи (этого компьютера) не получат доступа к созданному файлу. «1С:Предприятие» создает файлы с разрешением на запись и чтение для пользователя-владельца и группы-владельца. Поэтому, для обеспечения совместной работы пользователей нужно внести их в одну группу, и назначить данную группу в качестве владельца каталога, в котором находится информационная база. После этого для данного каталога нужно установить `sticky-bit` на группу командой `chmod g+s ib_dir`, где `ib_dir` – имя каталога, в котором расположена информационная база. В результате в качестве группы-владельца вновь создаваемых файлов в данном каталоге будет назначаться не основная группа пользователя, который создает данные файлы, а группа-владелец основного каталога информационной базы.

Настройка ОС для совместной работы с хранилищем конфигурации выполняется таким же образом, только в качестве каталога информационной базы выступает каталог с хранилищем конфигурации.

2.5.2.2. Установка шрифтов

Для корректной работы «1С:Предприятия», в ОС Linux должны быть установлены шрифты из состава [Microsoft Core Fonts](#). Установку данных шрифтов можно осуществить следующими способами:

- Использовать пакет, входящий в состав дистрибутива (проверяется в каждом дистрибутиве).
- Для RPM-варианта ОС Linux информацию по установке можно получить по адресу: <http://corefonts.sourceforge.net/> (на английском языке).
- Можно выполнить установку вручную. Для этого необходимо:
 - Скачать все файлы со шрифтами по ссылке: <http://sourceforge.net/projects/corefonts/files/the%20fonts/final/>.
 - Распаковать файлы.
 - Поместить файлы шрифтов в каталог `~/fonts` каталога пользователя (или каталогов пользователей), от имени которого (-ых) выполняется запуск «1С:Предприятия». Здесь `~` – домашний каталог пользователя.

«1С:Предприятие» выполняет поиск TrueType-шрифтов в следующих каталогах:

- `/usr/share/fonts/truetype/msttcorefonts` – стандартный каталог для размещения шрифтов для DEB-вариантов систем.
- `/usr/share/fonts/msttcorefonts` – стандартный каталог для размещения шрифтов, устанавливаемых в RPM-вариантах систем пакетом `corefonts`.
- `~/fonts` – стандартный каталог для размещения пользовательских шрифтов в Linux.

2.5.2.3. Аутентификация операционной системы при использовании веб-сервера Apache

Существует возможность настройки поддержки аутентификации операционной системы для тонкого и веб-клиентов при использовании веб-сервера Apache. В данном разделе предполагается, что веб-сервер Apache уже установлен и настроен для обеспечения доступа с помощью веб-клиента.

ВНИМАНИЕ! Для настройки аутентификации операционной системы в сети должен быть развернут PDC под управлением Windows 2000 и выше.

Для настройки нужно выполнить следующее:

- Необходимо установить модуль аутентификации `mod_auth_kerb`. Он входит в большинство дистрибутивов, нужно лишь установить соответствующий пакет. Для ОС Fedora этот пакет называется `mod_auth_kerb`, а для Debian – `libapache2-mod-auth-kerb`. Если же в поставку используемой операционной системы этот модуль не входит, можно загрузить его исходный код с [домашней страницы](http://modauthkerb.sourceforge.net/) проекта: <http://modauthkerb.sourceforge.net/>.
- Возможны следующие варианты установки:
 - Модуль устанавливается из дистрибутива операционной системы. В этом случае необходимо только перезапустить веб-сервер, и

модуль подключится.

- Если модуль компилируется и устанавливается самостоятельно (инструкции по установке находятся здесь: <http://modauthkerb.sourceforge.net/install.html>, на английском языке), то необходимо добавить в конфигурационный файл `httpd.conf` веб-сервера Apache строку, указанную ниже, и перезапустить Apache:

[Копировать в буфер обмена](#)

```
LoadModule auth_kerb_module /путь/к/файлу/mod_auth_kerb.so
```

Для проведения аутентификации модулю требуется закрытый ключ Kerberos на имя [HTTP/Server.domain@DOMAIN](#). Его нужно сгенерировать по принципу, описанному в руководстве по настройке аутентификации Kerberos. Следует учесть, что для учетной записи, с которой будет ассоциировано имя [HTTP/Server.domain@DOMAIN](#), нужно установить флажок *Учетная запись доверена для делегирования (Account is trusted for delegation)*.

Предположим, что файл с ключом называется `HTTP.keytab` и располагается он в домашнем каталоге пользователя `usr1cv8`.

Теперь необходимо добавить в секцию, описывающую виртуальный каталог веб-сервера, следующие строки:

[Копировать в буфер обмена](#)

```
<Directory "/home/usr1cv8/www/MyApp">
  AllowOverride None
  Options None
  Order allow,deny
  Allow from all
  SetHandler lc-application
  ManagedApplicationDescriptor /home/usr1cv8/www/MyApp/default.vrd
  AuthName "1C:Enterprise web client"
  AuthType Kerberos
  Krb5Keytab /home/usr1cv8/HTTP.keytab
  KrbVerifyKDC off
  KrbDelegateBasi coff
  KrbServiceName HTTP/Server.domain@DOMAIN
  KrbSaveCredentials on
  KrbMethodK5Passwd off
  KrbMethod Negotiateon
  Require valid-user
</Directory>
```

Необходимо указывать корректный путь к файлу ключа и при этом файл должен быть доступен для чтения тому пользователю, от имени которого запускается Apache.

ВНИМАНИЕ! Kerberos-аутентификация в домене, содержащем контроллеры как Windows 2000, так и Windows 2003, веб-серверы на Linux и серверы «1С:Предприятия» на Windows, может не работать в силу особенностей реализации Kerberos на Windows 2000.

2.6. Особенности регистрации компонент

Программа установки выполняет регистрацию некоторых компонент (COM-соединение и т. д.). При этом регистрация выполняется следующим образом:

- **COM-соединение.** Программа установки выполняет регистрацию «для компьютера». Регистрацию «для пользователя» можно выполнить с помощью команды:

[Копировать в буфер обмена](#)

```
regsvr32 -n -i:user comcntr.dll
```

- **Клиентское приложение (COM-объект «V83.Application»).** Программа установки (и запуск клиентского приложения с ключом `RegServer`) выполняет регистрацию «для компьютера». Если привилегий для выполнения данной операции недостаточно, пользователю предлагается согласиться с регистрацией «для пользователя».

Глава 3. Установка конфигураций

3.1. Общие сведения о каталогах шаблонов

Информационные базы создаются из шаблонов. Установка шаблонов выполняется специальной программой установки, которая создается при создании комплекта поставки из конфигулятора. Шаблон представляет собой совокупность файлов поставки, файла манифеста и сопутствующих файлов, из которых производится создание информационной базы. Подробнее процесс создания поставки описан в книге «1С:Предприятие 8.3. Руководство разработчика».

Для использования в качестве шаблонов конфигурации и/или информационные базы должны быть установлены на компьютере пользователя определенным образом – все шаблоны должны находиться в подкаталогах определенной структуры и сопровождаться файлами-манифестами, описывающими установленные шаблоны.

В системе может существовать несколько каталогов шаблонов, которые могут располагаться, в том числе, на сетевых дисках. Таким образом, может быть создана единая база каталогов шаблонов, из которых пользователи смогут выполнять установку или обновление конфигураций.

По умолчанию каталог хранения шаблонов имеет имя `tplmts` и расположен в каталоге `%APPDATA%\1C\1Cv8`. Пользователь может изменить положение каталога шаблонов и указать системе ссылки на другие каталоги (с произвольными именами). Подробное описание процесса указания новых каталогов шаблонов см. [здесь](#). В документации будет описываться работа с каталогом шаблонов по умолчанию, однако все сказанное будет применимо и для других каталогов шаблонов.

Каталог шаблонов делится на подкаталоги поставщиков – каждый поставщик решений выбирает подкаталог, основываясь на наименовании своей организации (например, фирма «1С» помещает свои решения в каталог «1С»). Внутри выбранного подкаталога порядок размещения решений не регламентирован. Однако для каждого решения рекомендуется выбирать каталог с именем, соответствующим этому решению.

Внутри подкаталога поставщиков производится разделение по каталогам, соответствующим версиям выпускаемых решений. Например, `tplmts\1C\Accounting\1.5.7.5`.

Рекомендуется придерживаться приведенного порядка организации каталогов во избежание пересечений между различными поставщиками.

3.2. Установка шаблона конфигурации

Для того чтобы установить конфигурацию, необходимо установить ее шаблон. Для этого нужно запустить файл `setup.exe`, который расположен в каталоге с конфигурацией.

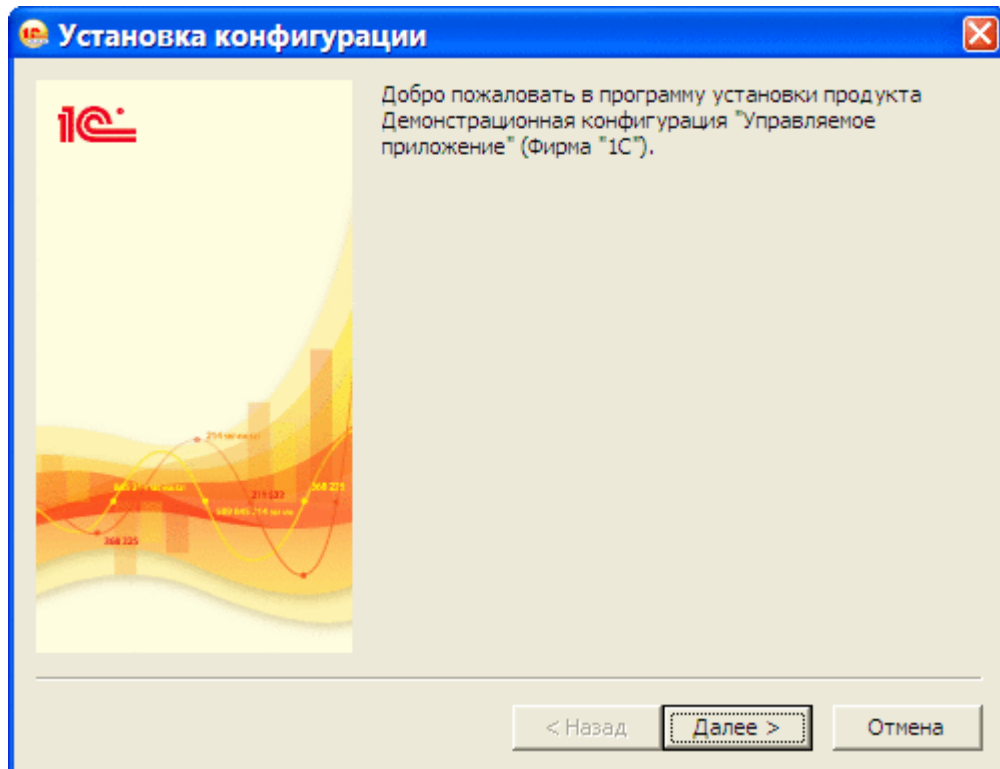


Рис. 12. Установка конфигурации

Затем необходимо указать, в какой каталог выполнить установку шаблона конфигурации. Путь к каталогу шаблонов, который будет указан по умолчанию, определяется следующим образом:

- Перебираются параметры `ConfigurationTemplatesLocation` файла `1cestart.cfg` (см. [здесь](#)) и в них ищется каталог шаблонов, расположенный на локальной машине, в который есть доступ по записи для пользователя, выполняющего установку. Если в файле `1cestart.cfg` указано несколько параметров с такими каталогами, то будет выбран первый в порядке указания в файле.
- Если не обнаружено каталогов шаблонов на локальной машине, то будет создан каталог `%APPDATA%\1C\1Cv8\tmlpts`, и он будет использован в качестве каталога шаблонов по умолчанию. Также запись об этом каталоге будет указана первым параметром `ConfigurationTemplatesLocation` файла `1cestart.cfg`.

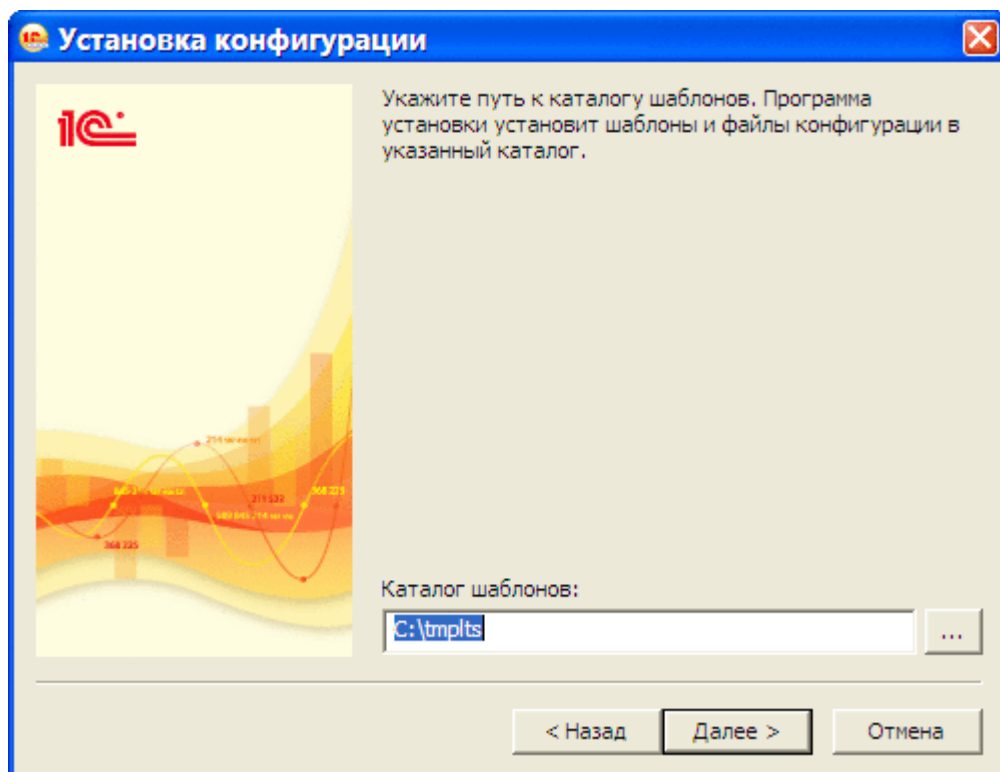


Рис. 13. Выбор каталога шаблонов

Если пользователя не устраивает предложенный по умолчанию путь, то он может ввести путь к другому каталогу. Система выполнит попытку установки шаблона конфигурации в указанный каталог и добавит этот каталог первым параметром `ConfigurationTemplatesLocation` файла `1cestart.cfg`, если попытка установки окончилась успешно.

Далее программа установки выполнит копирование файлов шаблона конфигурации в указанный каталог.

Имеется возможность уменьшить количество шагов мастера установки шаблона конфигурации. Для этого программу `setup.exe` необходимо запускать с ключом `/s`. В этом случае будет показан только стартовый диалог с приветствием, и затем будет показан ход копирования файлов с шаблоном конфигурации. При этом каталог шаблонов конфигурации будет получен из файла `1cestart.cfg` (см. [здесь](#)), а если там нет записи о расположении каталога шаблонов, то будет создан каталог `%APPDATA%\1C\1Cv8\tplts`, и он будет использован в качестве каталога шаблонов по умолчанию. Также запись об этом каталоге будет указана первым параметром `ConfigurationTemplatesLocation` файла `1cestart.cfg`.

3.3. Создание новой информационной базы из шаблона

Для того чтобы из установленного шаблона создать конкретную информационную базу, необходимо запустить «1С:Предприятие», затем в открывшемся окне нажать кнопку [Добавить...](#)

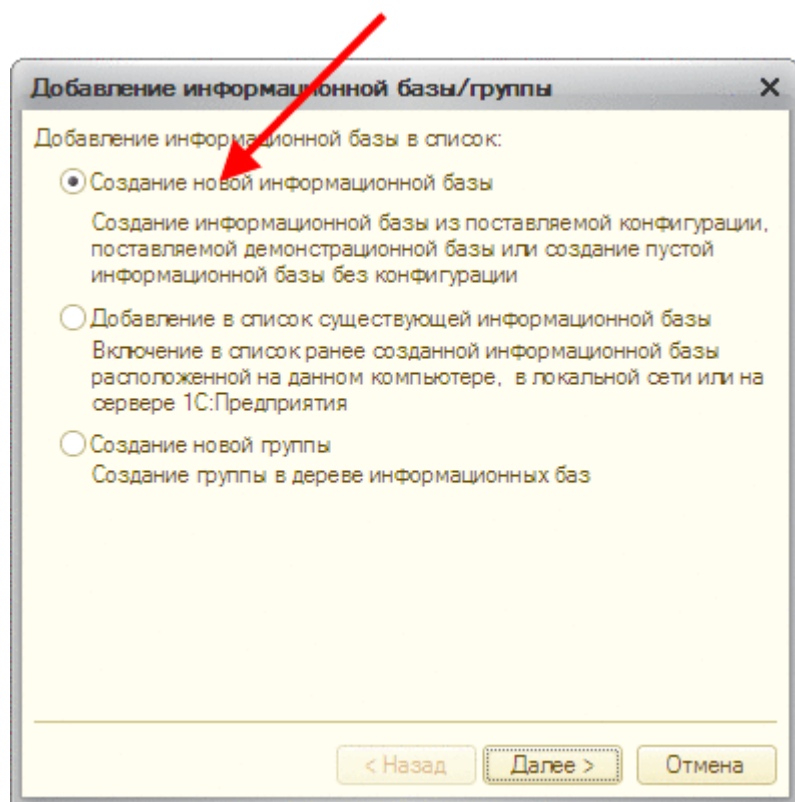


Рис. 14. Добавить новую информационную базу

Затем нужно выбрать тот шаблон, который был установлен ранее (см. [здесь](#)), и продолжить установку (нажать кнопку [Далее >](#)). Процесс формирования дерева шаблонов может занимать существенное время.

Если информационная база создается с помощью программы запуска (см. [здесь](#)), то имеется возможность создать информационную базу из шаблона любой версии (т. е. «1С:Предприятие» версий 8.0, 8.1, 8.2 и 8.3). Если же создание выполняется с помощью толстого клиента, то для выбора доступны только шаблоны той версии, что и запускаемый файл.

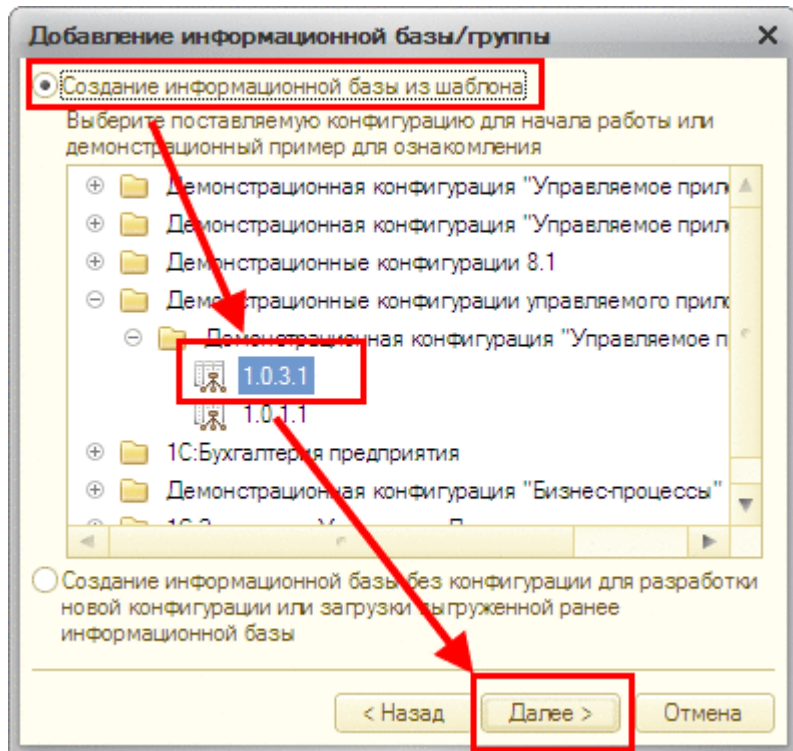


Рис. 15. Выбор шаблона

Далее нужно указать имя информационной базы и различные параметры (см. [здесь](#)), после чего система выполнит создание информационной базы.

Если планируется создать информационную базу для последующей загрузки в нее данных из файла выгрузки (*.dt) или для разработки новой конфигурации, то в окне [Добавление информационной базы/группы](#) (см. [рис. 15](#)) нужно выбрать пункт [Создание информационной базы без конфигурации...](#)

Глава 4. Запуск компонентов системы

При установке «1С:Предприятия» в меню **Пуск – Программы** будет создана группа **1С Предприятие 8**, в которой будет сформирована структура меню, аналогичная представленной на [рис. 16](#).

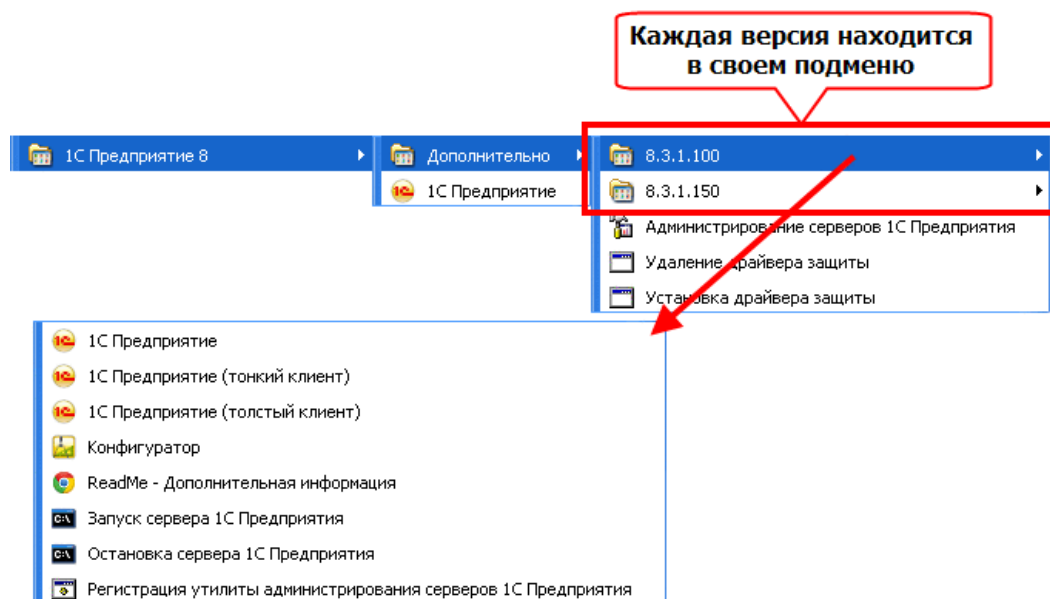


Рис. 16. Структура меню

Пункты	Назначение
1С Предприятие	Вызов программы запуска (1cstart)
8.3.3.100	Каталоги, содержащие ссылки на запуск компонентов системы конкретной версии (в данном случае установлены две версии: 8.3.3.100 и 8.3.3.150)
8.3.3.150	
Установка драйвера защиты	Запуск установки драйвера защиты
Удаление драйвера защиты	Запуск удаления драйвера защиты
1С Предприятие (тонкий клиент)	Запуск системы в режиме тонкого клиента «1С:Предприятия»
1С Предприятие (толстый клиент)	Запуск системы в режиме толстого клиента «1С:Предприятия»
Конфигуратор	Запуск системы в режиме Конфигуратор
ReadMe – Дополнительная информация	Дополнительная информация, не вошедшая в документацию
Конвертор ИБ 1С Предприятия 7.7	Программа для конвертации информационных баз в формате «1С:Предприятия 7.7»
Администрирование серверов 1С Предприятия	Утилита администрирования кластера серверов (если были установлены компоненты доступа к кластеру серверов «1С:Предприятия»)
Запуск сервера 1С Предприятия	Запуск сервера «1С:Предприятия» как сервиса (если при установке сервера был установлен флажок Установить сервер 1С:Предприятия 8 как сервис Windows) или как приложения (если при установке сервера был установлен флажок Установить сервер 1С:Предприятия 8 как сервис Windows). Остановка сервера в этом случае выполняется как закрытие обычного приложения
Регистрация утилиты администрирования серверов 1С Предприятия	Выполняет регистрацию утилиты администрирования серверов «1С:Предприятия» (radmin.dll) конкретной версии, после чего можно подключаться к серверам этой версии с помощью утилиты администрирования

4.1. Режимы работы системы

Система «1С:Предприятие» может работать в одном из следующих режимов:

Режим работы	Описание
Конфигуратор	Режим конфигурирования системы. Позволяет редактировать структуры данных, выполнять обновление конфигурации, формировать список пользователей системы с назначением прав доступа на работу в системе, выполнять выгрузку и загрузку данных
1С:Предприятие	Исполняемая часть системы. На основе структур данных, описанных в конфигураторе, выполняет собственно ввод и обработку информации (работу со справочниками, документами, отчетами и так далее). Исполняемая часть системы, в свою очередь, может использоваться в трех разных вариантах: <ul style="list-style-type: none"> • тонкий клиент – исполняемый файл 1cv8c;

- **веб-клиент** – нет исполняемого файла (его роль играет веб-браузер);
- **толстый клиент** – исполняемый файл `1cv8`.

Толстый клиент может выполнять как конфигурации, которые написаны для предыдущих версий системы «1С:Предприятия», так и конфигурации, написанные в режиме управляемого приложения.

Тонкий и веб-клиенты могут исполнять только те конфигурации, которые написаны в режиме управляемого приложения.

4.2. Запуск клиентского приложения или конфигуратора

Запустить «1С:Предприятие» в каком-либо режиме запуска можно несколькими способами:

- С помощью программы запуска (`1cestart`) – рекомендованный способ.
- С помощью интерактивной программы запуска (`1cv8s`).
- С помощью исполняемого файла толстого (`1cv8`) или тонкого (`1cv8c`) клиента конкретной версии системы.
- С помощью веб-браузера (только веб-клиент).

Для запуска системы используются конфигурационные файлы:

- Локальный конфигурационный файл – `1cestart.cfg`, описание см. [здесь](#).
- Локальный конфигурационный файл для всех пользователей – `1cestart.cfg`, см. [здесь](#).
- Общий конфигурационный файл – `1cescmn.cfg`, описание см. [здесь](#) (только для ОС Windows).

Далее будут подробно описаны способы запуска.

4.2.1. Программа запуска

Расположение файла `1cestart`:

- Для ОС Windows:
 - Для 32-разрядной версии ОС: `C:\Program Files\1cv8\common`.
 - Для 64-разрядной версии ОС: `C:\Program Files (x86)\1cv8\common`.
- Для ОС Linux:
 - Для 32-разрядной версии ОС: `/opt/1C/v8.3/i386`.
 - Для 64-разрядной версии ОС: `/opt/1C/v8.3/x86_64`.

Программа запуска позволяет запускать все виды клиентских приложений (толстый клиент, тонкий клиент, веб-клиент) и конфигуратор.

Программа запуска может быть запущена либо без параметров, либо с указанием ссылки на конкретную информационную базу.

Если на клиентском компьютере установлена ОС Windows, то:

- Программа запуска также может быть расположена на сетевом ресурсе (для ее работы не требуются дополнительные программные компоненты) и позволяет выполнять как начальную установку системы на компьютер, так и установку новых версий системы программ «1С:Предприятие». Если программа запуска находит общий конфигурационный файл в каталоге, откуда она запущена, то ссылка на этот файл записывается в параметр `CommonCfgLocation` локального конфигурационного файла.
- При установке системы с помощью программы запуска возможно появление предложения о перезагрузке операционной системы.

4.2.1.1. Запуск без параметров

Если программа запуска запускается без указания параметров, то используется следующий алгоритм запуска:

- Если выполняется запуск с сетевого диска, то происходит попытка обнаружения общего конфигурационного файла в каталоге запуска. В случае успеха происходит считывание параметров из этого файла.
- Происходит попытка обнаружения локального конфигурационного файла. В случае успеха происходит считывание параметров из этого файла.
- Происходит поиск установленных версий платформы в соответствии с данными, полученными из параметров `InstalledLocation` конфигурационных файлов. Если данный параметр не указан в конфигурационных файлах, запуск прекращается с выдачей сообщения об ошибке.
- Определяется максимальный номер установленной версии «1С:Предприятия».
- Определяется максимальный номер версии, доступной к установке, в каталогах, полученных из параметров `DistributiveLocation`

конфигурационных файлов.

• Если существует версия с большим номером, доступная для установки, происходит автоматическая установка новой версии с параметрами, полученными из параметров `InstallComponents` конфигурационных файлов. Если этот параметр не указан, то выполняется установка тонкого клиента, толстого клиента и компонентов доступа к серверам «1С:Предприятия».

Установка будет выполнена в следующих случаях:

- Пользователь, который запускает программу запуска, входит в группу локальных администраторов.
- Пользователь, который запускает программу запуска, не входит в группу локальных администраторов, но для пользователя и компьютера разрешена установка программ (ключ реестра `AlwaysInstallElevated`).
- Выполняется запуск интерактивной программы запуска из каталога версии (существующей или установленной на предыдущем шаге). Запуск выполняется с указанием параметра `/AppAutoCheckVersion`.

4.2.1.2. Запуск с указанием информационной базы

Если программа запуска запускается с указанием имени информационной базы (параметр `/IBName`), то используется следующий алгоритм запуска:

- Выполняется считывание параметров из локального (`1cestart.cfg`, см. [здесь](#)) и общего (`1cescmn.cfg`, см. [здесь](#)) конфигурационных файлов.
- Формируется общий список информационных баз из локального списка баз (файл `ibases.v8i`, см. [здесь](#)) и параметров `CommonInfoBases` конфигурационных файлов.
- Если указанное имя информационной базы не найдено в получившемся списке, запуск прекращается с выдачей сообщения об ошибке.
- Если обнаружена информационная база с указанным именем, то происходит определение параметров запуска из свойств информационной базы и запускается соответствующий клиент с заданными параметрами. Из свойств информационной базы определяются следующие параметры:
 - вид клиента;
 - номер версии, требуемый для работы;
 - прочие параметры, хранящиеся в свойствах информационной базы.
- Запуск выполняется с указанием параметра `/AppAutoCheckVersion`.

4.2.2. Интерактивная программа запуска

Расположение файла `1cv8s`:

- Для ОС Windows:
 - Для 32-разрядной версии ОС: `C:\Program Files\1cv8\common`.
 - Для 64-разрядной версии ОС: `C:\Program Files (x86)\1cv8\common`.
- Для ОС Linux:
 - Для 32-разрядной версии ОС: `/opt/1C/v8.3/i386`.
 - Для 64-разрядной версии ОС: `/opt/1C/v8.3/x86_64`.

Программа запуска позволяет запускать все виды клиентских приложений (толстый клиент, тонкий клиент, веб-клиент) и конфигуратор.

Интерактивная программа запуска использует в своей работе некоторые компоненты системы «1С:Предприятие», поэтому последующий запуск клиентского приложения версии, равной версии интерактивной программы запуска, происходит быстрее, чем отдельный запуск исполняемого файла конкретного клиента.

Интерактивная программа запуска может быть запущена как интерактивно ([Пуск – Программы – 1С Предприятие 8 – Дополнительно – 8.3.3.100 – 1С Предприятие](#)), так и посредством программы запуска (см. [здесь](#)).

При первом запуске интерактивная программа запуска формирует единый список информационных баз, который хранится в файле `ibases.v8i` (см. [здесь](#)). В этот список попадают информационные базы от всех версий «1С:Предприятия». Перенос списка информационных баз версий 8.0 и 8.1 в данный список сопровождается вопросом. Дальнейшая актуализация списка баз не предусмотрена. В процессе первого запуска также выполняется определение расположения каталогов шаблонов конфигураций предыдущих версий и сохранение обнаруженных путей в параметре `ConfigurationTemplatesLocation` файла `1cestart.cfg` (см. [здесь](#)).

Интерактивная программа запуска может быть запущена либо без параметров, либо с указанием ссылки на конкретную информационную базу.

4.2.2.1. Запуск без параметров

В случае если интерактивная программа запуска вызвана без параметров, открывается окно выбора информационной базы (см. [здесь](#)).

После того как выбрана конкретная информационная база, интерактивная программа запуска использует следующий алгоритм:

- Если интерактивная программа запуска запущена из каталога конкретной версии без указания параметра `/AppAutoCheckVersion`, то:
 - Для запуска используются исполняемые файлы только той версии, из каталога которой запущена интерактивная программа запуска.
 - Если у запускаемой информационной базы задан автоматический выбор типа клиента, то выполняется запуск тонкого клиента с передачей ему параметра `/AppAutoCheckMode` (см. [здесь](#)).
- Если интерактивная программа запуска запущена программой запуска или в интерактивном режиме, с указанием параметра `/AppAutoCheckVersion`:
 - Происходит определение нужной версии для запуска информационной базы и выполняется поиск исполняемых файлов нужной версии (см. [здесь](#)).
 - Если конкретная версия системы не установлена на компьютере и не может быть установлена, запуск прекращается с выдачей сообщения об ошибке.
 - Затем определяется запускаемый клиент и другие параметры запуска, и происходит попытка запуска необходимого клиента с нужными параметрами из каталога версии (включая параметр `/AppAutoCheckVersion`).
 - Если в каталоге версии не обнаружен нужный клиент, запуск прекращается с выдачей сообщения об ошибке.

4.2.2.2. Запуск с параметрами

Запуск интерактивной программы запуска с параметром, указывающим на конкретную информационную базу (параметр `/IBName`), не отличается от запуска программы запуска (см. [здесь](#)).

4.2.3. Необходимый клиент конкретной версии

Конкретный клиент (толстый или тонкий) может быть запущен только из каталога конкретной версии двумя способами:

- Выбором соответствующего пункта меню, например, для запуска толстого клиента версии 8.3.3.100, необходимо выбрать следующую команду (если эта версия установлена на компьютере): **Пуск – Программы – 1С Предприятие 8 – Дополнительно – 8.3.3.100 – 1С Предприятие (толстый клиент)**.
- Запуском исполняемого файла необходимого клиента. Так, для запуска клиента из предыдущего пункта следует запустить файл `1cv8`, расположенный по следующему пути: `C:\Program Files\1cv8\8.3.3.100\bin`.

Порядок запуска тонкого клиента аналогичен вышеприведенному со следующими отличиями:

- в меню следует выбирать пункт **1С Предприятие (тонкий клиент)**;
- исполняемый файл имеет имя `1cv8c`.

Если соответствующий клиент запущен без параметров `/AppAutoCheckVersion` или `/AppAutoCheckMode`, то будет предпринята попытка выполнить запуск выбранной информационной базы выбранным клиентом конкретной версии.

4.2.4. Веб-клиент

Для запуска веб-клиента нужно запустить веб-браузер и набрать URL информационной базы. При этом веб-браузер должен быть особым образом настроен. Подробности настройки см. [здесь](#).

Информацию по настройке веб-серверов для работы с веб-клиентом см. [здесь](#).

4.2.4.1. Выбор языка интерфейса и региональных установок

Язык интерфейса веб-клиента можно указать следующими способами (в порядке повышения приоритета):

- в настройках предпочтительных языков веб-браузера;
- в командной строке (параметр `L`).

При выборе языка интерфейса выполняются следующие действия:

- При обработке запроса к ресурсу, которому соответствует информационная база (например, <http://localhost/demo>), производится выбор языка локализации:
 - При наличии в URL параметра `L` анализируется значение данного параметра. Если в результате анализа параметра язык не подобран, производится анализ заголовка `Accept-Language`.
 - При отсутствии параметра в URL производится анализ стандартного заголовка HTTP – `Accept-Language` (который содержит предпочтительные языки, установленные в браузере).
- Выбор доступного языка осуществляется из набора установленных на веб-сервере локализаций:
 - Если точного соответствия не найдено (например, в параметре указан язык `en_US`), производится усечение имени языка и выполняется повторный поиск (в примере: `en`).

• Если соответствующий язык не был найден в процессе анализа, языком по умолчанию является английский (**en**):

- Выбранный язык добавляется к базовому URL приложения (в примере получается: <http://localhost/demo/en>), и осуществляется автоматическая переадресация веб-браузера на новый URL.

Региональные установки сеанса веб-клиента (влияющие на отображение значений типа **Число** и **Дата**) можно указать следующими способами (в порядке повышения приоритета):

- в настройках предпочтительных языков веб-браузера;
- в командной строке (параметр **VL**).

Выбор региональных установок сеанса выполняется следующим образом:

- При наличии в URL параметра **VL** используются региональные установки, соответствующие локализации, код которой указан в параметре. Если в качестве значения параметра указан код несуществующей локализации, работа веб-клиента завершается с ошибкой.
- При отсутствии параметра в URL производится анализ стандартного заголовка HTTP – **Accept-Language** (который содержит предпочтительные языки, установленные в браузере).

ПРИМЕЧАНИЕ. Веб-браузер Safari не поддерживает настройку предпочтительных языков. Вместо этого используется язык интерфейса операционной системы.

4.2.4.2. Аутентификация с помощью POST-запроса

Возможны ситуации, когда необходимо запустить «1С:Предприятие», минуя стандартное окно аутентификации пользователей. Это может потребоваться, когда аутентификацию в «1С:Предприятии» необходимо сделать либо через специализированную форму (например, интегрированную в какой-либо веб-сайт), либо логин и пароль входа в информационную базу хранятся в отдельной базе данных.

Для реализации этих требований существует возможность выполнять аутентификацию сеанса веб-клиента с помощью POST-запроса к специальному ресурсу информационной базы: [e1cib/start](#). В этом случае процесс запуска можно представить следующим образом:

1. Выполняется POST-запрос с целью аутентификации клиента.
2. Если аутентификация выполнена успешно, создается сеанс от лица пользователя, переданного в POST-запросе.
3. Выполняется запуск веб-клиента, в командную строку которого передаются следующие параметры из POST-запроса: [LowClientConnectionSpeed](#), [LaunchParameter](#), [LocaleCode](#), [Zone](#).
4. Запущенный веб-клиент подключается к аутентифицированному (на шаге 2) сеансу.

СОВЕТ. Для выполнения аутентификации рекомендуется использовать протокол HTTPS.

В запросе передаются следующие параметры:

Usr **обязательный**

Имя пользователя.

Pwd **необязательный**

Пароль пользователя.

Значение по умолчанию – пустая строка.

LowClientConnectionSpeed **необязательный**

Скорость соединения.

Возможные значения:

- **on** – низкая скорость соединения.
- **off** – нормальная скорость соединения (значение по умолчанию).

LaunchParameter **необязательный**

Параметры, которые необходимо передать в прикладное решение (аналог параметра **c** командной строки веб-клиента).

Значение по умолчанию – пустая строка.

SystemLanguage **необязательный**

Язык интерфейса. Если не задано – определение языка интерфейса и региональных установок, см. [здесь](#).

LocaleCode **необязательный**

Язык интерфейса. Если не задано – определение языка интерфейса и региональных установок, см. [здесь](#).

Zone *необязательный*

Значения разделителей. Подробную информацию о задании значений разделителей в веб-клиенте можно получить в книге «1С:Предприятие 8.3. Руководство разработчика».

AuthFailHandling *необязательный*

Определяет поведение системы в случае ошибки аутентификации. Возможные значения:

- **error** – возвращает код ошибки (код ошибки – 402) и текст сообщения об ошибке.
- **start** – выполняется запуск веб-клиента с запросом аутентификации средствами «1С:Предприятия».
- **redirect** – осуществляется переход на URL, заданный параметром **AuthFailRedirectURL**.

Значение по умолчанию – **error**.

AuthFailRedirectURL *необязательный*

Содержит URL, на который следует перейти в случае ошибки аутентификации, если параметр **AuthFailHandling** установлен в значение **redirect**. URL должен быть абсолютным.

ПРИМЕЧАНИЕ. Параметры, переданные в теле запроса, имеют приоритет над параметрами командной строки запуска веб-клиента.

Пример:

Ниже приведен пример HTML-страницы, который демонстрирует работу собственной формы аутентификации для информационной базы, расположенной по адресу <http://localhost/demoapp>.

[Копировать в буфер обмена](#)

```
<HTML xmlns="http://www.w3.org/1999/xhtml"><HEAD>
<META http-equiv="Content-Type" content="text/html; charset=utf-8" />
<BODY>
<FORM action="http://localhost/demoapp/elcib/start" method="post">
  Пользователь: <INPUT id="usr" name="usr" /><BR />
  Пароль: <INPUT id="pwd" type="password" value="" name="pwd" />
  <BR />Низкая скорость: <INPUT id="lowclientconnections speed" type="checkbox" name="lowclientconnections speed" /><BR />
  Параметр запуска: <INPUT id="launchparameter" name="launchparameter" /><BR />
  Язык интерфейса: <SELECT id="systemlanguage" name="systemlanguage">
    <OPTION value="ru" selected="">Русский</OPTION>
    <OPTION value="en">Английский</OPTION>
  </SELECT><BR />
  Код локализации сеанса: <SELECT id="localecode" name="localecode">
    <OPTION value="ru" selected="">Русский</OPTION>
    <OPTION value="en">Английский</OPTION>
  </SELECT><BR />
  Область данных: <INPUT id="zone" name="zone" />
  <INPUT id="authfailhandling" type="hidden" value="error" name="authfailhandling" />
  <P><INPUT type="submit" value="OK" /> </P>
</FORM>
</BODY>
</HTML>
```

В результате будет показана следующая форма аутентификации:

Пользователь:

Пароль:

Низкая скорость:

Параметр запуска:

Язык интерфейса:

Код локализации сеанса:

Область данных:

Рис. 17. Форма POST-запроса

4.2.5. Специальные параметры запуска

4.2.5.1. Параметр /IBName

Параметр **/IBName** позволяет указать наименование запускаемой информационной базы. В этом случае программа запуска (или исполняемый файл конкретного клиента) выполнит поиск указанной информационной базы в списке информационных баз.

Если баз с указанным именем более одной, запуск будет прекращен с выдачей сообщения об ошибке.

Если база будет обнаружена, произойдет попытка запуска этой информационной базы в соответствии с параметрами и выбранной программой запуска (или исполняемым файлом конкретного клиента).

Примечание. Если в наименовании информационной базы есть кавычки, они должны быть удвоены при указании имени в параметре: База ""СтройТоргВсе"".

4.2.5.2. Параметр /AppAutoCheckVersion

С помощью данного параметра имеется возможность автоматически «подобрать» для запуска ту версию системы, которая необходима для работы с выбранной информационной базой.

В случае указания в строке запуска параметра /AppAutoCheckVersion система выполняется следующий алгоритм:

- Определяется версия запускаемой информационной базы.
- Если это версия 8.1 или 8.0, определяется местоположение запускаемых файлов требуемой версии и выполняется запуск необходимого исполняемого файла нужной версии. При этом для «1С:Предприятия 8.0» выполняется замена параметра командной строки `DESIGNER` на `CONFIG` (для обеспечения совместимости).
- При использовании версии 8.2 и выше:
 - Если указан полный номер версии, выполняется поиск необходимой версии (параметры `InstalledLocation` конфигурационных файлов). Если необходимая версия не установлена на компьютер, то производится поиск дистрибутива нужной версии (параметры `DistributiveLocation` конфигурационных файлов). Если версия найдена, она устанавливается; если нет, запуск прекращается с выдачей сообщения об ошибке.
 - Если указан неполный номер версии, то производится попытка получить необходимый номер среди используемых версий (параметр `DefaultVersion` конфигурационных файлов). Если установить полный номер версии не удалось, то ищется максимальный номер установленной версии (параметры `InstalledLocation` конфигурационных файлов) и максимальный номер версии, доступной к установке (параметры `DistributiveLocation` конфигурационных файлов). Если номер версии, доступной для установки, больше номера версии, установленной на компьютере, происходит установка версии с большим номером.
- Затем производится запуск интерактивной программы запуска необходимой версии с параметром /AppAutoCheckVersion.

4.2.5.3. Параметр /AppAutoCheckMode

С помощью данного параметра имеется возможность автоматически «подобрать» для запуска тот вариант клиентского приложения, который необходим для работы.

В случае указания в строке запуска приложения параметра /AppAutoCheckMode система выполняет следующее:

- Определяется режим запуска конкретного пользователя.
- Определяет основной режим запуска запускаемой информационной базы.
- Если определенный режим запуска не соответствует запускаемому клиенту и у текущего пользователя имеется право на запуск требуемого клиентского приложения, то выполняется перезапуск необходимого клиентского приложения той же версии. В противном случае продолжается загрузка запускаемого клиентского приложения.

4.2.5.4. Параметр /url

Данный параметр позволяет выполнять переход по навигационной ссылке в запущенном клиентском приложении (при соблюдении некоторых условий). Такой параметр доступен у программы запуска, тонкого и толстого клиентского приложения.

Общая схема работы с данным параметром:

- Если значение параметра содержит внешнюю ссылку (содержит информацию об адресе информационной базы), то на текущем компьютере выполняется поиск клиентского приложения, запущенного для работы с требуемой информационной базой, и в нем выполняется попытка открыть объект, описываемый навигационной ссылкой. Если требуемая информационная база не обслуживается ни одним клиентским приложением, то выполняется запуск нового клиентского приложения с указанием параметра командой строки `/url`.
- Если значением параметра является внутренняя ссылка, то значение параметра игнорируется и выполняется обычный запуск клиентского приложения.

Рассмотрим более подробно действия, выполняемые при поиске клиентского приложения. Поиск приложения выполняется следующим образом:

- Выполняется поиск запущенного клиентского приложения, которое обслуживает информационную базу с указанной (в значении параметра) строкой соединения. При поиске не учитывается регистр имени хоста и информационной базы. При поиске не выполняется преобразование DNS-имени в IP-адрес и обратно. Это означает, что если клиентское приложение использует в строке соединения DNS-адрес, а во внешней ссылке используется IP-адрес, то клиентское приложение не будет обнаружено.
- Если приложение найдено и в приложении не открыто модального окна или окна, блокирующего весь интерфейс, то приложение активизируется и выполняется попытка перехода по внутренней ссылке, полученной из внешней ссылки, которая передана в качестве значения параметра командной строки `/url` клиентского приложения.
- Если приложение найдено и в приложении открыто модальное окно или окно, блокирующее весь интерфейс, то найденное приложение игнорируется и система продолжает поиск.
- Если приложение не найдено, то выполняется запуск нового клиентского приложения, которому в параметре `/url` передается

внутренняя навигационная ссылка, полученная из исходной ссылки.

- Если навигационная ссылка, переданная программе запуска или клиентскому приложению, содержит только адрес информационной базы (например, `e1c://host/ib-name`), то после открытия клиентского приложения больше никаких действий не выполняется.

При переходе по внутренней навигационной ссылке выполняются следующие действия:

- Попытка перехода предпринимается после вызова обработчика `ПриНачалеРаботыСистемы`.
- Если клиентское приложение работает в режиме `Формы в отдельных окнах` и внутренняя ссылка является ссылкой на точку навигации, то вместо открытия рабочего стола выполняется переход по ссылке. В противном случае открытие формы происходит после открытия рабочего стола.
- При возникновении ошибки выводится диагностическое сообщение, и приложение продолжает работу.

Если в командной строке запуска указывается несколько параметров, то:

- При указании параметра `/Execute` параметр `/url` игнорируется;
- Параметр `/url`, в котором указана абсолютная навигационная ссылка, приводит к тому, что параметры соединения с информационной базой, указанные в других параметрах, игнорируются. Игнорируются следующие параметры командной строки: `/F`, `/S`, `/WS`, `/IBName`. Для параметра `/IBConnection` игнорируются части строки соединения, описывающие информационную базу.

Для перехода можно использовать внешние навигационные ссылки, использующие следующие схемы:

- `e1c:` – толстый и тонкий клиенты;
- `http:` или `https:` – только тонкий клиент.

Выполнить действие, эквивалентное использованию параметра `/url`, можно с помощью:

- метода встроенного языка `ПерейтиПоНавигационнойСсылке()`;
- с помощью специального диалога из окна со списком информационных баз;
- с помощью стандартной формы полнотекстового поиска (только для схемы `e1c:`).

4.2.6. Варианты подключения к информационной базе

Существует несколько способов расположения информационной базы и способов подключения к ней (выбирается в диалоге добавления информационной базы, см. [здесь](#)):

- Информационная база расположена на локальном компьютере или на компьютере в локальной сети.

Используется тонким и толстым клиентами в файловом варианте.

При работе тонкого клиента в файловом варианте работы на компьютере, где запущен сам тонкий клиент, организуется специализированная среда. В рамках этой специализированной среды выполняются:

- загрузка необходимых для работы системы серверных компонентов;
- загрузка прикладной конфигурации;
- другие действия, необходимые для организации нормальной работы системы с информационной базой.

При этом взаимодействие между тонким клиентом и этой специализированной средой выполняется по тем же протоколам, что и в случае работы в клиент-серверном варианте или через веб-сервер. Таким образом, с точки зрения тонкого клиента данная среда выступает в роли сервера. С точки зрения операционной системы данная специализированная среда не выделена в отдельный процесс и выполняется в рамках процесса тонкого клиента.

- Информационная база расположена на сервере «1С:Предприятия».

Используется тонким и толстым клиентами в клиент-серверном варианте.

- Информационная база расположена на веб-сервере.

Используется тонким клиентом и веб-клиентом в файловом или клиент-серверном варианте.

Для подключения через веб-сервер необходимо установить и настроить соответствующим образом веб-сервер. Описание настроек различных веб-серверов см. [здесь](#).

В качестве строки соединения с информационной базой при подключении через веб-сервер нужно указать URL, например, следующего вида <http://MyServer/DemoBase>.

4.2.7. Выбор информационной базы

Следующий этап запуска системы «1С:Предприятие» – выбор информационной базы. Для этого служит выдаваемое на экран окно `Запуск 1С:Предприятия`.

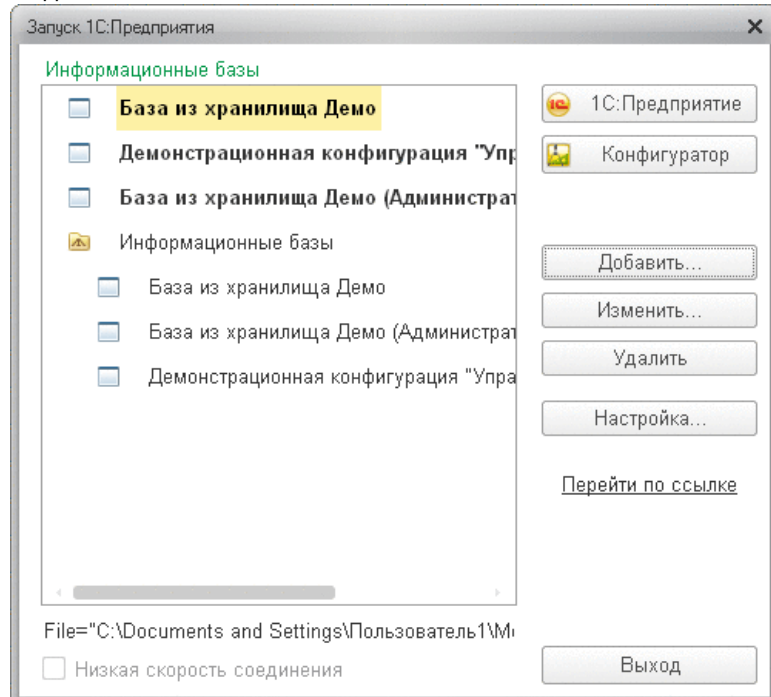


Рис. 18. Запуск «1С:Предприятия»

В списке **Информационные базы** содержится список информационных баз. Каждая строка списка связана с каким-либо каталогом, в котором находятся файлы информационной базы системы «1С:Предприятие» (для файлового варианта), или сервером и информационной базой на сервере (для клиент-серверного варианта).

В этом списке должна быть выбрана одна из информационных баз. Для выбора необходимо щелкнуть левой кнопкой мыши на названии нужной информационной базы.

Кнопки **Изменить**, **Добавить** и **Удалить** служат для управления списком информационных баз системы «1С:Предприятие» (можно использовать горячие клавиши **F2**, **Ins** и **Del**). Назначение этих кнопок см. [здесь](#).

Размеры окна можно изменить. Положение окна на экране и его размер запоминаются до следующего сеанса.

После того как установлены все необходимые параметры запуска системы «1С:Предприятие», необходимо нажать кнопку **1С:Предприятие** для запуска в режиме 1С:Предприятие или **Конфигуратор** для запуска в режиме Конфигуратор. Нажатие кнопки **Выход** позволяет отказаться от запуска.

Гиперссылка **Перейти по ссылке** предназначена для запуска клиентского приложения при наличии внешней навигационной ссылки на объект прикладного решения. При нажатии на гиперссылку открывается диалог, в котором нужно указать навигационную ссылку и нажать кнопку **Перейти**. Подробнее о действиях, которые выполняются при запуске клиентского приложения с использованием внешней навигационной ссылки, см. [здесь](#).

4.2.8. Несовпадение версии клиента и сервера

В случае клиент-серверной системы, версия клиентского приложения может отличаться от версии серверного приложения. Работать в этом случае невозможно – версии клиента и сервера должны в точности совпадать.

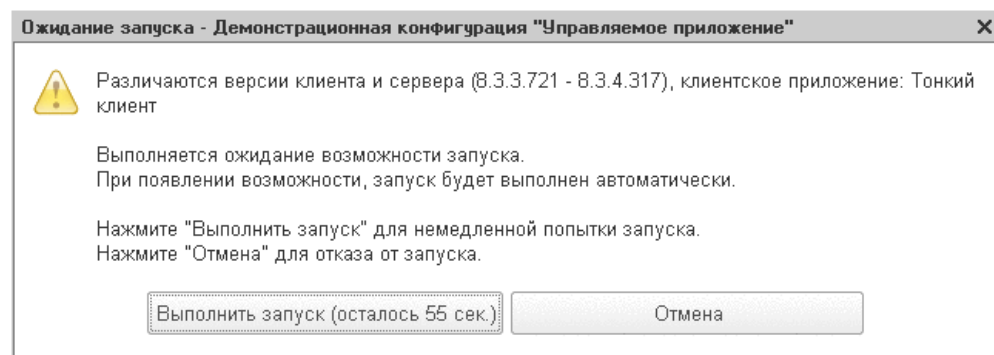


Рис. 19. Несовпадение версии клиента и сервера

Если тонкий клиент (**1cv8c**) запущен с ключом `/AppAutoCheckVersion`, то в случае несовпадения версии клиента и сервера выполняется попытка поиска и установки версии, которая требуется для работы с сервером. Поиск дистрибутива клиентского приложения выполняется в следующем порядке:

- В каталоге, куда установлено «1С:Предприятие» (свойство `InstalledLocation` файлов `1cestart.cfg` и `1cescmn.cfg`);
- В каталогах, указанных как место расположения дистрибутивных файлов новых версий (свойство `DistributiveLocation` файлов `1cestart.cfg` и `1cescmn.cfg`, а также свойство `CommonCfgLocation` файла `1cestart.cfg`);

- По URL, который возвращается в серверном исключении о несовпадении версий клиентского и серверного приложения (параметр `PublishDistributiveLocation` файла `conf.cfg` и атрибут `pubdst` элемента `ws` файла `default.vrd`);

- С помощью интернет-сервисов получения дистрибутива клиентского приложения.

При этом надо помнить, что:

- Поиск и получение дистрибутива с помощью интернет-сервисов осуществляется тонким клиентом, только в том случае, если подключение к информационной базе осуществляется по http-соединению.

- Получение дистрибутива через URL, указанный в исключении о несовпадении версии клиента и сервера, осуществляется тонким клиентом при любом клиент-серверном подключении.

4.2.9. Аутентификация пользователей

Если для выбранной информационной базы существует список пользователей, которым разрешена работа с ней (создание и редактирование такого списка выполняется в конфигураторе системы «1С:Предприятие»), на экран будет выдан диалог [Аутентификация пользователя](#).

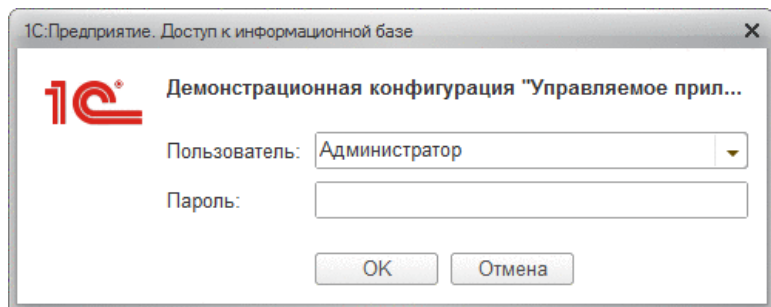


Рис. 20. Аутентификация пользователя

В этом диалоге необходимо указать имя пользователя, что можно осуществить несколькими способами:

- Щелкнуть мышью в поле [Пользователь](#) и выбрать имя из списка.
- Ввести имя пользователя в поле ввода [Пользователь](#), если список очень большой или в настройках пользователя не установлено свойство [Показывать в списке выбора](#) (см. [здесь](#)).

Если пользователю назначен пароль, его следует ввести в поле [Пароль](#). После указания имени и пароля пользователя процесс запуска продолжится, если нажать кнопку [OK](#). Нажатие кнопки [Отмена](#) позволяет отказаться от запуска.

4.2.10. Использование клиентских сертификатов

4.2.10.1. Общая информация

При работе через публичные каналы связи (Интернет) большое значение приобретает возможность защиты информации, передаваемой по этому каналу, от перехвата и подмены. Рассмотрим организацию такого соединения в системе на базе «1С:Предприятие».

Рассмотрим общую схему организации безопасного соединения. В ее основе лежит инфраструктура открытых ключей (PKI), которая связывает открытые ключи с личностью пользователя посредством удостоверяющего центра.

Чтобы получить представление о работе этой инфраструктуры, обратимся к простому примеру. Представим, что мы находимся в некоем мире, где любой человек может проверить удостоверение другого человека в ведомстве, которое это удостоверение выдало.

В этом мире один человек ([Прохожий](#)) встречает другого человека ([Полицейского](#)), который хочет удостовериться в том, что человек перед ним – действительно [Прохожий](#). Для этого [Полицейский](#) просит у [Прохожего](#) его паспорт. Прежде чем предъявлять свои документы, [Прохожий](#) хочет убедиться, что перед ним реальный [Полицейский](#). Он просит у [Полицейского](#) его удостоверение личности, связывается с [Министерством Управления Полиции](#) и по номеру проверяет, что человек перед ним – действительно тот, за кого он себя выдает, т. е. [Полицейский](#). После успешной процедуры аутентификации [Прохожий](#) отдает свой паспорт [Полицейскому](#). В паспорте написано, что он выдан [Министерством Выдачи Документов](#) и указан номер паспорта. [Полицейский](#) связывается с [Министерством](#) и с помощью номера паспорта удостоверяется в том, что человек перед ним – действительно [Прохожий](#).

Но если [Прохожий](#) окажется за пределами своей страны, то описанный выше алгоритм аутентификации не сработает, т. к. [Полицейский](#) другой страны ничего не знает про [Министерство Выдачи Документов](#). Поэтому [Прохожего](#) задержат до выяснения личности другим путем.

Теперь представим эту простую схему с точки зрения объектов PKI и сетевой инфраструктуры. **Клиентское приложение** «1С:Предприятия» выступает в роли [Прохожего](#). **Веб-сервер**, с помощью которого клиентское приложение хочет получить доступ к информационной базе, выступает в роли [Полицейского](#). [Министерство Выдачи Документов](#) и [Министерство Управления Полицией](#) играют роль [Удостоверяющих Центров](#). **Сертификат**, используемый при установке HTTPS-соединения, представлен в виде паспорта [Прохожего](#) и удостоверения личности [Полицейского](#).

Теперь вся схема выглядит следующим образом: при попытке клиентского приложения подключиться к веб-серверу, происходит проверка клиентским приложением сертификата сервера. Проверка происходит с помощью удостоверяющего центра, который указан в сертификате веб-сервера (если таковой центр присутствует в списке корневых удостоверяющих центров на компьютере, где установлено клиентское приложение). Если проверка прошла успешно, то клиентское приложение предоставляет свой сертификат (клиентский сертификат) для проверки веб-серверу. Сервер делает это с помощью своего списка корневых удостоверяющих центров. Если проверка прошла успешно – клиентское приложение и веб-сервер устанавливают защищенное соединение (HTTPS-соединение). При этом клиентское приложение шифрует передаваемые данные с помощью открытого ключа сервера (и расшифровывает данные,

полученные от сервера), а сервер – шифрует и расшифровывает данные с помощью своего закрытого ключа. Очевидно, что закрытые ключ клиентского приложения и веб-сервера – не совпадают и неизвестны сторонам.

Выше приведена общая схема установки защищенного соединения. Более подробно эти схемы будут описаны в следующем разделе.

4.2.10.2. Схемы установки защищенного соединения

Защищенное соединение может быть установлено между тонким клиентом или веб-клиентом и веб-сервером, посредством которого выполняется подключение к информационной базе. Существует несколько схем установки такого соединения (в зависимости от наличия тех или иных сертификатов на обеих сторонах соединения), которые будут рассмотрены ниже. Нужно помнить, что при любой установке HTTPS-соединения, оно будет зашифровано.

Сервер	Клиент	Особенности
Сертификат+ Корневые-	Сертификат- Корневые-	Сертификаты сервера и клиента не проверены. До версии 8.3.3 доступен только этот режим
Сертификат+ Корневые-	Сертификат- Корневые+	Проверен только сертификат сервера. Сертификата клиента не проверяется
Сертификат+ Корневые+	Сертификат- Корневые- или Сертификат- Корневые+	Такая ситуация не поддерживается
Сертификат+ Корневые+	Сертификат+ Корневые-	Сертификат сервера не проверен, сертификат клиента проверен
Сертификат+ Корневые+	Сертификат+ Корневые+	Проверяются сертификаты обеих сторон

В таблице использованы следующие термины:

- **Сертификат** – означает наличие (**Сертификат+**) или отсутствие (**Сертификат-**) соответствующего сертификата:
 - Для сервера – серверного сертификата.
 - Для клиента – клиентского сертификата.
- **Корневые** – означает наличие (**Корневые+**) или отсутствие (**Корневые-**) списка сертификатов удостоверяющих центров (УЦ или СА), с помощью которых можно проверить предъявленный сертификат. Список удостоверяющих центров должен позволять проверить сертификат, предоставленный клиентским приложением или веб-сервером.

В случае веб-клиента, наличие или отсутствие сертификата или списка корневых сертификатов, определяется установкой сертификатов в хранилище сертификатов, с которым работает используемый веб-браузер.

Для тонкого клиента сертификат (и списки корневых сертификатов) можно указать с помощью параметров командной строки запуска или с помощью параметров запуска информационной базы (см. [здесь](#)).

4.2.10.3. Источники и форматы сертификатов

В качестве источников сертификатов могут выступать следующие хранилища:

- Системное хранилище сертификатов – для ОС Windows.
- Хранилище сертификатов Network Security Services (NSS, <http://www.mozilla.org/projects/security/pki/nss/>) – для ОС Linux или Windows.
- Файловые сертификаты – для ОС Linux или Windows.

Для использования сертификатов хранилищ NSS необходимо наличие установленных в системе библиотек NSS:

- Для ОС Windows – `nspr4.dll` (или `libnspr4.dll`), `ssl3.dll`, `nss3.dll` и другие связанные библиотеки. Необходимо добавить в переменную окружения `PATH` каталог, содержащий скомпилированную версию библиотек NSS и NSPR.
- Для ОС Linux – `libnspr4.so`, `libssl3.so`, `libnss3.so` и другие связанные библиотеки. Если в качестве источника сертификатов выбрано хранилище сертификатов NSS, а необходимые библиотеки NSS отсутствуют, то соединение может быть установлено только с сервером, не требующим клиентский сертификат и без проверки серверного сертификата. При этом канал все равно будет шифроваться ключами серверного сертификата.

В случае отсутствия вышеперечисленных библиотек, установка соединения с использованием сертификатов хранилища NSS не возможна. При использовании ОС Linux, рекомендуется для операций с хранилищем NSS использовать утилиту `certutil` (https://developer.mozilla.org/en-US/docs/NSS/tools/NSS_Tools_certutil).

Допустимые форматы файловых сертификатов:

- **PEM** (base-64 encoded X.509) – зашифрованные ключи и сертификаты стандарта X.509 в текстовом формате. Данные сертификатов и ключей кодируются в кодировке base-64. Закрытые ключи сертификатов защищены паролем. Данный формат файлов сертификатов используется по умолчанию, например, веб-сервером Apache. **Ошибка! Закладка не определена.** Если закрытый ключ клиентского сертификата хранится в отдельном файле, то необходимо добавить содержимое этого файла к файлу клиентского сертификата.
- **P12/PFX** (PKCS#12) – зашифрованные ключи и сертификаты стандарта PKCS#12. Файл может быть защищен паролем. Это основной формат экспорта и импорта системных хранилищ сертификатов для ОС Windows и хранилища сертификатов NSS. Используется, например, веб-сервером Microsoft Internet Information Services. Файл клиентского сертификата должен содержать его закрытый ключ.

Формат файла выбирается по его расширению:

- *.p12, *.pfx – формат файла P12,
- *.pem – формат файла PEM,
- по умолчанию выбирается формат файла PEM.

4.3. Перезапуск системы

В некоторых случаях открытие информационной базы невозможно. Система уведомляет об этом пользователя и предлагает повторить попытку соединения с информационной базой через 60 секунд.

К таким случаям относятся:

- Конфигурация уже открыта в режиме Конфигуратор (при попытке запуска в режиме Конфигуратор).
- Для информационной базы установлен монопольный режим работы.
- Различаются версии клиентского приложения и сервера «1С:Предприятия».
- Не обнаружен сервер «1С:Предприятия».
- Не обнаружен сервер баз данных.
- Административно установлен запрет соединения с информационной базой.

В таких случаях на экран выводится окно, в котором сообщается о причине (в приведенном на рисунке ниже примере это факт открытия данной информационной базы конфигуратором), и предлагается выбрать автоматический перезапуск системы через 1 минуту или отказаться от запуска.

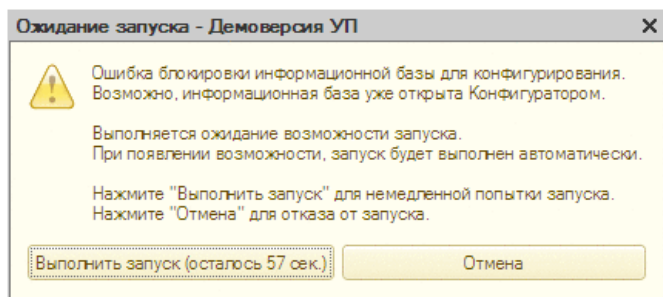


Рис. 21. Ожидание перезапуска

Перезапуск конфигулятора также предлагается после выполнения загрузки информационной базы и в случае динамического обновления информационной базы в клиент-серверном варианте работы.

При работе в режиме 1С:Предприятие в случае возникновения критической ошибки система предлагает осуществить перезапуск с теми же параметрами текущего пользователя.

Глава 5. Ведение списка информационных баз

Управляющие элементы, расположенные в окне [Запуск 1С:Предприятия](#), позволяют управлять списком информационных баз: добавлять новые базы и группы баз, перемещать базы из одной группы в другую, изменять свойства существующих информационных баз и удалять из списка ненужные информационные базы.

Список информационных баз представлен в виде списка (используется по умолчанию) или в виде дерева. Выбор режима осуществляется в диалоге настройки запуска.

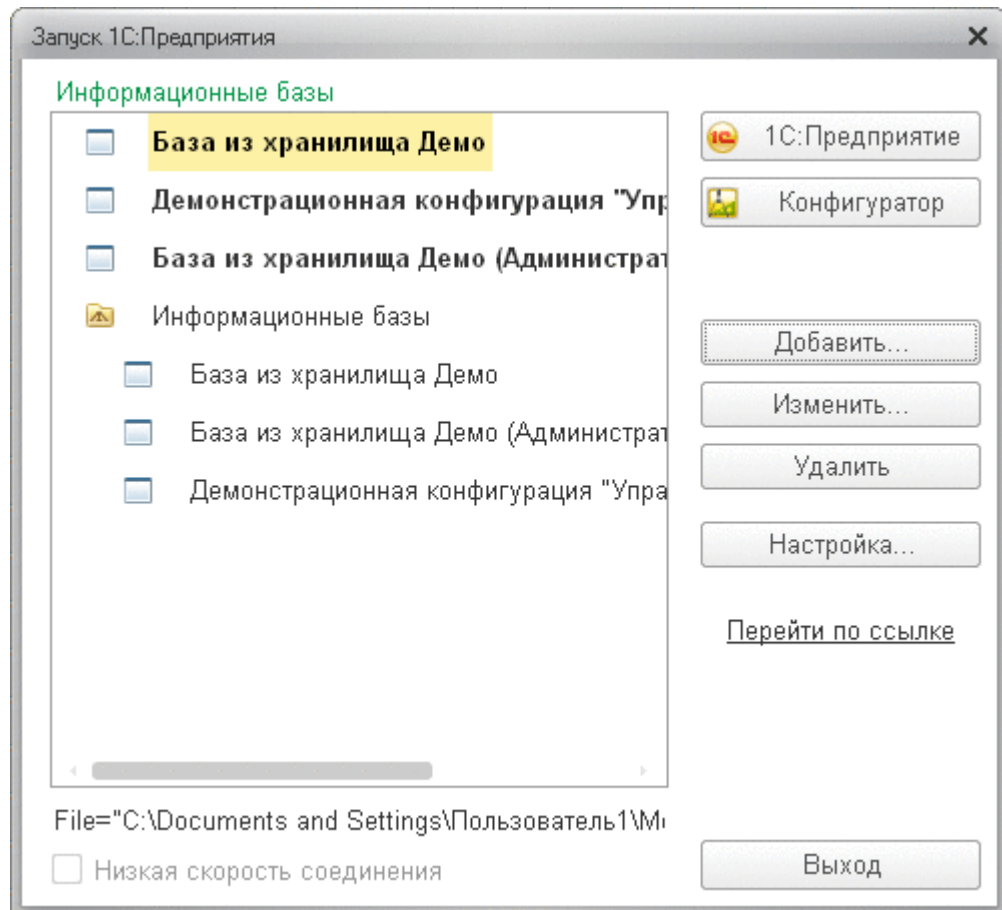


Рис. 22. Запуск «1С:Предприятия»

5.1. Добавление информационной базы

5.1.1. Добавление новой информационной базы

Для добавления в список новой информационной базы следует нажать кнопку [Добавить...](#) На экран будет выдан диалог для выбора режима добавления.

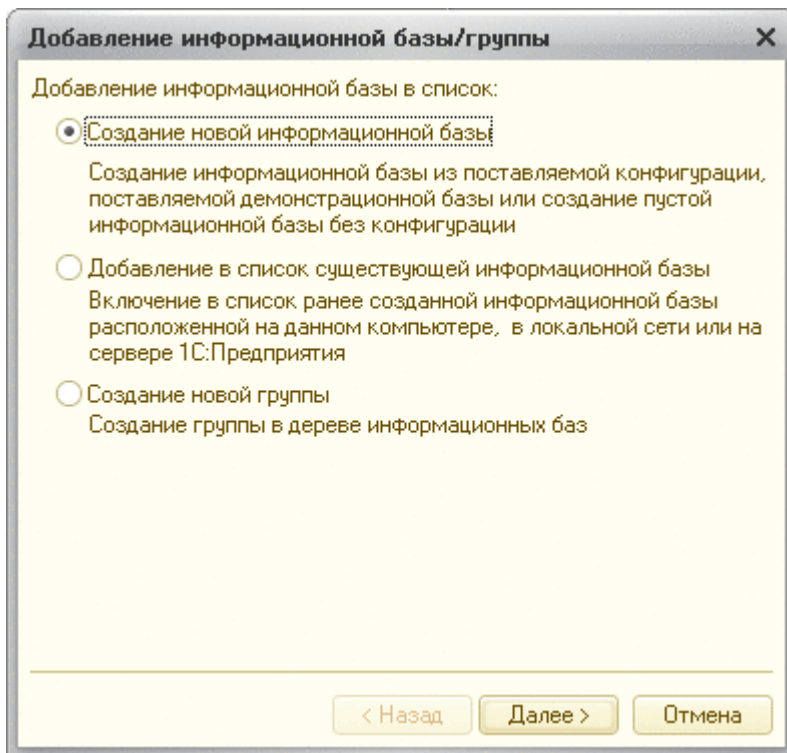


Рис. 23. Выбор режима добавления в список информационных баз

В режиме [Создание новой информационной базы](#) система «1С:Предприятие» предоставляет возможность создания информационной базы на основании базы-шаблона или создания пустой базы.

После нажатия кнопки [Далее>](#) на экран выводится диалог выбора.

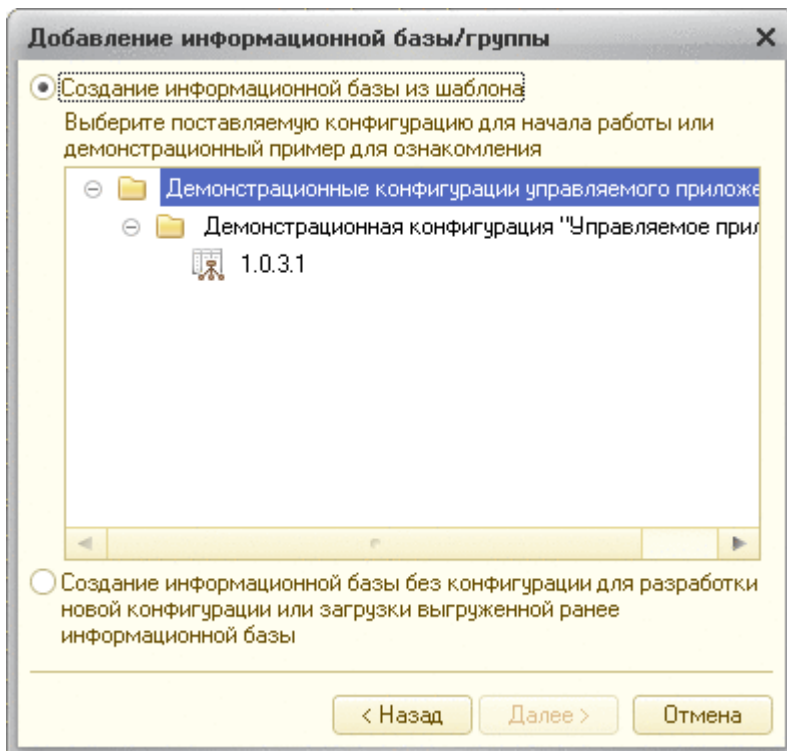


Рис. 24. Выбор добавляемой конфигурации

Если выбран способ создания информационной базы из шаблона, то в списке шаблонов нужно выбрать исходный шаблон.

На следующем шаге будет предложено указать название информационной базы и выбрать тип расположения информационной базы.

Наименование информационной базы представляет собой произвольную строку символов. Длина

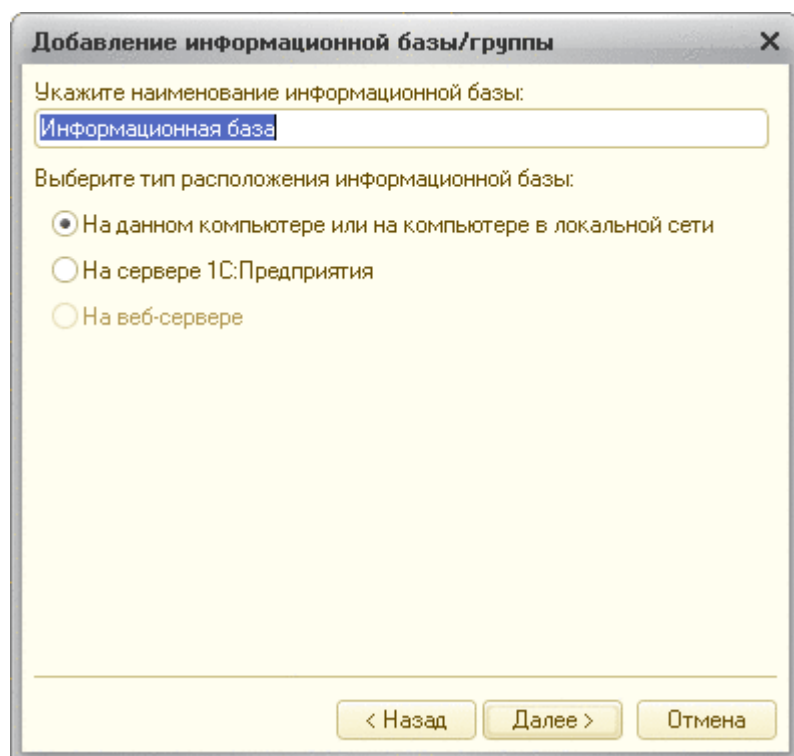
наименования информационной базы ограничена 255 символами, поэтому можно присваивать информационным базам смысловые имена. Наименование информационной базы должно быть уникально во всем списке информационных баз.

ПРИМЕЧАНИЕ. Разрешается создавать несколько информационных баз с одинаковой строкой соединения с базой данных. Это может понадобиться в случае, если необходимо осуществлять доступ к одной информационной базе с помощью различных клиентов, не изменяя свойства самой информационной базы.

Создание файлового варианта информационной базы рассмотрено в следующем разделе, а описание создания информационной базы, расположенной на сервере «1С:Предприятия», – см. [здесь](#).

5.1.1.1. Создание в файловом варианте

Для создания информационной базы в файловом варианте необходимо выбрать тип расположения информационной базы, как на [рис. 25](#).



Добавление информационной базы/группы

Укажите наименование информационной базы:

Информационная база

Выберите тип расположения информационной базы:

На данном компьютере или на компьютере в локальной сети

На сервере 1С:Предприятия

На веб-сервере

< Назад Далее > Отмена

Рис. 25. Добавление файлового варианта информационной базы

Следует выбрать каталог, в котором будет находиться информационная база. Если будет указано имя несуществующего каталога, то при запуске системы «1С:Предприятие» он будет автоматически создан.

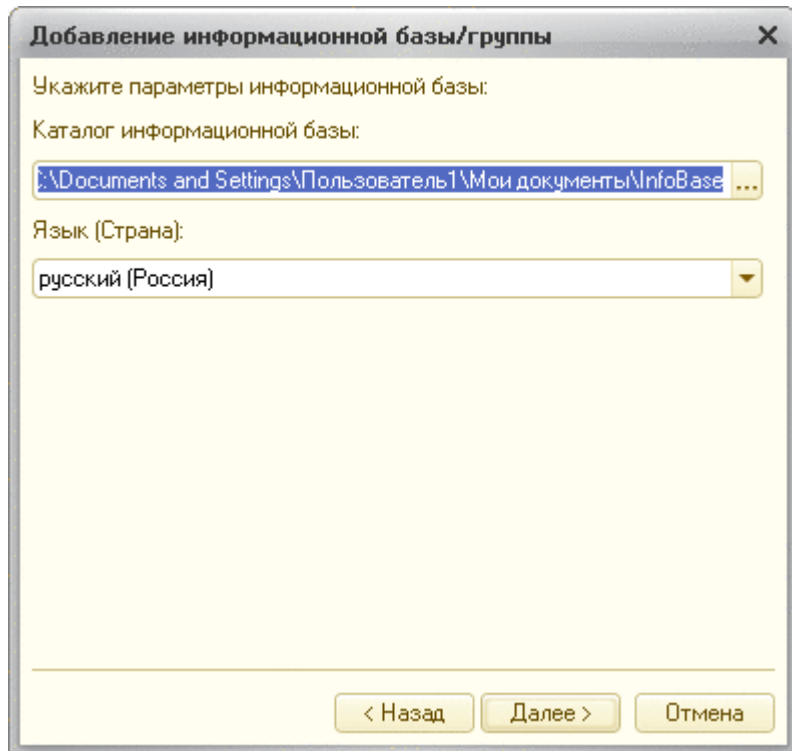


Рис. 26. Выбор каталога размещения и языка информационной базы

Нажатием кнопки выбора можно открыть стандартный диалог выбора существующего каталога.

ПРИМЕЧАНИЕ. Имя каталога с информационной базой должно удовлетворять требованиям RFC 2396, раздел 2.4.3 Excluded US-ASCII Characters (<http://tools.ietf.org/html/rfc2396.html>). В имени каталога нельзя использовать следующие символы: «<», «>», «#», «%», «"», «{», «}», «[», «]», «|», «\», «^», «`», а также символы с кодами от 0 до 31 и 127.

Если выбран режим создания пустой информационной базы, то в поле **Язык** выбирается язык, по правилам которого будут храниться и сортироваться данные в информационной базе. Если выбран шаблон, то поле для выбора языка может не выводиться.

После нажатия кнопки **Далее >** будет открыта страница задания параметров запуска (см. [здесь](#)).

Если была создана пустая информационная база и в указанном каталоге отсутствуют файлы конфигурации, то при выборе режима запуска **1С:Предприятие** также будет выводиться окно выбора режима создания информационной базы. Если выбрать режим использования шаблона, то будет создана информационная база по данному шаблону.

5.1.1.2. Создание в клиент-серверном варианте

Для создания информационной базы в клиент-серверном варианте необходимо выбрать тип расположения информационной базы, как на [рис. 27](#).

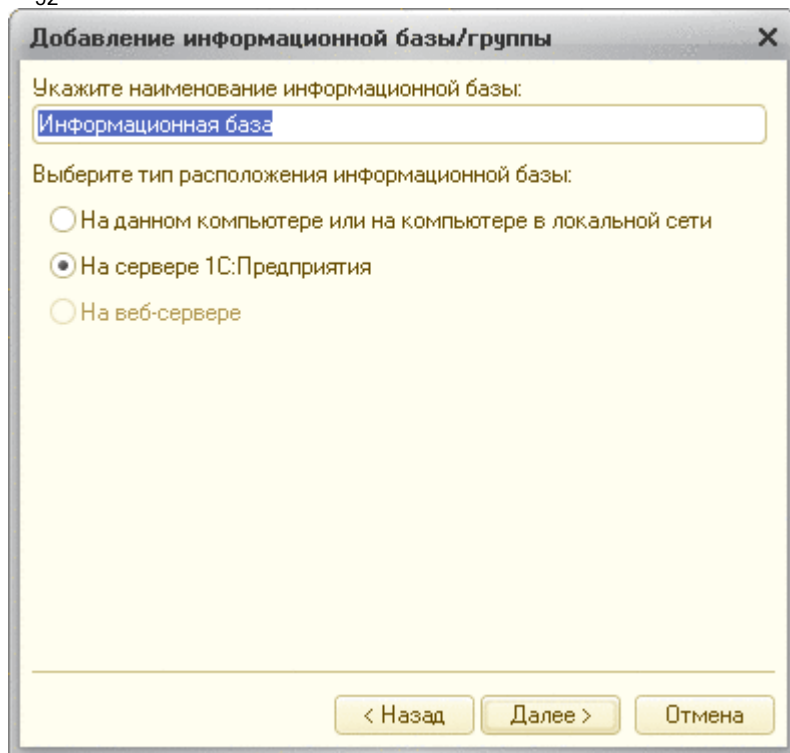


Рис. 27. Добавление информационной базы на сервере

Информационная база в клиент-серверном варианте идентифицируется двумя параметрами:

- адресом кластера серверов «1С:Предприятия»;
- именем информационной базы.

Как было отмечено выше, адрес кластера серверов «1С:Предприятия» состоит из имени центрального сервера и номера сетевого порта, через который работает менеджер кластера (например, [Test_Server:1541](#)). Если менеджер кластера использует сетевой порт, назначаемый по умолчанию (1541), то достаточно указать только имя сервера. Имя информационной базы уникально в пределах кластера серверов «1С:Предприятия».

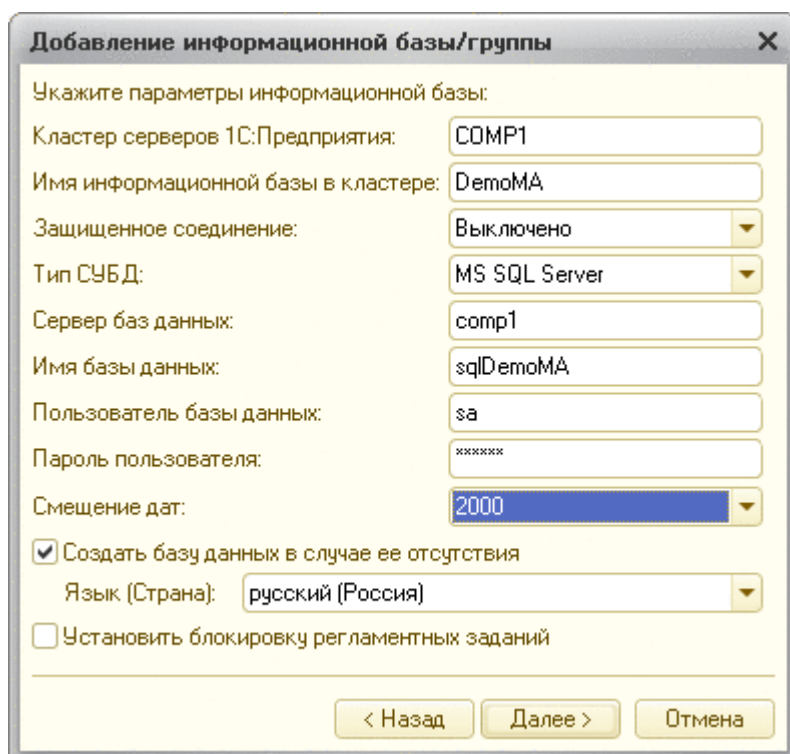


Рис. 28. Создание новой базы

ПРИМЕЧАНИЕ 1. Если используется резервирование кластеров (см. книгу «1С:Предприятие 8.3. Клиент-серверный вариант. Руководство администратора»), то список резервирования может задаваться непосредственно в поле **Кластер серверов 1С:Предприятия:** в формате **Server1, Server2:Port, Server3**. В таком виде его удобнее использовать, например, при создании списка общих информационных баз.

ПРИМЕЧАНИЕ 2. Имя базы данных СУБД должно удовлетворять требованиям **RFC 2396**, раздел **2.4.3 Excluded US-ASCII Characters** (<http://tools.ietf.org/html/rfc2396.html>). В имени базы данных нельзя использовать следующие символы: «<», «>», «#», «%», «"», «{», «}», «[», «]», «|», «\», «^», «`», а также символы с кодами от 0 до 31 и 127. Кроме того, эти требования могут конкретизироваться конкретной СУБД.

Характеристики информации, вводимые в данном окне и необходимые для создания информационной базы, зависят от используемой СУБД. Дальнейшая информация приведена в разрезе конкретной СУБД.

5.1.1.2.1. Microsoft SQL Server

- **Тип СУБД:** Microsoft SQL Server.
- **Сервер баз данных:** имя сервера. Может задаваться именем компьютера (если на компьютере установлен один экземпляр сервера) или именем конкретного экземпляра (если установлено несколько экземпляров). Например, `Server/instance`.

Если сервер «1С:Предприятия» и Microsoft SQL Server расположены на одном компьютере, и для Microsoft SQL Server установлен Native Client (собственный клиент), то имеется возможность использовать для связи серверов протокол **SHARED MEMORY**. Для этого необходимо перед именем Microsoft SQL Server указать префикс `lpc:`. В этом случае имя сервера СУБД будет выглядеть следующим образом: `lpc:Server/instance`.

- **Имя базы данных:** первым символом имени базы данных может быть буква латинского алфавита, а также символ «_». Последующими символами могут быть буквы латинского алфавита, цифры, а также символы «_», «\$». Длина имени ограничена 63 символами. В имени недопустимы пробелы. Имя не может быть зарезервированным словом языка запросов сервера баз данных.
- **Пользователь базы данных:** имя пользователя сервера баз данных, от лица которого будет осуществляться доступ к базе данных. Существенно, чтобы указанный пользователь был либо администратором сервера баз данных (`sa`), либо владельцем базы данных (если она уже существует), для того чтобы в дальнейшем беспрепятственно модифицировать структуру выбранной базы данных. В последнем случае этот пользователь должен иметь доступ на чтение к базе данных `master` и полный доступ к базе данных `tempdb`. Кроме перечисленного, указанный пользователь должен быть членом фиксированной серверной роли `processadmin` или `sysadmin`.
- **Пароль пользователя:** пароль пользователя, от лица которого будет выполняться доступ к базе данных.
- **Смещение дат – 0 или 2000.** Данный параметр определяет число лет, которое будет прибавляться к датам при их сохранении в базе данных Microsoft SQL Server и вычитаться при их извлечении. Наличие данного параметра определяется особенностями хранения дат в Microsoft SQL Server. Тип **DATETIME**, используемый в Microsoft SQL Server, позволяет хранить даты в диапазоне с 1 января 1753 года по 31 декабря 9999 года. И если при работе с информационной базой может возникнуть необходимость хранения дат, предшествующих нижней границе данного диапазона, то в качестве значения параметра следует выбрать 2000. Если же такие даты встречаться не будут, то в качестве смещения дат можно выбрать 0. После создания информационной базы значение данного параметра не может быть изменено.

ВНИМАНИЕ! Если в прикладном решении используются регистры накопления или регистры бухгалтерии, то в поле **Смещение дат** необходимо установить значение **2000**. Если при создании базы данных в это поле было введено значение **0**, то следует выгрузить информационную базу в файл (см. [здесь](#)), заново создать базу данных с указанием в поле **Смещение дат** значения **2000** и выполнить загрузку информационной базы (см. [здесь](#)).

5.1.1.2.2. PostgreSQL

- **Тип СУБД:** PostgreSQL.

- **Сервер баз данных:** имя экземпляра сервера.

- **Имя базы данных:** первым символом имени базы данных может быть буква стандарта Юникод 3.2, а также символы «_», «@». Последующими символами могут быть буквы, определенные стандартом Юникод 3.2, а также символы «_», «@», «\$». Длина имени ограничена 128 символами. В имени недопустимы пробелы. Имя не может быть зарезервированным словом языка запросов сервера баз данных.

- **Пользователь базы данных:** имя пользователя сервера баз данных, от лица которого будет осуществляться доступ к базе данных. Указанный пользователь должен обладать привилегиями `SUPERUSER`.

5.1.1.2.3. IBM DB2

- **Тип СУБД:** IBM DB2.

- **Сервер баз данных:** имя экземпляра сервера. Если на компьютере имеются экземпляры сервера баз данных, отличные от установленного по умолчанию, то необходимо указать также имя установленного экземпляра IBM DB2, заданное при его установке, через слеш. Например, `computer/db2name`.

- **Имя базы данных:** имя базы данных должно быть уникальным в пределах того расположения, в котором она занесена в каталог. В реализациях менеджера баз данных DB2 для систем Linux это положение представляет собой каталог, а в Windows – логический диск. Имя базы данных может начинаться только с буквы латинского алфавита, последующие символы могут быть буквами латинского алфавита, а также цифрами. Длина имени ограничена 8 символами.

- **Пользователь базы данных:** имя пользователя сервера баз данных, от лица которого будет осуществляться доступ к базе данных. Указанный пользователь должен либо обладать привилегиями `CREATEDB` или `SUPERUSER`, либо быть собственником базы данных (если она уже существует). Длина имени пользователя базы данных не должна превышать 8 символов.

5.1.1.2.4. Oracle Database

- **Тип СУБД:** Oracle Database.

- **Сервер баз данных:** имя экземпляра сервера. При создании информационной базы в качестве имени сервера баз данных указывается TNS-name. Это означает, что при создании информационной базы необходимо в качестве имени сервера баз данных написать строку типа `//имя_сервера_БД/имя_сервиса` (возможны и другие вариации написания TNS-name).

- **Имя базы данных:** **база данных** в терминах «1С:Предприятия» соответствует **схема данных** в терминах Oracle Database. При создании информационной базы данных в «1С:Предприятии» в Oracle Database создается пользователь и его схема данных. Имя схемы данных должно содержать только английские буквы, цифры и символ «_». Имя базы данных может начинаться только с буквы латинского алфавита. Длина имени ограничена 30 символами.

- **Пользователь базы данных:** при создании информационной базы данных в «1С:Предприятии» необходимо указать пользователя, от имени которого будет выполняться обращение к базе данных. Следует указать пользователя, обладающего правами `DBA` (например, `SYSTEM`) в том случае, если схема данных будет создаваться системой и другого пользователя, если схема уже создана администратором базы данных Oracle Database. Это может быть пользователь, чья схема данных используется для «1С:Предприятия», т. е. можно указать в свойствах **Имя базы данных** и **Пользователь базы данных** одно и то же имя.

Уточнение по необходимым правам

Если используется существующая схема данных, то пользователь, от имени которого будет выполняться обращение к базе данных, может не иметь права `DBA`. В этом случае для этого пользователя должны быть установлены следующие права доступа:

- `CREATE SESSION`,
- `CREATE PROCEDURE`,
- `CREATE TRIGGER`,
- `CREATE SEQUENCE`,
- `CREATE TABLE`.

Вместо указания конкретных прав, можно задать права на следующие роли:

- `CONNECT`,
- `RESOURCE`.

Также пользователю должны быть выделены квоты на табличные пространства `V81C_DATA`, `V81C_INDEX`, `V81C_LOB`, `V81C_INDEX_BIG` (если оно существует).

Для задания прав доступа можно воспользоваться следующим примером:

[Копировать в буфер обмена](#)

```
create user <username> identified "<password>";
grant create session to <username>;
grant create table to <username>;
grant create trigger to <username>;
grant create sequence to <username>;
grant create procedure to <username>;
alter user <username> quota UNLIMITED on v81c_data;
alter user <username> quota UNLIMITED on v81c_index;
alter user <username> quota UNLIMITED on v81c_lob;
```

Уточнение по сроку действия пароля

При использовании Oracle Database версии 11 с настройками по умолчанию срок действия пароля установлен 180 дней. По истечении 180 дней может произойти ошибка: [Сервер баз данных не обнаружен. Ошибка СУБД: ORA-28002: дата действия пароля истечет в течение 7 дней](#). В этом случае необходимо сменить пароль средствами СУБД:

- запустить утилиту [SQL*Plus](#);
- подключиться к базе данных;
- выполнить команду `ALTER USER <ИмяПользователя> IDENTIFIED BY <Пароль>`.

Пароль изменится на тот, который будет указан в запросе. Необходимо помнить, что сервер «1С:Предприятия» хранит пароль в файле настроек и будет пытаться зайти со старым паролем. Поэтому проще при выполнении команды `ALTER USER` указать старый пароль, тем самым не меняя пароль фактически.

Чтобы отключить срок действия пароля, следует обратиться к документации по Oracle Database соответствующей версии.

5.1.1.2.5. Общие параметры

- **Пароль пользователя:** пароль пользователя, от лица которого будет выполняться доступ к базе данных.
- При установленном флажке [Создать базу данных в случае ее отсутствия](#) база данных будет создана, если указанный сервер баз данных не содержит базу данных с указанным именем. Если же флажок не установлен, то попыток создания базы данных предприниматься не будет.
- Значение параметра [Язык \(Страна\)](#) выбирается из предложенного списка и определяет набор

национальных настроек, которые будут использованы при работе с информационной базой. В дальнейшем значение данного параметра можно изменить с помощью конфигулятора. Если создается новая информационная база из шаблона, содержащего файл выгрузки информационной базы (*.dt), то параметр **Язык** не отображается, т. к. информация о языке уже содержится в файле выгрузки информационной базы.

- Если установлен флажок **Установить блокировку регламентных заданий**, то в созданной информационной базе выполнение регламентных заданий будет запрещено. Если же флажок не установлен, то при подключении базы к серверу имеющиеся регламентные задания незамедлительно запустятся.

5.1.1.2.6. Создание базы данных

Если все параметры указаны корректно, то выполняются следующие действия:

- Производится попытка установить соединение с указанной базой данных в указанном сервере баз данных с применением указанных параметров пользователя.
- Если база данных отсутствует и установлен флажок **Создать базу данных в случае ее отсутствия**, то производится попытка создания требуемой базы данных. При создании информационной базы данных в Oracle Database создается пользователь с таким же паролем, как и имя пользователя. Во время создания пользователя его учетная запись блокируется. Сервер «1С:Предприятия» при установке соединения к Oracle Database использует имя пользователя и пароль, заданные при создании информационной базы данных.
- Если в указанной базе данных обнаруживается уже существующая информационная база «1С:Предприятия», то устанавливается связь с ней. Если же не обнаруживается, то инициализируется новая информационная база. Если при указании параметров создания новой информационной базы был указан шаблон, то данный шаблон будет применен при инициализации.

Описание страницы задания параметров запуска информационной базы см. [здесь](#).

5.1.1.2.7. Одновременное использование одной базы данных несколькими информационными базами

Платформа не запрещает при создании информационной базы на сервере «1С:Предприятия» указывать в нескольких информационных базах одну и ту же базу данных. Однако организация служебных данных кластера подразумевает, что одной базе данных соответствует одна совокупность служебных данных. Одновременное использование нескольких экземпляров служебных данных с одной базой данных нарушает их логическую целостность.

Если несколько информационных баз одновременно используют одну базу данных, то не будет работать:

- механизм блокировки информационной базы (в частности, запуск двух конфигураторов может привести к разрушению конфигурации);
- механизм блокировки объектов;
- менеджер управляемых блокировок;
- получение оперативной отметки времени;
- другие механизмы, использующие разделяемые данные, хранимые менеджером кластера.

Параллельная модификация данных в таких условиях может привести к их непоправимому разрушению. Чтение из базы данных также может выдать недостоверный результат.

Поэтому при штатном использовании информационной базы одновременно использовать одну базу данных несколькими информационными базами не следует.

В то же время одновременное подключение нескольких информационных баз к одной базе данных может оказаться полезным для целей отладки конфигураций и для расследования ошибок конфигураций и платформы. Поэтому в системе «1С:Предприятие» не предусмотрено программного запрета на

одновременное подключение нескольких информационных баз к одной базе данных. Однако пользоваться этой возможностью следует лишь в исключительных случаях с соблюдением необходимых предосторожностей:

- Параллельная модификация данных несколькими информационными базами в одной базе данных может привести к их непоправимому разрушению.
- Чтение данных одной информационной базой при параллельном использовании их другой информационной базой в обеих информационных базах может оказаться недостоверным, если хотя бы одна информационная база их модифицировала или устанавливала любые блокировки.

5.1.2. Добавление существующей информационной базы

Если выбран режим добавления существующей информационной базы, то в список может быть добавлена информационная база, расположенная на локальном рабочем месте, в локальной сети, на сервере «1С:Предприятия» или на веб-сервере (только для тонкого клиента и веб-клиента, см. [здесь](#)).

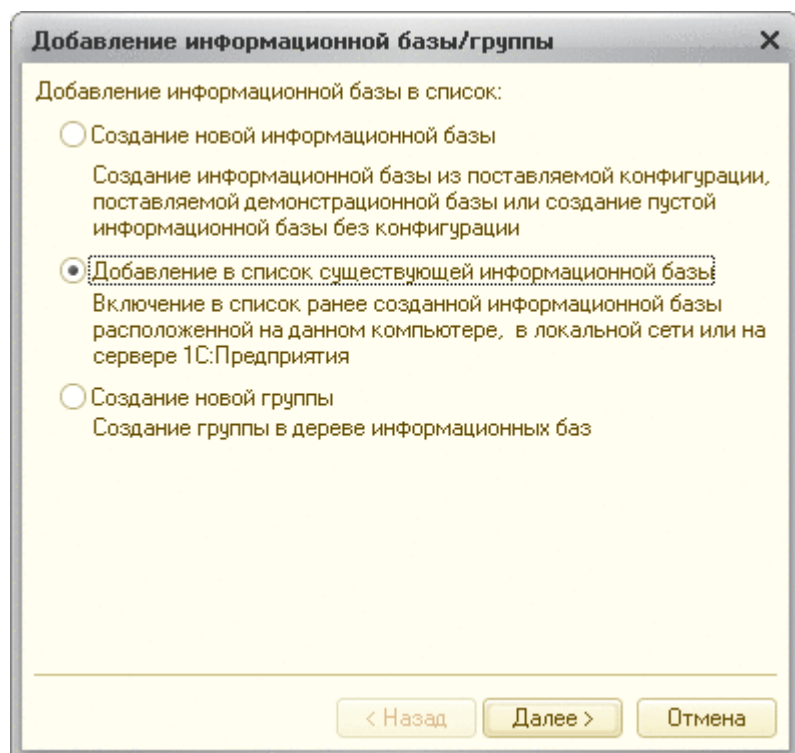


Рис. 29. Добавление существующей информационной базы

После нажатия кнопки [Далее >](#) на экран выводится диалог ввода наименования и выбора типа расположения информационной базы.

Наименование информационной базы представляет собой произвольную строку символов. Длина наименования информационной базы ограничена 255 символами, поэтому можно присваивать информационным базам смысловые имена.

Добавление существующей файловой информационной базы рассмотрено в следующем разделе, а добавление существующей информационной базы на сервере «1С:Предприятия» см. [здесь](#).

5.1.2.1. Файловый вариант

Для файлового варианта следует выбрать каталог, в котором будет находиться информационная база. Если будет указано имя несуществующего каталога, то при запуске системы «1С:Предприятие» он будет автоматически создан.

Нажатием кнопки выбора можно открыть стандартный диалог для выбора существующего каталога с информационной базой.

ПРИМЕЧАНИЕ. Имя каталога с информационной базой должно удовлетворять требованиям [RFC 2396](#),

раздел [2.4.3 Excluded US-ASCII Characters](http://tools.ietf.org/html/rfc2396.html) (<http://tools.ietf.org/html/rfc2396.html>). В имени каталога нельзя использовать следующие символы: «<», «>», «#», «%», «"», «{», «}», «[», «]», «|», «\», «^», «`», а также символы с кодами от 0 до 31 и 127.

Описание страницы задания параметров запуска информационной базы см. [здесь](#).

5.1.2.2. Клиент-серверный вариант

При добавлении в список уже существующей информационной базы на экран выводится окно [Добавление информационной базы/группы](#).

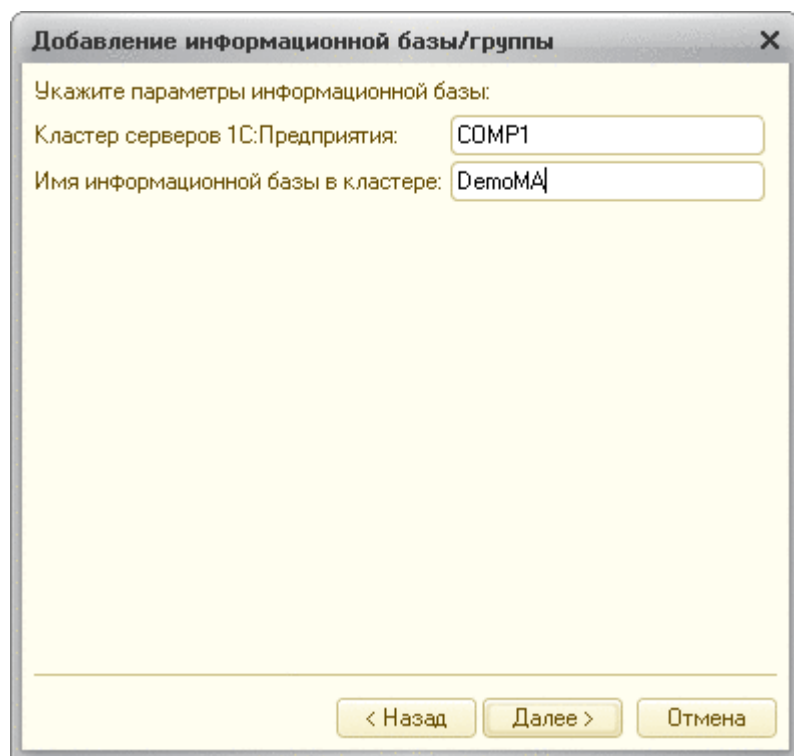


Рис. 30. Добавление существующей информационной базы

В полях потребуется указать:

- Адрес кластера серверов «1С:Предприятия». Адресом кластера считается адрес центрального сервера этого кластера с указанием номера сетевого порта процесса менеджера кластера (по умолчанию **1541**).
- Имя информационной базы.

ПРИМЕЧАНИЕ 1. Если в качестве адреса центрального сервера «1С:Предприятия» указан IP-адрес в точечной нотации, то его присутствие в DNS (hosts) не требуется.

ПРИМЕЧАНИЕ 2. Если используется резервирование кластеров (см. книгу «1С:Предприятие 8.3. Клиент-серверный вариант. Руководство администратора»), то список резервирования может задаваться непосредственно в поле [Кластер серверов 1С:Предприятия:](#) в формате [Server1,Server2:Port,Server3](#). В таком виде его удобнее использовать, например, при создании списка общих информационных баз.

ПРИМЕЧАНИЕ 3. Имя базы данных СУБД должно удовлетворять требованиям [RFC 2396](http://tools.ietf.org/html/rfc2396.html), раздел [2.4.3 Excluded US-ASCII Characters](http://tools.ietf.org/html/rfc2396.html) (<http://tools.ietf.org/html/rfc2396.html>). В имени базы данных нельзя использовать следующие символы: «<», «>», «#», "%», «"», «{», «}», «[», «]», «|», «\», «^», «`», а также символы с кодами от 0 до 31 и 127. Кроме того, эти требования могут конкретизироваться конкретной СУБД.

При этом не производится проверка того, существует ли информационная база с указанными параметрами или нет.

Если при запуске конфигулятора информационная база с указанными параметрами не обнаружена,

будет выдано соответствующее сообщение и вопрос о создании новой информационной базы. В случае положительного ответа конфигуратор выводит форму [Создание информационной базы/группы](#).

Создание информационной базы/группы

Укажите параметры информационной базы:

Кластер серверов 1С:Предприятия: COMP1

Имя информационной базы в кластере: DemoMA

Защищенное соединение: Выключено

Тип СУБД: PostgreSQL

Сервер баз данных:

Имя базы данных:

Пользователь базы данных:

Пароль пользователя:

Смещение дат: 0

Создать базу данных в случае ее отсутствия

Язык (Страна): русский (Россия)

Установить блокировку регламентных заданий

< Назад Далее > Отмена

Рис. 31. Параметры информационной базы

В поля данной формы нужно внести параметры, необходимые для создания новой информационной базы.

Описание страницы задания параметров запуска информационной базы см. [здесь](#).

5.1.2.3. База, расположенная на веб-сервере

Для добавления существующей информационной базы, расположенной на веб-сервере, необходимо запустить тонкий клиент «1С:Предприятия» (файл [1cv8c](#)).

В диалоге добавления информационной базы следует выбрать вид расположения информационной базы [На веб-сервере](#).

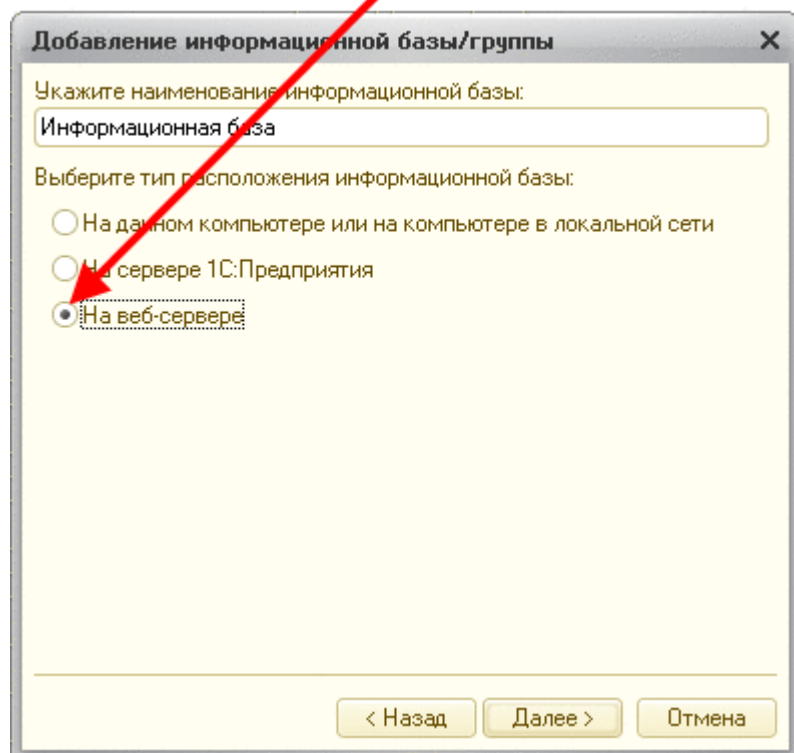


Рис. 32. Добавление информационной базы на веб-сервере

На следующей странице следует указать параметры подключения к веб-серверу (строку соединения, скорость соединения и т. д.).

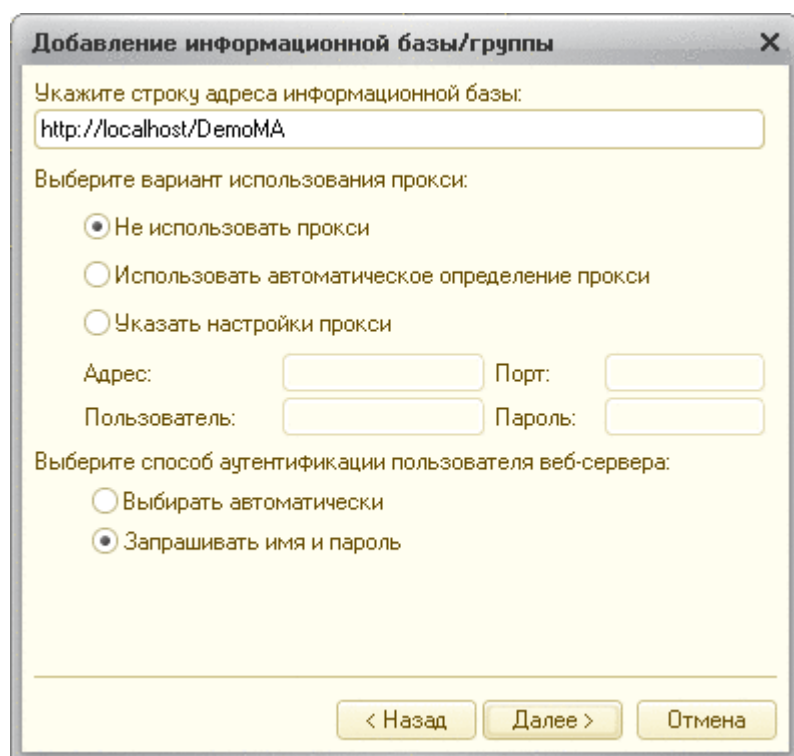


Рис. 33. Параметры подключения к веб-серверу

Параметр **Выберите способ аутентификации** пользователя веб-сервера позволяет выбрать способ аутентификации:

- **Выбирать автоматически** – в этом случае происходит попытка аутентификации на веб-сервере средствами ОС. Если попытка завершилась неудачно, запросить логин/пароль явным образом.
- **Запрашивать имя и пароль** – всегда запрашивать логин/пароль для аутентификации на веб-сервере.

Описание страницы задания параметров запуска информационной базы см. [здесь](#).

5.1.3. Параметры запуска информационной базы

На данной странице можно указать параметры запуска информационной базы.

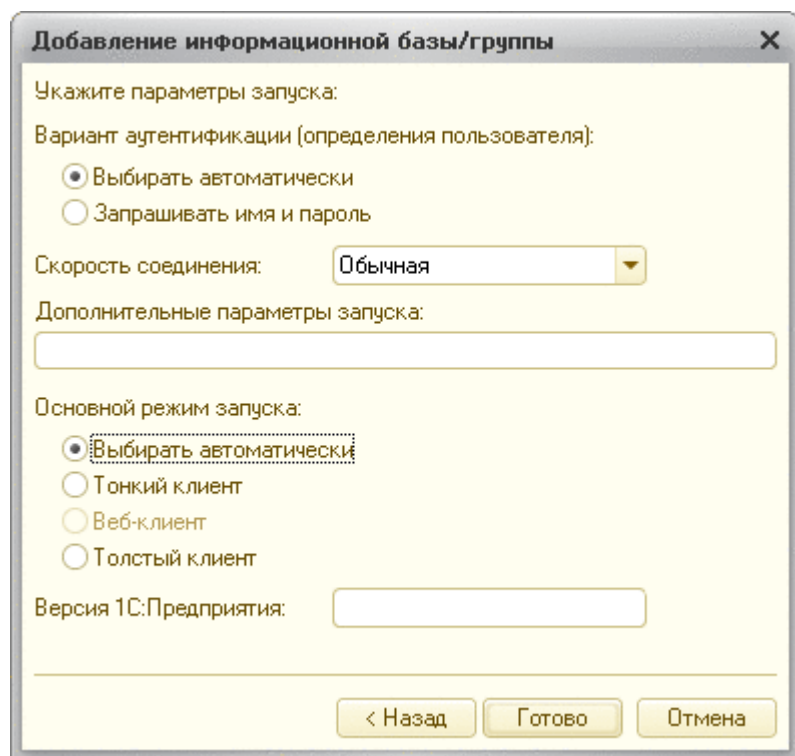


Рис. 34. Параметры запуска информационной базы

Параметр **Вариант аутентификации** может принимать следующие значения:

- **Выбирать автоматически** – в этом случае вначале будет предпринята попытка выполнить аутентификацию средствами ОС, а в случае неудачи – предложено ввести логин/пароль для доступа к информационной базе.
- **Запрашивать имя и пароль** – в этом случае аутентификация будет всегда выполняться с помощью окна ввода логина и пароля.

Параметр **Скорость соединения** позволяет определить скорость соединения с информационной базой или сервером «1С:Предприятия». Параметр может принимать следующие значения:

- **Обычная** – обычная скорость. При работе системы не будет никаких особенностей.
- **Низкая** – низкая скорость соединения. В этом режиме работа «1С:Предприятия» будет сопровождаться некоторыми особенностями, которые описаны в книге «1С:Предприятие 8.3. Руководство разработчика».
- **Выбирать при запуске** – в этом режиме выбирать скорость соединения можно будет при каждом запуске информационной базы с помощью флажка **Низкая скорость соединения** в нижней части окна **Запуск 1С:Предприятия**. Если в свойствах информационной базы указано конкретное значение (**Обычная** или **Низкая**), то флажок **Низкая скорость соединения** в диалоге **Запуск 1С:Предприятия** недоступен для изменения и отражает значение, выбранное в свойствах информационной базы.

Флажок **Низкая скорость соединения** в диалоге **Запуск 1С:Предприятия** тонкого клиента будет доступен для изменения в том случае, если в списке информационных баз есть хотя бы одна информационная база с требуемой версией «1С:Предприятия» 8.2 и выше и с параметром **Скорость соединения**, установленным в значение **Выбирать при запуске**. В остальных случаях флажок недоступен для изменения и отображает скорость соединения, указанную в свойствах информационной базы.

Поле **Дополнительные параметры запуска** позволяет указать различные параметры командной строки, которые будут переданы исполняемому файлу. Подробнее про параметры командной строки можно прочитать во встроенной справке (раздел **Запуск 1С:Предприятие 8 и параметры запуска**). Параметры **L** и **VL**, указанные в этом поле, будут действовать только в случае запуска информационной базы с

помощью интерактивной программы запуска (см. [здесь](#)).

Параметр **Основной режим запуска** определяет, какой клиент будет использоваться для доступа к информационной базе:

- **Выбирать автоматически** – в этом режиме вид клиентского приложения будет определяться исходя из свойства конфигурации **Основной режим запуска** и свойства пользователя **Режим запуска**.
- **Тонкий клиент** – для запуска будет использоваться тонкий клиент.
- **Веб-клиент** – для запуска будет использоваться веб-клиент. Данный вид клиента доступен только в том случае, если доступ к информационной базе выполняется через веб-сервер.
- **Толстый клиент** – для запуска будет использоваться толстый клиент. Данный вид клиента будет недоступен, если доступ к информационной базе выполняется через веб-сервер.

Поле **Версия 1С:Предприятия**: позволяет указать конкретный номер версии, который необходимо использовать для доступа к данной информационной базе. Кроме того, в этом поле можно указывать значения **8.1**, **8.0**. В этом случае для доступа к информационной базе будет использоваться версия 8.0 или 8.1 «1С:Предприятия», установленная на данном компьютере. Указывать конкретный номер версии в этом случае не имеет смысла.

5.1.4. Параметры настройки сертификатов

В случае если выбрано добавление информационной базы на веб-сервере (см. [здесь](#)), а в поле **Укажите строку адреса информационной базы** указать протокол HTTPS, например, вместо адреса <http://localhost/DemoMA> указать адрес <https://localhost/DemoMA>, то будет доступна страница с указанием параметров настройки сертификатов.

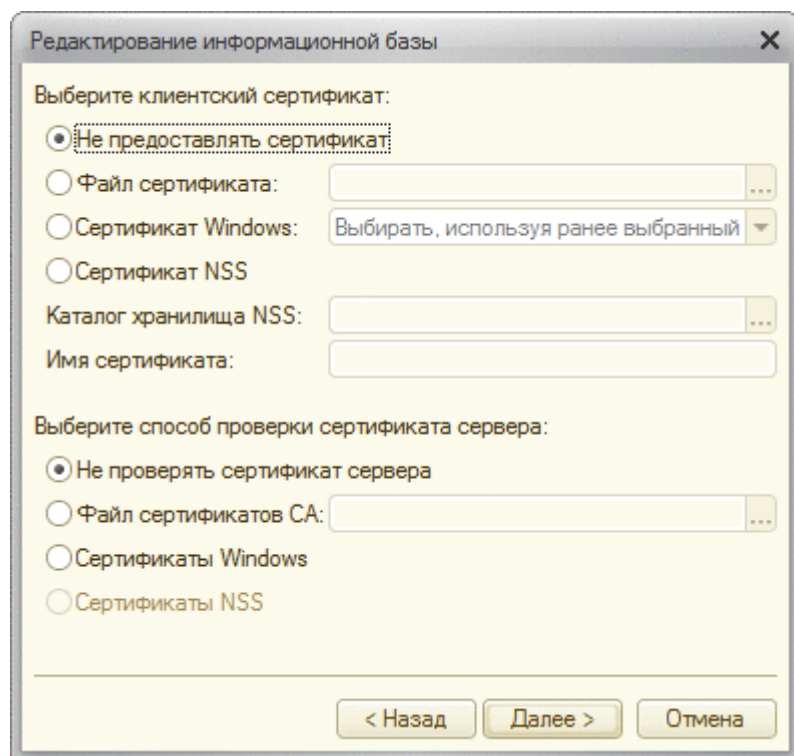


Рис. 35. Параметры настройки сертификатов

На этой странице настраивается, откуда выбирать клиентский сертификат и как проверять серверный сертификат. Опишем эти группы параметров более подробно:

- **Выберите клиентский сертификат** – выбирается месторасположение клиентского сертификата:
 - **Не предоставлять сертификат** – установка соединения возможна только с веб-серверами, которые не требуют предоставления клиентского сертификата.

- **Файл сертификата** – позволяет выбрать файл, хранящий клиентский сертификат и его закрытый ключ. Если файл защищен паролем, то при установлении соединения пользователю будет предложено ввести пароль.
- **Сертификат Windows** – клиентский сертификат получается из системного хранилища сертификатов ОС Windows. В том случае, если в системе окажется более одного клиентского сертификата, подходящего для устанавливаемого соединения, существует возможность указать системе, что делать в этом случае:
 - **Выбирать, используя ранее выбранный** – пользователю предоставляется выбор используемого сертификата через системный диалог выбора сертификата. В дальнейшем выбранный сертификат используется автоматически.
 - **Выбирать всегда** – пользователю предоставляется выбор используемого сертификата через системный диалог выбора сертификата независимо от того был ли ранее выбран какой либо сертификат. Выбранный сертификат в дальнейшем может использоваться автоматически с опцией **Выбирать, используя ранее выбранный**.
 - **Подбирать автоматически** – используется произвольный сертификат, подходящий для устанавливаемого соединения. Диалог выбора сертификата не отображается.
- **Сертификат NSS** – клиентский сертификат получается из хранилища сертификатов NSS. В этом случае необходимо указать дополнительно следующие настройки:
 - **Каталог хранилища NSS** – необходимо выбрать каталога, содержащий файлы хранилища сертификатов NSS. Если доступ к хранилищу защищен паролем, то при установке соединения пользователю будет предложено указать пароль доступа.
 - **Имя сертификата** – имя конкретного клиентского сертификата в выбранном хранилище сертификатов NSS. Если имя не указано, то используемый сертификат будет выбираться автоматически. При использовании для хранения сертификатов хранилища веб-браузера Mozilla Firefox, указывать значение для данного свойства не рекомендуется, т. к. имя сертификата зависит от используемого браузера и может быть нечитаемым.
- **Выберите способ проверки сертификатов сервера** – указывает, каким образом должны проверяться сертификаты, предъявленные веб-сервером:
 - **Не проверять сертификат сервера** – сертификат веб-сервера не проверяется, и, соответственно, не используются сертификаты удостоверяющих центров (УЦ).
 - **Файл сертификатов CA** – позволяет выбрать файл, хранящий сертификаты удостоверяющих центров. Если файл защищен паролем, то при установлении соединения пользователю будет предложено указать пароль доступа.

Данный параметр недоступен, если в качестве клиентского сертификата выбран сертификат NSS.

- **Сертификаты Windows** – указывает, что сертификаты удостоверяющих центров следует получать из системного хранилища сертификатов Windows.

Данный параметр недоступен, если в качестве клиентского сертификата выбран сертификат NSS.

- **Сертификаты NSS** – указывает, что сертификаты удостоверяющих центров следует получать из хранилища сертификатов NSS, указанного в настройках клиентского сертификата.

Данный параметр доступен только в том случае, если в качестве клиентского сертификата выбран сертификат NSS.

5.2. Редактирование информационной базы

Для корректировки имени или каталога какой-либо информационной базы из списка следует выбрать в списке название базы и нажать кнопку **Изменить**. Для изменения свойств информационной базы будет использоваться диалог, аналогичный диалогу добавления существующей информационной базы (см. [здесь](#)).

5.3. Удаление информационной базы из списка

Для удаления информационной базы из списка следует выбрать в списке ее название и нажать клавишу **Удалить**. Выбранная информационная база будет удалена из списка.

Примечание. Выполнение этой операции удаляет только информацию из списка, но не удаляет каталог с информационной базой с жесткого диска компьютера или базу, находящуюся на сервере «1С:Предприятия». Эту операцию следует провести самостоятельно.

5.4. Порядок информационных баз в списке

Если в настройках диалога запуска (см. раздел ниже) не установлен режим сортировки по наименованию, то положение информационных баз в списке можно изменить с помощью мыши или команд контекстного меню.

Для перемещения строки информационной базы следует выбрать в списке нужную строку, нажать левую клавишу мыши и, не отпуская ее, переместить указатель в нужном направлении. При перемещении выводится контур предполагаемого места размещения.

Затем следует переместить указатель в нужное место списка и отпустить клавишу мыши.

Порядок строк списка информационных баз также можно изменить с помощью команд контекстного меню **Переместить вверх** (**Ctrl + Shift + Стрелка вверх**) и **Переместить вниз** (**Ctrl + Shift + Стрелка вниз**). Действие команд «закольцовано», т. е. при достижении первой или последней позиции и повторном выполнении команды строка переместится в конец или начало списка.

Также доступны команды **Сортировать по возрастанию** и **Сортировать по убыванию**, которые соответствующим образом упорядочивают список информационных баз.

Если в настройках диалога запуска (см. [здесь](#)) включено отображение в виде дерева, то при перетаскивании строки информационной базы следует помнить о следующих особенностях:

- Если контур указывает на группу, то перетаскиваемая строка будет размещена в конце списка указанной группы.
- Для перемещения информационной базы или группы в нужную позицию группы предварительно нужно раскрыть эту группу.

5.5. Ведение иерархического списка информационных баз

В данном разделе описываются действия по созданию и реорганизации списка информационных баз, отображаемого в виде дерева.

5.5.1. Добавление группы информационных баз

Группы информационных баз имеет смысл создавать, когда на рабочем месте осуществляется работа с несколькими однотипными информационными базами или когда число баз достаточно велико и не сразу удастся найти требуемую базу.

Режим создания групп доступен, если в настройке диалога запуска был установлен режим **Отображать в виде дерева**.

При включенном режиме список баз представляется в виде дерева, в котором уже существует корневая группа **Информационные базы**. Эту группу нельзя изменять и удалять.

Для добавления группы информационных баз нужно выбрать группу, в которой требуется создать новую группу (указать строку группы или любую строку с информационной базой этой группы), и нажать кнопку **Добавить**. На экран будет выдан диалог для выбора режима добавления.

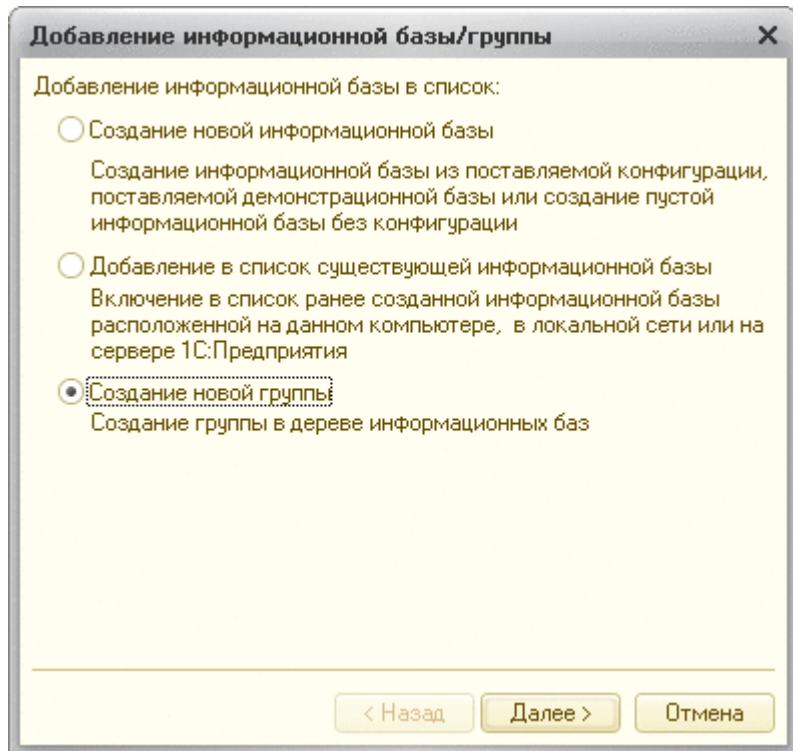


Рис. 36. Создание новой группы

Выбрать режим [Создание новой группы](#) и нажать кнопку [Далее >](#).

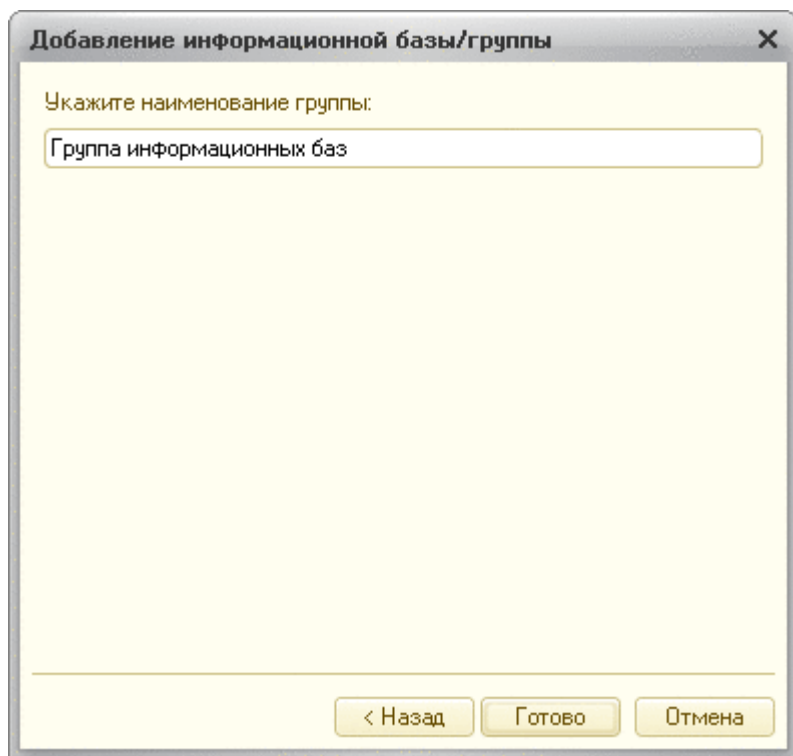


Рис. 37. Название новой группы

Затем ввести наименование группы (в наименовании группы недопустим символ «/») и нажать кнопку [Готово](#). Созданная группа помещается в указанную группу информационных баз (в конец списка указанной группы, если не установлен режим сортировки по наименованию).

5.5.2. Редактирование группы информационных баз

Для корректировки наименования группы информационных баз следует выбрать в списке строку, содержащую группу, и нажать кнопку [Изменить](#). На экран будет выдано окно [Редактирование группы](#), содержащее название выбранной группы информационных баз.

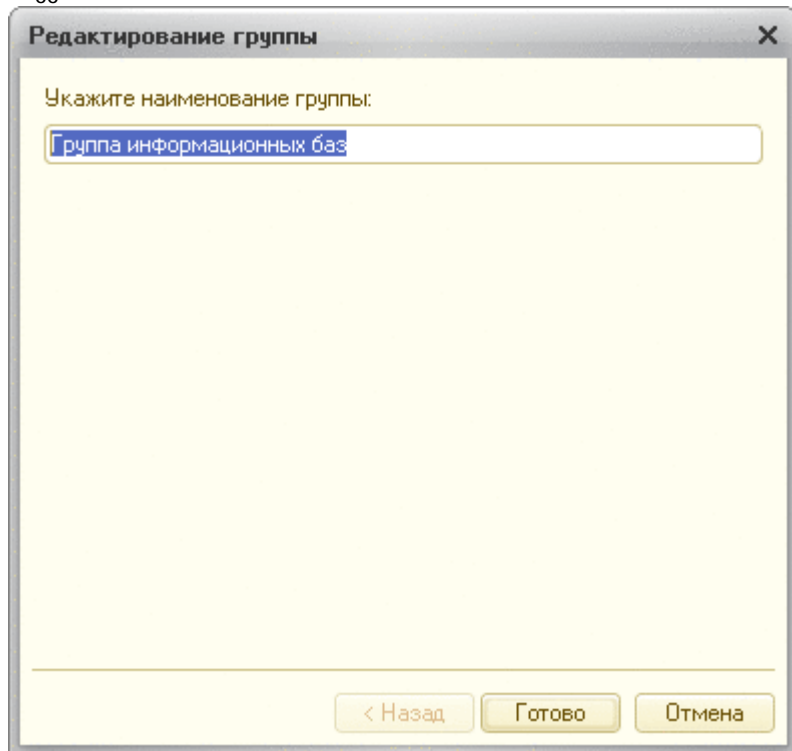


Рис. 38. Редактирование группы

Нужно ввести новое наименование (в наименовании группы недопустим символ «/») и нажать кнопку [Готово](#).

5.5.3. Удаление группы информационных баз

Для удаления группы информационных баз из списка следует выбрать в списке ее название и нажать кнопку [Удалить](#). Выбранная группа информационных баз будет удалена из списка.

ВНИМАНИЕ! Из списка будут удалены все информационные базы, входящие в группу.

5.6. Настройка окна запуска

В диалоге запуска нужно нажать кнопку [Настройка](#). На экран выводится диалог настройки запуска.

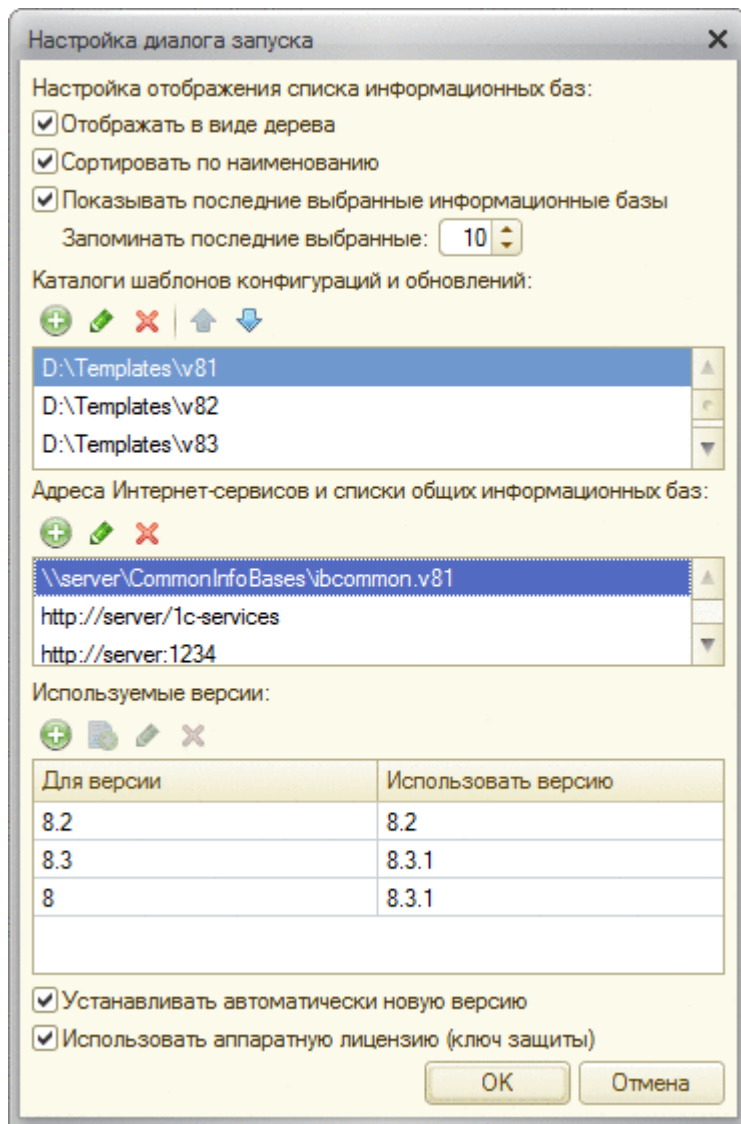


Рис. 39. Настройка диалога запуска

Данный диалог вызывается в том случае, если настройка выполняется из интерактивной программы запуска. При выполнении настройки из толстого клиента (1cv8) в окне настройки будет отсутствовать поле **Используемые версии**, а в случае вызова настройки из тонкого клиента (1cv8c) также не будет поля **Каталоги шаблонов конфигураций и обновлений**.

Если флажок **Отображать в виде дерева** установлен, то список информационных баз показывается в виде дерева.

Если флажок **Сортировать по наименованию** установлен, то список сортируется по наименованиям внутри каждой группы.

Если флажок **Показывать последние выбранные информационные базы** установлен, то в поле **Запоминать последние выбранные** указывается число последних вызванных баз.

Список последних выбранных баз показывается в верхней части общего списка. Наименования информационных баз выделяются жирным шрифтом. Этот список выводится в порядке выбора – вверху списка располагается база, которая выбиралась последней. Установка сортировки списка не влияет на порядок следования баз в списке последних выбранных. В данном списке допускается только выбор. Редактирование и удаление информационной базы становятся доступными при выборе ее в общем списке баз.

В поле **Каталоги шаблонов конфигураций и обновлений** указывается перечень каталогов, в которых располагаются шаблоны конфигураций и обновлений. Например, в этом списке может быть общефирменный каталог шаблонов и каталог шаблонов, используемый для локальных задач.

ПРИМЕЧАНИЕ. Поле **Каталог шаблонов конфигураций и обновлений** недоступно для окна настройки тонкого клиента.

Поле [Адреса Интернет-сервисов и списки общих информационных баз](#) предназначено для редактирования состава списков общих информационных баз. При запуске системы «1С:Предприятие» информационные базы, указанные в списках общих информационных баз, будут добавлены к основному списку информационных баз. Если в локальном конфигурационном файле указан параметр `CommonCfgLocation`, то к основному списку информационных баз также будут добавлены информационные базы, указанные в параметрах `CommonInfoBases` (если указаны) общего конфигурационного файла (`1cescmn.cfg`, описание см. [здесь](#)). Также к списку информационных баз будут добавлены информационные базы, полученные через Интернет-сервисы. Описание Интернет-сервисов получения списка общих информационных баз см. [здесь](#).

Пути к каталогу шаблонов или списку общих информационных баз отображаются в окне настройки только в том случае, если эти пути заданы с помощью соответствующих параметров локального конфигурационного файла `1cestart.cfg` (описание см. [здесь](#)). Если эти пути указаны в общем конфигурационном файле (`1cescmn.cfg`, описание см. [здесь](#)), то в диалоге настройки они не отображаются.

Поле [Используемые версии](#) содержит перечень уточнений номеров версий, используемых системой. Этот перечень используется в тех случаях, когда с информационными базами необходимо работать с помощью версии, которая отличается от максимальной версии, установленной на компьютере. Например, если при указании строки сопоставления вида `8.3=8.3.3.100` в свойствах информационной базы будет указана версия `8.3`, для ее запуска будет использоваться версия `8.3.3.100`, а не версия с максимально доступным номером.

Параметр [Использовать аппаратную лицензию \(ключ защиты\)](#) отвечает за поиск аппаратного ключа защиты при запуске клиентского приложения. Изменение параметра начинает действовать со следующего сеанса работы и изменяет значение параметра `UseHwLicenses` в файле `1cestart.cfg` (см. [здесь](#)).

ПРИМЕЧАНИЕ. При настройке диалога запуска параметры `CommonInfoBases`, `ConfigurationTemplatesLocation`, `DefaultVersion`, `UseHwLicenses` изменяются только в локальном конфигурационном файле пользователя, от имени которого выполняется настройка диалога запуска.

Параметр [Устанавливать автоматически новую версию](#) управляет возможностью автоматической установки новой версии. Изменение флажка влияет на значение параметра `AppAutoInstallLastVersion` в файле `1cestart.cfg` (см. [здесь](#)).

5.7. Списки общих информационных баз

Списки общих информационных баз представляют собой файлы с расширением `v8i`, которые содержат ссылки на общие информационные базы.

Расположение списков общих информационных баз задается в параметре `CommonInfoBases` конфигурационных файлов (описание файла `1cestart.cfg` см. [здесь](#), описание файла `1cescmn.cfg` см. [здесь](#)). Списки общих информационных баз имеют такой же формат, как и основной список информационных баз (см. [здесь](#)).

Список общих информационных баз может быть сформирован вручную или с использованием возможности сохранения существующих ссылок на информационные базы в файл. Для этого следует выполнить команду контекстного меню списка информационных баз [Сохранить ссылку в файл](#).

Список общих информационных баз может быть непосредственно использован для запуска системы «1С:Предприятие». При запуске файла с расширением `v8i` будет запущена система «1С:Предприятие», и в диалоге запуска будут отображены только те ссылки, которые содержатся в данном списке общих информационных баз.

СОВЕТ. Рекомендуется в списках общих баз указывать скорость соединения [Обычная](#) (если нет удаленных пользователей или информационная база не находится на удаленном сервере), что исключит отображение флажка [Низкая скорость соединения](#).

Имеется возможность также указать адрес Интернет-сервиса, который будет предоставлять список общих информационных баз в тех случаях, когда невозможно использовать список общих информационных баз, расположенный в локальной сети. Например, информационная база используется

через Интернет (подключение через веб-сервер). Подробнее об Интернет-сервисе получения списка общих информационных баз см. [здесь](#).

Глава 6. Администрирование информационной базы

При работе с системой «1С:Предприятие» необходимо выполнять различные действия, связанные с администрированием системы, например:

- ведение списка пользователей;
- назначение прав пользователям;
- резервное копирование;
- создание технологического журнала для разбора ошибок и т. д.

Конфигуратор содержит средства администрирования, предназначенные для решения указанных задач.

Так, в системе «1С:Предприятие» существует возможность создания списка пользователей, которым разрешена работа с системой. Этот список будет использоваться для авторизации пользователя при его входе в систему. Следует обратить внимание, что список пользователей системы «1С:Предприятие» не является частью конфигурации: он создается отдельно в конкретной организации, в которой используется система.

Для каждого пользователя может быть установлен пароль на вход в систему. Пароль используется для подтверждения прав пользователей на работу в системе «1С:Предприятие».

Другой важной задачей административного характера является ведение резервного копирования. Проведение этой процедуры должно выполняться периодически, чтобы в случае разрушения базы данных иметь возможность с минимальными потерями восстановить исходные данные. Частота определяется интенсивностью изменений данных. Чем чаще меняются данные, тем чаще следует производить резервное копирование.

В этой главе будут рассмотрены вопросы администрирования «1С:Предприятия», которые можно выполнять с помощью конфигуратора.

6.1. Ведение списка пользователей

Список пользователей вызывается на экран выбором пункта [Администрирование – Пользователи](#).

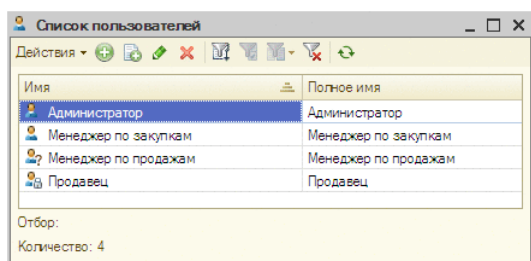


Рис. 40. Список пользователей

Окно со списком пользователей имеет панель инструментов и табличное поле с двумя колонками:

- В колонке **Имя** выводится список пользователей, зарегистрированных для работы с системой «1С:Предприятие 8».
- Колонка **Полное имя** может содержать расшифровку имени, выданного в первой колонке.

Пользователи, для которых определен пароль доступа, отображаются пиктограммами с замочком (пользователь [Продавец](#) на [рис. 40](#)).

Пользователи, для которых не определена роль или аутентификация, отображаются пиктограммами с вопросом (пользователь [Менеджер по продажам](#) на [рис. 40](#)).

С помощью пунктов меню **Действия** осуществляется ведение списка пользователей, настройка показа списка (отбор, состав и порядок колонок, сортировка), а также вывод списка в табличный или текстовый документы.

6.1.1. Добавление нового пользователя

Для добавления нового пользователя необходимо выбрать пункт **Действия – Добавить** в окне [Список пользователей](#). На экран будет выдано окно для редактирования параметров пользователя.

На закладке **Основные** указывается имя и полное имя пользователя.

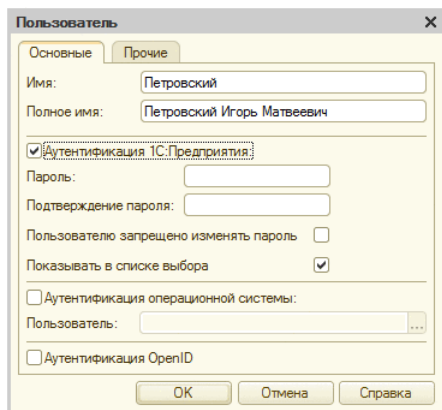


Рис. 41. Новый пользователь

В имени пользователя не рекомендуется указывать символ ":". Уникальность пользователя информационной базы поддерживается по совокупности значений трех полей: имя, полное имя и имя пользователя операционной системы (если включена аутентификация средствами операционной системы). Для поля **Имя** уникальность поддерживается по первым 64 символам, для поля **Полное имя** – по первым 128 символам, для поля **Пользователь операционной системы** – по первым 128 символам. Рекомендуется не допускать для поля **Имя** превышения длины в 64 символа.

СОВЕТ. Желательно задавать пользователям смысловые имена, используя фамилию сотрудника, наименование должности, характер выполняемых им функций и тому подобное. В дальнейшем это имя будет использоваться сотрудником для входа в систему «1С:Предприятие».

Для пользователя необходимо указать способ аутентификации. Подробнее про виды аутентификации, которые поддерживает «1С:Предприятие», см. [здесь](#).

ПРИМЕЧАНИЕ. Клиентское приложение, работающее в ОС Linux, не поддерживает аутентификацию операционной системы.

Каждый из флажков **Аутентификация...** (**Аутентификация 1С:Предприятия**, **Аутентификация операционной системы**, **Аутентификация OpenID**) определяет возможность выполнения аутентификации для данного пользователя тем или иным способом. Эти флажки не влияют на порядок попыток аутентификации. При назначении видов аутентификации следует помнить о следующих особенностях:

- Если сняты все флажки **Аутентификация...**, то данному пользователю запрещен доступ к прикладному решению.
- Для того чтобы выполнялась попытка аутентификации с помощью протокола OpenID, необходимо, чтобы соответствующим образом была настроена публикация данной информационной базы на веб-сервере (см. [здесь](#)).
- Пользователю будет запрещен доступ к прикладному решению, если он аутентифицирован средствами ОС или OpenID, но у вида аутентификации сброшен флажок, разрешающий эту аутентификацию для пользователя.
- Отключить попытку аутентификации средствами ОС или с помощью OpenID можно соответствующими ключами командной строки запуска клиентского приложения.

ВНИМАНИЕ! В системе должен быть по крайней мере один пользователь, который обладает административными правами и допускает аутентификацию средствами «1С:Предприятия».

Если установлен флажок **Пользователю запрещено изменять пароль**, то это означает, что данный пользователь не может изменять свой пароль (используется, если включена аутентификация системы «1С:Предприятие»).

Если флажок **Показывать в списке выбора** установлен, то данный пользователь будет отображаться в списке выбора при соединении с информационной базой системы «1С:Предприятие». Если пользователю запрещена аутентификация средствами системы «1С:Предприятие», то флажок **Показывать в списке** становится недоступным для редактирования, а пользователь не будет отображен в списке выбора при соединении с информационной базой.

На закладке **Прочие** указываются доступные роли и язык. Если ролей в конфигурации определено несколько, то для пользователя можно также указать несколько ролей. Кроме того, для пользователя можно указать режим запуска «1С:Предприятия». Если используется значение **Авто**, то при запуске будет использоваться режим запуска, установленный в свойстве конфигурации **Основной режим запуска**. Указание конкретного режима запуска можно использовать в тех случаях, когда некоторые пользователи должны работать в особом режиме. Например, какой-либо пользователь работает в режиме управляемого приложения. Тогда в поле **Режим запуска** стоит указать **Управляемое приложение**.

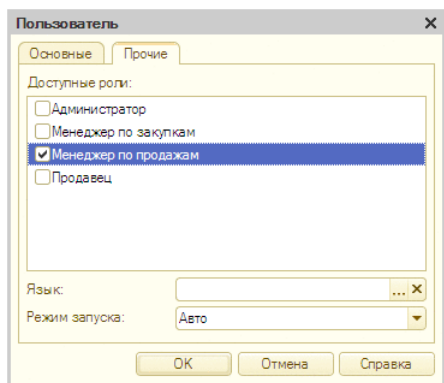


Рис. 42. Прочие параметры нового пользователя

В окне для редактирования свойств пользователя не обязательно заполнять сразу все поля – это можно сделать позднее.

6.1.2. Копирование пользователя

Нового пользователя можно создать путем копирования существующего. Используя такую возможность, нет необходимости создавать нового пользователя с нуля – достаточно скопировать одного из уже существующих в списке пользователей и отредактировать его свойства.

Для копирования нужно выбрать исходную строку списка пользователей и выполнить команду **Действия – Скопировать**.

При копировании имя пользователя может быть преобразовано с целью соблюдения уникальности. Остальные свойства нового пользователя будут такие же, как и у пользователя, выбранного в качестве образца (кроме пароля).

6.1.3. Установка пароля

С целью предотвращения входа пользователей в систему «1С:Предприятие» под чужими именами каждому пользователю, которому разрешена работа с системой, может быть установлен пароль на вход. Как и имя пользователя, пароль служит для подтверждения полномочий пользователя на работу в системе.

В поле для ввода пароля нужно ввести пароль пользователя. Пароль представляет собой произвольную строку, состоящую из букв и цифр. Размер пароля не должен превышать 255 символов.

При вводе пароль отображается звездочками, поэтому будьте внимательны.

В поле **Подтверждение пароля** следует указать введенный пароль еще раз для исключения ошибки ввода. Если повторно введенный пароль отличается от первоначального, то при нажатии кнопки **ОК** на экран выводится предупреждение **Пароль и подтверждение пароля не совпадают**, и пароль не будет установлен.

Если вы передумали устанавливать пароль, то нужно нажать кнопку **Отмена**.

ВНИМАНИЕ! Присвоенный пользователю пароль нельзя посмотреть, поэтому будьте внимательны при установке пароля и хорошо запомните его.

Если пользователь забыл свой пароль, необходимо задать ему новый пароль.

Пользователи, имеющие пароль, отличаются в списке пользователей пиктограммой (замочек на пиктограмме – см. пользователь **Продавец** на [рис. 40](#)).

6.1.4. Удаление пользователя

Для удаления пользователя следует в списке пользователей выделить его имя и выбрать пункт **Действия – Удалить** окна **Список пользователей**.

Для подтверждения удаления пользователя в появившемся на экране запросе следует нажать **ОК**.

6.1.5. Редактирование свойств пользователя

Для редактирования параметров пользователя предназначен пункт **Администрирование – Пользователи** меню конфигулятора. Выбрав нужного пользователя в списке, следует выбрать пункт **Действия – Изменить** меню окна **Список пользователей**.

В окне **Параметры пользователя** можно изменить параметры выбранного пользователя.

6.1.6. Установка отбора

Для удобства просмотра списка пользователей можно использовать отбор. В списке пользователей выбирается пункт **Действия – Установить отбор...**

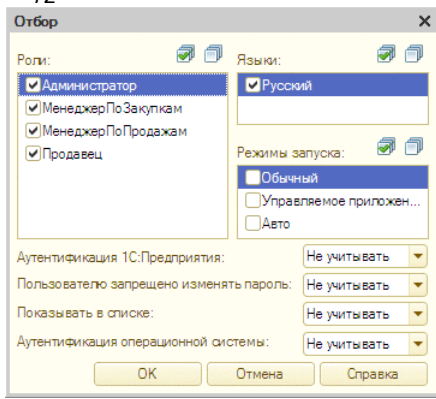


Рис. 43. Установка отбора

Отбор можно выполнить по роли, языку, режиму запуска, а также с учетом аутентификации пользователя.

6.1.7. Виды аутентификации

Аутентификация – проверка принадлежности предъявленного идентификатора (имени) конкретному пользователю системы, проверка подлинности. Система «1С:Предприятие» поддерживает несколько различных вариантов аутентификации, которые будут рассмотрены в следующих разделах.

6.1.7.1. Аутентификация средствами системы «1С:Предприятие»

Пользователь может быть аутентифицирован системой «1С:Предприятие» с помощью ввода его имени и пароля (в диалоге аутентификации, в виде параметров командной строки или строки соединения с информационной базой для внешнего соединения или automation-сервера). В этом случае проверка наличия пользователя и корректности ввода его пароля выполняет система «1С:Предприятие».

6.1.7.2. Аутентификация операционной системы

Пользователь может быть аутентифицирован неявно средствами операционной системы. Для этого пользователю должен быть поставлен в соответствие некоторый пользователь операционной системы. При старте системы, «1С:Предприятие» запрашивает у операционной системы пользователя, который аутентифицирован в системе в данный момент. Для этого в ОС Windows используется интерфейс SSPI, а в ОС Linux – GSS-API. Затем выполняется проверка, что данному пользователю операционной системы сопоставлен пользователь «1С:Предприятия». Если поиск заканчивается успешно – считается, что пользователь системы «1С:Предприятие» аутентифицирован успешно, и диалог аутентификации не отображается.

ПРИМЕЧАНИЕ 1. Клиентское приложение для ОС Linux не поддерживает аутентификацию средствами операционной системы.

ПРИМЕЧАНИЕ 2. Не поддерживается аутентификация пользователя средствами ОС в том случае, если клиентское приложение подключается к информационной базе через веб-сервер Apache, работающий под управлением ОС Windows.

Пользователь операционной системы указывается в формате: `\\имя_домена\имя_пользователя`.

Если необходимо принудительно выполнить аутентификацию средствами системы «1С:Предприятие», то в командной строке запуска клиентского приложения следует указать ключ командной строки `/WA-`. Соответственно, ключ командной строки `/WA+` предназначен для принудительного применения аутентификации средствами операционной системы (действует по умолчанию).

6.1.7.3. Аутентификация с помощью OpenID

OpenID (<http://openid.net/>) – это протокол, который позволяет пользователю использовать единую учетную запись для аутентификации на множестве не связанных друг с другом ресурсов, систем и т. д. Система «1С:Предприятие» использует протокол, созданный на основе протокола OpenID версии 2.0 по модели Direct Identity.

ПРИМЕЧАНИЕ 1. Данный способ аутентификации не применим при обращении к веб-сервисам, опубликованным из «1С:Предприятия».

ПРИМЕЧАНИЕ 2. В роли провайдера OpenID выступает информационная база «1С:Предприятия».

Общая схема работы выглядит следующим образом:

- Пользователь пытается выполнить вход в систему.
 - Система определяет, что в информационной базе работает OpenID-аутентификация (по файлу публикации `default.vrd`).
 - Провайдеру OpenID отправляется запрос на выполнение аутентификации.
 - Если необходимо выполнить интерактивное действие (выполняется первая аутентификация для данного идентификатора или закончено время жизни признака аутентификации данного идентификатора), то провайдер сообщает системе о необходимости запросить имя и пароль пользователя. Система выполняет интерактивное действие и возвращает провайдеру OpenID запрошенные данные.
- Признак аутентифицированности пользователя хранятся в файлах cookie, которые размещаются в хранилище, индивидуальном для каждого веб-браузера. Тонкий клиент использует собственное хранилище.
- Если провайдер аутентифицирует пользователя, то системе возвращается признак того, что пользователь аутентифицирован.

OpenID-аутентификация работает только в тех случаях, когда доступ к информационной базе осуществляется по протоколу HTTP и HTTPS. Это означает, что использовать OpenID-аутентификацию могут только веб-клиент и тонкий клиент, подключенный к информационной базе через веб-сервер. При OpenID-аутентификации возможны кросс-доменные запросы при работе с помощью тонкого клиента, а также с помощью веб-браузеров Mozilla Firefox, Google Chrome, Safari и Microsoft Internet Explorer версий 8 и 9. В веб-браузере Microsoft Internet Explorer версий 6.0 и 7, после ввода имени пользователя и пароля, открывается окно сообщения с запросом подтверждения на выполнение операции. Если пользователь подтверждает выполнение операции – процесс аутентификации продолжается. В противном случае вновь предлагается ввести имя пользователя и пароль.

В качестве OpenID-провайдера выступает информационная база системы «1С:Предприятие». В качестве OpenID-идентификатора используются имена пользователей информационной базы (см. [здесь](#)). Такая информационная база должна быть особым образом опубликована на веб-сервере (в файле публикации `default.vrd` расположен особый элемент) и доступна для информационной базы, которая желает выполнять аутентификацию с помощью OpenID.

ВНИМАНИЕ! Взаимодействие с OpenID-провайдером осуществляется только по HTTPS-соединению.

В качестве OpenID-идентификатора пользователя выступает свойство `Имя` пользователя информационной базы OpenID-провайдера. Пароль пользователя также задается в информационной базе OpenID-провайдера. Пароль, заданный в информационной базе, которая является клиентом OpenID-провайдера, игнорируется при выполнении аутентификации с помощью OpenID.

Если необходимо принудительно выполнить аутентификацию с помощью OpenID, то в командной строке запуска клиентского приложения следует указать ключ командной строки `/OIDA+` (действует по умолчанию). Соответственно, ключ командной строки `/OIDA-` предназначен для принудительного отключения аутентификации с помощью OpenID.

Подробнее о настройке веб-сервера для работы с OpenID-аутентификацией см. [здесь](#).

6.2. Список активных пользователей

В процессе работы бывает необходимо определить, какие пользователи работают в данный момент с информационной базой.

Для получения списка пользователей нужно выбрать пункт [Сервис – Активные пользователи](#). На экран выводится окно со списком пользователей, работающих в данный момент с базой данных.

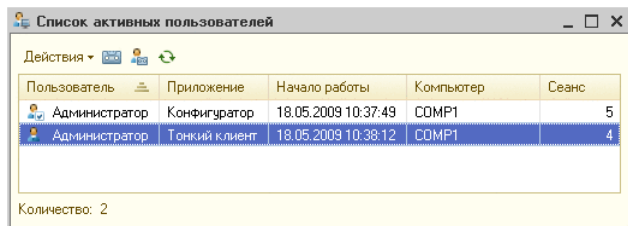


Рис. 44. Список активных пользователей

При открытии текущая строка показывает данные пользователя, вызвавшего окно (текущий сеанс).

Текущий пользователь отличается в списке пиктограммой (имеет пометку на пиктограмме).

С помощью пунктов меню [Действия](#) можно настроить показ списка, а также вывести его в табличный или текстовый документ.

Список активных пользователей можно сортировать по любой колонке.

6.3. Блокировка установки сеансов пользователями

Система «1С:Предприятие» позволяет устанавливать блокировки сеансов пользователей с информационной базой. Можно запретить установку сеансов пользователей с информационной базой с отображением сообщения о причине запрета.

Эта возможность полезна, например, когда для выполнения административных действий требуется, чтобы текущие пользователи завершили свои сеансы работы и в то же время новые пользователи не могли подключиться к информационной базе.

При работе в клиент-серверном варианте работы установка блокировки может быть выполнена с помощью утилиты администрирования кластера серверов «1С:Предприятия».

Предусмотрена возможность соединения с информационной базой в обход установленной блокировки сеансов. Для этого используется параметр командной строки `/UC<код разрешения>` и параметр строки соединения `UC=<код разрешения>`. Если при установке блокировки задан непустой код разрешения, то для установки соединения в обход блокировки необходимо в параметре `/UC` указать этот код разрешения. Если код разрешения содержит пробелы, он должен быть заключен в кавычки.

Если используется веб-клиент или тонкий клиент, работающий через веб-сервер, имеется возможность указания кода разрешения в параметре `UC` строки соединения файла-дескриптора (см. [здесь](#)). В этом случае рекомендуется выполнять дополнительную публикацию информационной базы на веб-сервере.

Программный способ

Кроме этого, при работе в любом режиме установка блокировки может быть выполнена средствами встроенного языка. Для этого используется объект встроенного языка `БлокировкаСеансов`, который можно создать с помощью конструктора и установить необходимые свойства блокировки установок соединений.

Метод глобального контекста `УстановитьБлокировкуСеансов()` позволяет установить созданную блокировку, а метод `ПолучитьБлокировкуСеансов()` – получить установленную блокировку.

6.4. Региональные установки информационной базы

Режим настройки региональных установок информационной базы позволяет управлять форматом отображения даты, времени, чисел, логических констант и влияет на порядок сортировки строк в списках информационной базы данных. Для вызова режима нужно выбрать пункт [Администрирование – Региональные установки ИБ](#).

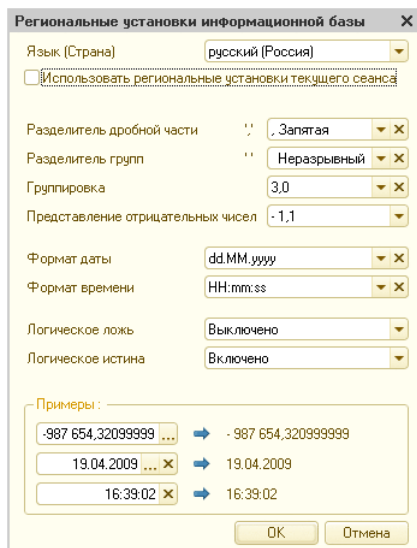


Рис. 45. Региональные установки

Если какое-либо свойство не установлено, то форматы отображения чисел, даты и времени будут определяться установками по умолчанию, принятыми в системе «1С:Предприятие» для указанного языка (страны). Язык (страна) определены при создании информационной базы.

Если установлено свойство [Использовать региональные установки текущего сеанса](#), то значения типа `Число` и `Дата` отображаются (в том числе в полях ввода, календаре и калькуляторе) в соответствии с региональными настройками текущего сеанса. Эти настройки определяются на основании региональных установок клиентского компьютера, но могут быть переопределены параметром `/VL`.

В нижней части диалога выводятся примеры отображения числа, даты и времени с выбранными в диалоге установками.

Значения типа `Булево` отображаются в соответствии с языком интерфейса платформы. Это значение может быть указано с помощью параметра `/L`.

Язык. Выбор языка (страны) для данной установки информационной базы.

ВНИМАНИЕ! Если в качестве СУБД используется PostgreSQL, то для существующей информационной базы изменение языка (страны) не может быть выполнено

произвольным образом. Установленное значение языка (страны) можно изменить только на такое, которое будет использовать тот же порядок сортировки строк (collation) СУБД, что и существующее значение. Например, **русский (Россия)** может быть изменен на **белорусский (Беларусь)**, но не может быть изменен на **украинский (Украина)**.

Если в качестве СУБД используется IBM DB2, то смена значения языка (страны) не поддерживается.

Разделитель дробной части. Символ-разделитель целой и дробной части чисел можно выбрать из выпадающего списка или указать самостоятельно в поле ввода. Пример символа будет выведен в рамке слева от поля ввода.

Разделитель групп. Символ-разделитель групп цифр в целой части числа можно выбрать из выпадающего списка или указать самостоятельно в поле ввода. Пример символа будет выведен в рамке слева от поля ввода.

Группировка. Свойство задает формат группировки цифр в целой части числа. Форматную строку можно выбрать из выпадающего списка или указать самостоятельно.

Формат группировки указывается следующим образом: **<количество цифр в группе><символ-разделитель>... ..<0>**.

В качестве символа-разделителя можно использовать любой символ, отличный от цифры.

Например, последовательность символов **3,2,0** означает, что цифры будут сгруппированы следующим образом (отсчет цифр в числе идет слева направо только в целой части):

- первую группу образуют первые три цифры числа;
- затем идет символ-разделитель групп (заданный настройками операционной системы или указанный в свойстве **Разделитель групп**);
- все оставшиеся цифры числа будут сгруппированы по две.

Символ **0** в конце форматной строки означает «и так же – до конца». То есть если в вышеприведенном примере форматной строки убрать **0**, указав **3,2**, группировка изменится следующим образом:

- первую группу образуют первые три цифры числа;
- затем идет символ-разделитель групп;
- вторую группу образуют следующие 2 цифры числа;
- затем идет символ-разделитель групп;
- все оставшиеся цифры числа будут собраны вместе.

Выбор одного символа **0** в этом поле означает, что цифры в целой части чисел не будут разделяться на группы.

Представление отрицательных чисел. Из выпадающего списка можно выбрать вид отрицательных чисел. Выбор **Авто** означает, что вид отрицательных чисел будет определяться установками операционной системы.

Формат даты. Устанавливает формат отображения даты. Можно использовать в различных сочетаниях следующие символы:

Символы	Описание
D	Число месяца. Числа меньше 10 выводятся без лидирующего нуля
Dd	Число месяца. Числа меньше 10 выводятся с лидирующим нулем
M	Номер месяца. Номера месяцев меньше 10 выводятся без лидирующего нуля
MM	Номер месяца. Номера месяцев меньше 10 выводятся с лидирующим нулем
MMMM	Наименование месяца словами
Y	Две последние цифры года. Года меньше 10 выводятся без лидирующего нуля
Yu	Две последние цифры года. Года меньше 10 выводятся с лидирующим нулем
Yuyu	Год четырьмя цифрами

Перечисленные выше символы и группы символов можно указывать в любой последовательности. Для разделения дня, месяца и года можно указывать различные символы-разделители.

Формат времени. Устанавливает формат отображения времени. Можно использовать в различных сочетаниях следующие символы:

Символы	Описание
h, H	часы в 12-часовом (h) или 24-часовом (H) формате. Часы меньше 10 выводятся без лидирующего нуля
hh, HH	часы в 12-часовом (hh) или 24-часовом (HH) формате. Часы меньше 10 выводятся с лидирующим нулем
m	минуты. Минуты меньше 10 выводятся без лидирующего нуля
mm	минуты. Минуты меньше 10 выводятся с лидирующим нулем
s	секунды. Секунды меньше 10 выводятся без лидирующего нуля
ss	секунды. Секунды меньше 10 выводятся с лидирующим нулем

Перечисленные выше символы и группы символов можно указывать в любой последовательности. Для разделения часов, минут и секунд можно указывать различные символы-разделители.

ВНИМАНИЕ! При использовании региональных настроек для определения представления даты в поле ввода следует выбирать только такие настройки, которые поддерживаются полем ввода.

Логическое ложь, логическое истина. Позволяет указать вид логических констант. Можно выбрать из выпадающего списка либо ввести самостоятельно.

6.5. Параметры информационной базы

Режим настройки параметров информационной базы позволяет задавать время ожидания блокировки данных и указывать необходимость использования ограничений на пароли пользователей.

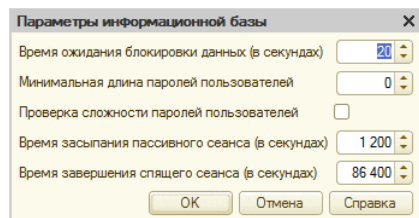


Рис. 46. Параметры информационной базы

Для настройки доступны перечисленные ниже параметры:

Время ожидания блокировки данных (в секундах)

Определяет максимальное время ожидания установки транзакционной блокировки сервером баз данных. Например, если текущая транзакция должна установить блокировку на запись, а запись уже заблокирована другой транзакцией, то текущая транзакция будет ожидать снятия блокировки, но не дольше, чем значение данного параметра. Аналогичным образом этот параметр регулирует время ожидания транзакционной блокировки в режиме управляемых блокировок системы «1С:Предприятие».

Минимальная длина паролей пользователей

Указывает минимальную длину пароля пользователя. Если установлен параметр [Проверка сложности паролей пользователей](#), то минимальная длина пароля пользователя не может быть менее 7 символов.

Проверка сложности паролей пользователей

Если данный параметр установлен, пароли пользователей должны удовлетворять следующим требованиям:

- длина пароля не должна быть менее значения, указанного в параметре [Минимальная длина паролей пользователей](#);
- пароль должен состоять из символов, относящихся как минимум к трем из перечисленных групп:
 - заглавные буквы;
 - строчные буквы;
 - цифры;
 - специальные символы;
- пароль не должен совпадать с именем пользователя;
- пароль не должен являться последовательностью символов.

Использование ограничений на пароли пользователей информационной базы не влияет на существующие пароли. Ограничения будут применены только при изменении существующего пароля или при добавлении нового пользователя информационной базы. Однако проверка пароля выполняется по текущей настройке информационной базы. В частности, это означает, что при включении параметра [Проверка сложности паролей пользователей](#) проверка пароля начнет выполняться с учетом регистра.

Например, если для пользователя (по каким-то причинам) был задан пароль **PaRoL**, то до включения параметра [Проверка сложности паролей пользователей](#) пользователь мог вводить пароль и так: **parol**, и так: **PAROL**, и так: **PaRoL**, и всякий раз система разрешала ему доступ в систему. После установки параметра [Проверка сложности паролей пользователей](#) вход в систему будет разрешен только в том случае, если пароль будет введен с учетом регистра: **PaRoL**.

Время засыпания пассивного сеанса (в секундах)

Сеанс, не проявляющий активности в течении указанного времени, переводится в состояние [Спящий](#). Более подробно о сеансах можно посмотреть в книге «1С:Предприятие 8.3. Клиент-серверный вариант. Руководство администратора».

Время завершения спящего сеанса (в секундах)

Спящий сеанс уничтожается по истечении указанного интервала времени. Более подробно о сеансах можно посмотреть в книге «1С:Предприятие 8.3. Клиент-серверный вариант. Руководство администратора».

6.6. Выгрузка информационной базы данных в файл

Текущую информационную базу данных можно сохранить в файл на диске.

Для сохранения данных в файл нужно выбрать пункт [Администрирование – Выгрузить информационную базу данных в файл](#). На экран выводится стандартный диалог выбора файла. Следует выбрать каталог и указать имя файла, в который будут записаны данные.

Механизм выгрузки предназначен:

- для получения образа информационной базы независимо от способа хранения данных;
- для переноса информационной базы из одной СУБД (или файлового варианта) в другую СУБД (или в файловый вариант).

Перед выполнением выгрузки информационной базы рекомендуется выполнить процедуру тестирования (средствами конфигууратора или отдельной утилиты) и исправить все обнаруженные проблемы.

Не рекомендуется использовать данный способ для создания резервной копии информационной базы по следующим причинам:

- может возникнуть ситуация, при которой файл выгрузки будет невозможно загрузить, если в информационной базе, из которой производилась выгрузка, существовали ошибки;
- длительное время создания;
- необходимость монопольного доступа к базе данных;
- высокие требования к оперативной памяти.

ПРИМЕЧАНИЕ. Работа информационной базы в монопольном режиме не переводит базу данных MS SQL в однопользовательский режим (single user).

6.7. Загрузка информационной базы данных из файла

Для восстановления информационной базы данных из файла используется пункт [Администрирование – Загрузить информационную базу данных из файла](#).

На экран выводится стандартный диалог выбора файла. Необходимо выбрать каталог и указать имя файла, в который будут записаны данные.

ВНИМАНИЕ! При восстановлении текущая информационная база данных будет полностью заменена.

ПРИМЕЧАНИЕ 1. Для ускорения процесса загрузки информационной базы, при использовании СУБД Microsoft SQL Server, рекомендуется режим восстановления для базы данных устанавливать в значение [Простой](#) или [С неполным протоколированием](#). Смену режима можно выполнять или перед выполнением загрузки или на постоянной основе, если не требуется выполнять восстановление базы данных на произвольный момент времени. **Перед сменой режима восстановления базы данных необходимо выполнить резервное копирование базы данных!**

ПРИМЕЧАНИЕ 2. Файлы выгрузки информационной базы (.dt), созданные «1С:Предприятием» версий 8.1 и 8.2, могут быть загружены «1С:Предприятием» версии 8.3.

ПРИМЕЧАНИЕ 3. При попытке загрузить конфигурацию с неизвестным режимом совместимости, будет выдаваться ошибка с указанием требуемой версии. Запрещена загрузка файлов **1cv8.dt**, сформированных в версии 8.3.1 и выше, в «1С:Предприятии» младших версий (младше чем 8.3.1). Исключением является ситуация, когда в версии **8.3.1** свойство конфигурации [Режим совместимости](#) установлено в значение [Версия 8.2.16](#).

6.8. Создание резервной копии информационной базы

6.8.1. Файловый вариант информационной базы

ВНИМАНИЕ! Резервное копирование необходимо выполнять перед любой операцией, которая может повредить данные, находящиеся в информационной базе.

ВНИМАНИЕ! Во время выполнения операции резервного копирования информационной базы в файловом варианте к информационной базе не должно быть никаких подключений (в том числе и конфигуратором).

Создание резервной копии осуществляется в любой программе, поддерживающей работу с файлами. С помощью программы работы с файлами необходимо открыть каталог с информационной базой. Для создания копии информационной базы можно просто скопировать файл **1Cv8.1CD** в отдельный каталог. Для восстановления (в случае утери, порчи и т. д.) информационной базы достаточно скопировать сохраненный файл в прежний каталог.

Заметим, что для копирования информационной базы также можно использовать специализированное программное обеспечение, предназначенное для резервного копирования и восстановления данных.

6.8.2. Мобильное устройство

На мобильной платформе резервное копирование следует выполнять штатными средствами на ОС iOS и с помощью специализированных программ для ОС Android.

6.9. Тестирование и исправление информационной базы

В процессе работы системы «1С:Предприятие» могут возникать различные нештатные ситуации – отключение питания компьютера, «зависание» операционной системы, сбой оборудования и прочее. Такие ситуации, возникшие в процессе записи изменений в информационную базу системы «1С:Предприятие» (особенно при работе в файловом варианте), могут привести к ее некорректному состоянию. Внешние проявления некорректного состояния информационной базы могут быть различными, вплоть до невозможности запуска.

Процедура **Тестирование и исправление информационных баз** предназначена для диагностики и устранения ошибочных состояний информационных баз, имеющих различный формат (файловый или клиент-серверный).

Для запуска режима используется пункт **Администрирование – Тестирование и исправление ИБ**. На экран выводится диалог:

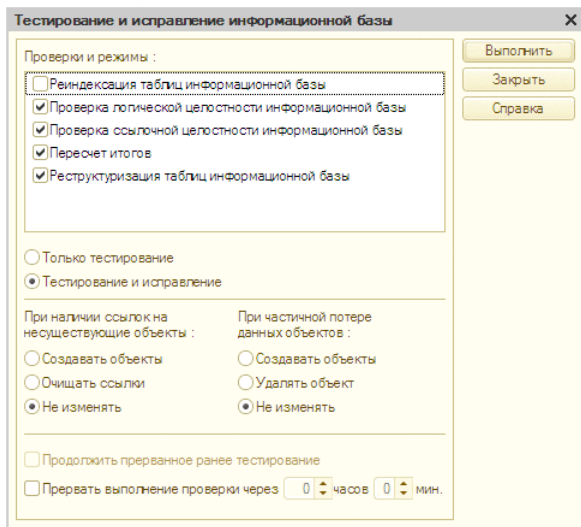


Рис. 47. Тестирование и исправление информационной базы

В списке проверок и режимов следует установить требуемые виды выполняемых действий. Виды тестирования можно производить независимо друг от друга. Для файлового варианта информационной базы возможно выполнение переиндексирования и сжатия базы данных. Для обоих вариантов (файловый и клиент-серверный) возможна проверка логической целостности данных и пересчет итогов.

Для некоторых распределенных информационных баз, у которых возможно получение данных, содержащих ссылки на объекты, не расположенные в тестируемой информационной базе, снятие флажка **Проверка ссылочной целостности информационной базы** позволит отключить создание «несуществующих» данных и как следствие не приведет к передаче этих данных в другие узлы распределенной информационной базы.

Под списком режимов расположены несколько групп настроек:

- В первой группе выбирается, что необходимо выполнить: тестирование или тестирование и исправление. В первом случае программа проведет проверку информационной базы без внесения в нее каких-либо изменений. Во втором случае будут выполнены директивы, указанные во второй группе настроек. Смысл переключателей понятен из их названия.
- Настройки второй группы определяют, что будет делать система при наличии ссылки на несуществующие объекты и при частичной потере данных в существующих объектах.
- Третья группа элементов управления позволяет выполнять длительные процедуры тестирования и исправления в несколько сессий.

Флажок **Прервать выполнение проверки через** позволяет задать интервал времени, по истечении которого тестирование будет прервано, а параметры тестирования и исправления сохранены до следующей сессии конфигулятора.

Флажок **Продолжить прерванное ранее тестирование** позволяет продолжить процесс с того места, на котором он был прерван в предыдущей сессии тестирования и исправления.

События тестирования и исправления отображаются в журнале регистрации.

Для запуска тестирования необходимо нажать кнопку **Выполнить**. Тестирование может быть прервано нажатием комбинации клавиш **Ctrl + Break**.

Программа произведет анализ возможности установки монопольного режима и установит монопольный режим. В случае обнаружения невозможности установки на экран выводится предупреждение: **Не удалось переключить доступ в монопольный режим. Имеются работающие пользователи**. Для получения информации о работающих пользователях нужно открыть список активных пользователей (выбрать пункт **Администрирование – Активные пользователи**, см. [здесь](#)).

Если монопольный режим установлен, запускается процесс выполнения указанных действий и на экран выводится информационная страница диалога о выполнении тестирования.

ПРИМЕЧАНИЕ. Работа информационной базы в монопольном режиме не переводит базу данных MS SQL в однопользовательский режим (single user).

После завершения работы монопольный режим снимается.

В комплект поставки входит утилита восстановления файлового варианта базы данных (**chdbfl.exe**). Описание работы с утилитой см. [здесь](#).

6.10. Отмена назначения главного узла распределенной информационной базы

В случае необходимости отменить назначение главного узла распределенной информационной базы, можно воспользоваться ключом командной строки `/ResetMasterNode` командной строки пакетного запуска конфигурирования. Эта операция равноценна вызову метода `УстановитьГлавныйУзел(Неопределено)` объекта `ПланыОбменаМенеджер`.

Необходимость в таком действии в том случае, например, если возникла необходимость выделить какое-либо из поддеревьев распределенной информационной базы в самостоятельную информационную базу или переподчинить какой-либо из узлов распределенной информационной базы. Более подробную информацию о распределенной информационной базе можно получить в книге «1С:Предприятие 8.3. Руководство разработчика».

6.11. Удаление данных области данных или информационной базы

При необходимости выполнить удаление области данных или всей информационной базы, необходимо воспользоваться ключом `/EraseData` командной строки пакетного запуска конфигурирования. Удаляемая область определяется с помощью параметра `/z` командной строки запуска. Более подробно про удаление области можно прочитать в книге «1С:Предприятие 8.3. Руководство разработчика».

Для удаления данных необходимо, чтобы пользователь, от лица которого выполняется удаление, обладал правом `Администрирование`, и имелась возможность получить монополярный доступ к информационной базе.

ВНИМАНИЕ! Если в сеансе не используется ни один разделитель или удаление данных выполняется в неразделенной информационной базе, будут **удалены все данные информационной базы**.

6.12. Журнал регистрации

Для выполнения административных обязанностей часто требуется выяснить, какие события происходили в определенный момент времени или какие действия выполнял тот или иной пользователь.

Для этих целей предназначен журнал регистрации. В этом журнале могут фиксироваться различные события. С его помощью администратор может получить историю работы пользователей с системой.

Журнал регистрации не хранится в базе данных и не сохраняется при операциях загрузки/выгрузки информационной базы.

При работе пользователей система «1С:Предприятие» фиксирует в журнале основные действия, выполняемые пользователем по модификации данных информационной базы, выполнению регламентных операций, подключению и отключению от системы и т. д.

Журнал регистрации работает как в режиме `Конфигуратор`, так и в режиме `1С:Предприятие`. Описание работы с журналом регистрации в режиме `1С:Предприятие` см. [здесь](#).

Журнал регистрации может формироваться в двух форматах:

- Формат `.lgd` используется в «1С:Предприятие» версии 8.3.5 и старше как формата журнала регистрации по умолчанию для новых информационных баз, а также после конвертации из формата `.lgf`. Более подробную информацию см. [здесь](#).
- Формат `.lgf` используется в версиях «1С:Предприятия» младше 8.3.5 и в «1С:Предприятие» версии 8.3.5 до выполнения конвертации в формат `.lgd`. Более подробную информацию см. [здесь](#).

6.12.1. Журнал регистрации формата .lgd

Журнал регистрации формата `.lgd` хранится в файле базы данных формата SQLite. Журнал располагается:

- Для файлового варианта информационной базы – в подкаталоге `1Cv8Log` каталога информационной базы.
- Для клиент-серверного варианта информационной базы – в подкаталоге `1Cv8Log` каталога информационной базы в каталоге служебных файлов кластера. Имя каталога можно определить по файлу реестра данных кластера.

При создании новой информационной базы журнал регистрации имеет формат `.lgd`.

6.12.1.1. Настройка журнала регистрации

С помощью пункта меню `Администрирование – Настройка журнала регистрации` можно настроить учет событий в журнале регистрации.

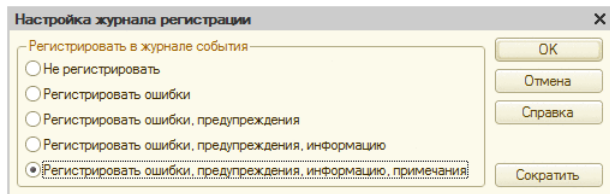


Рис. 48. Настройка журнала регистрации

В случае сетевой работы выбранную установку можно сохранить только тогда, когда с конфигурацией кроме администратора никто не работает.

Записи журнала регистрации хранятся в файле формата SQLite.

В процессе эксплуатации системы в журнале регистрации может накопиться значительное число записей. Для сокращения числа записей следует открыть окно настройки журнала и нажать кнопку `Сократить`. На экран выводится окно:

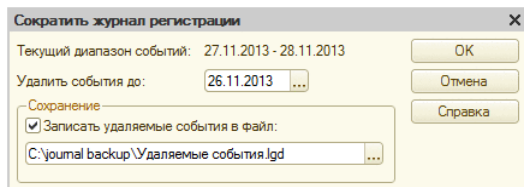


Рис. 49. Сокращение журнала регистрации

Сокращение записей производится до указанной в поле `Удалить события до`. Если требуется сохранить удаляемые записи, то нужно установить флажок `Записать удаляемые записи в файл` и указать имя файла-архива. Имеется возможность прервать операцию сокращения журнала регистрации. Для этого следует нажать сочетание клавиш `Ctrl+Break`. При этом часть записей будет удалена, а часть – останется неудаленной.

Для просмотра архива записей журнала регистрации нужно выбрать пункт `Файл – Открыть` и в стандартном диалоге выбора файла указать тип файла `Журнал регистрации (*.lgd;*.lgf)`. Выбрать нужный файл архива и нажать кнопку `Открыть`.

Настройка автоматического обновления и интервала обновления производится стандартным для табличного поля механизмом настройки списка.

События в журнале регистрации идентифицируются строкой. При этом для системных событий используются комбинации символов `_$` и `$_` (например, `$_InfoBase$_MasterNodeUpdate` или `$_PerformError$_`). `$_InfoBase$_MasterNodeUpdate` отобразится в виде строки `Информационная база. Изменение главного узла`. Использование этих комбинаций в именах событий, записываемых из встроеного языка, с помощью метода `ЗаписьЖурналаРегистрации()` запрещено. Созданные при помощи этого метода события отображаются как есть.

6.12.1.2. Сохранение журнала регистрации

Для сохранения журнала регистрации нужно открыть его и выбрать пункт **Файл – Сохранить копию**. На экран выводится диалог выбора каталога и файла, в который будет произведена выгрузка, а также тип файла (по умолчанию указывается тип журнала регистрации *.lgf). Выгрузка также возможна в формате XML (описание формата см. [здесь](#)).

Пример выгрузки журнала регистрации:

[Копировать в буфер обмена](#)

```
<v8e:EventLog
  xmlns:v8e = "http://v8.lc.ru/eventLog"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance">
  <v8e:Event>
  <v8e:Level>Warning</v8e:Level>
  <v8e:Date>Дата события</v8e:Date>
  <v8e:Application>Enterprise</v8e:Application>
  <v8e:ApplicationPresentation>
    1С:Предприятие</v8e:ApplicationPresentation>
  <v8e:EventName>имя события</v8e:EventName>
  <v8e:EventPresentation>
    презентация события</v8e:EventPresentation>
  <v8e:UserID>00000000-0000-0000-0000-000000000001</v8e:UserID>
  <v8e:UserName>Иванов</v8e:UserName>
  <v8e:Computer>Ivanov</v8e:Computer>
  <v8e:MetadataName>Справочники.Номенклатура</v8e:MetadataName>
  <v8e:MetadataPresentation>
    Справочники Номенклатура</v8e:MetadataPresentation>
  <v8e:Comment>Комментарий</v8e:Comment>
  <v8e>Data xsi:type="xsd:string">Какие-то данные</v8e>Data>
  <v8e:DataPresentation>Описание данных</v8e:DataPresentation>
  </v8e:Event>
</v8e:EventLog>
```

6.12.2. Журнал регистрации формата .lgf

Журнал регистрации формата .lgf хранится в текстовых файлах с расширениями .lgf (общая информация журнала регистрации) и .lgp (фрагмент журнала регистрации). Журнал располагается:

- Для файлового варианта информационной базы – в подкаталоге **1Cv8Log** каталога информационной базы.
- Для клиент-серверного варианта информационной базы – в подкаталоге **1Cv8Log** каталога информационной базы в каталоге служебных файлов кластера. Имя каталога можно определить по файлу реестра данных кластера.

6.12.2.1. Настройка журнала регистрации

С помощью пункта меню **Администрирование – Настройка журнала регистрации** можно настроить учет событий в журнале регистрации.

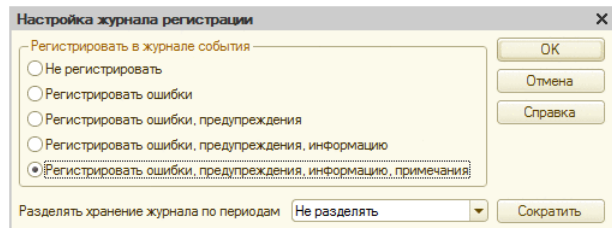


Рис. 50. Настройка журнала регистрации

В случае сетевой работы выбранную установку можно сохранить только тогда, когда с конфигурацией кроме администратора никто не работает.

Записи журнала регистрации хранятся в файлах. Каждый файл содержит записи определенного периода. Размер периода задается в поле **Разделять хранение журнала регистрации по периодам**. Новый файл открывается при наступлении (указывается в значении настройки) каждого нового:

- часа,
- дня,
- недели,
- месяца,
- года.

При создании новой информационной базы для журнала устанавливается режим регистрации событий всех уровней важности и периодичность разделения на файлы – день.

В процессе эксплуатации системы в журнале регистрации может накопиться значительное число записей. Для сокращения числа записей следует открыть окно настройки журнала и нажать кнопку **Сократить**. На экран выводится окно:

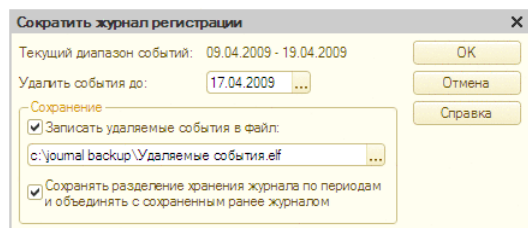


Рис. 51. Сокращение журнала регистрации

Сокращение записей производится до указанной в поле **Удалить события до**. При этом следует учитывать тот факт, что будут удалены все записи периода разделения журнала регистрации (см. выше описание поля **Разделять хранение журнала регистрации по периодам**), в который попадет указанная дата. Так, если указано разделение журнала по месяцам и указана дата **14.05.2009** (например), то будут удалены записи журнала регистрации по май 2009 года (включительно). Также следует помнить, что период разделения журнала регистрации может меняться (с течением времени), и удаляемый период будет определяться не текущим периодом разбиения, а тем периодом, который был установлен «во время» даты, указанной в поле **Удалить события до**.

Если требуется сохранить удаляемые записи, то нужно установить флажок **Записать удаляемые записи в файл** и указать имя файла-архива.

Если требуется периодически сокращать журнал и при этом иметь возможность просматривать уже удаленные события журнала, то следует установить флажок **Сохранять разделение хранения журнала по периодам и объединять с сохраненным ранее журналом**.

СОВЕТ. Для сохранения разделения по периодам при запуске конфигуратора в командном режиме можно также использовать команду `/ReduceEventLogSize KeepSplitting`.

Для просмотра архива записей журнала регистрации нужно выбрать пункт **Файл – Открыть** и в стандартном диалоге выбора файла указать тип файла **Журнал регистрации (*.lgf)**. Выбрать нужный файл архива и нажать кнопку **Открыть**.

Настройка автоматического обновления и интервала обновления производится стандартным для табличного поля механизмом настройки списка.

События в журнале регистрации идентифицируются строкой. При этом для системных событий используются комбинации символов `_$_` и `$__` (например, `$_InfoBase$_MasterNodeUpdate` или `$_PerformError$_`). `$__InfoBase$_MasterNodeUpdate` отобразится в виде строки **Информационная база. Изменение главного узла**. Использование этих комбинаций в именах событий, записываемых из встроенного языка, с помощью метода `ЗаписьЖурналаРегистрации()` запрещено. Созданные при помощи этого метода события отображаются как есть.

6.12.2.2. Сохранение журнала регистрации

Для сохранения журнала регистрации нужно открыть его и выбрать пункт **Файл – Сохранить копию**. На экран выводится диалог выбора каталога и файла, в который будет произведена выгрузка, а также тип файла (по умолчанию указывается тип журнала регистрации `*.lgf`). Выгрузка также возможна в формате XML (описание формата см. [здесь](#)).

Пример выгрузки журнала регистрации:

```
<v8e:EventLog
  xmlns:v8e = "http://v8.1c.ru/eventLog"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance">
  <v8e:Event>
    <v8e:Level>Warning</v8e:Level>
    <v8e:Date>Дата события</v8e:Date>
    <v8e:Application>Enterprise</v8e:Application>
    <v8e:ApplicationPresentation>
      1С:Предприятие</v8e:ApplicationPresentation>
    <v8e:EventName>Имя события</v8e:EventName>
    <v8e:EventPresentation>
      презентация события</v8e:EventPresentation>
    <v8e:UserID>00000000-0000-0000-0000-000000000001</v8e:UserID>
    <v8e:UserName>Иванов</v8e:UserName>
    <v8e:Computer>Ivanov</v8e:Computer>
    <v8e:MetadataName>Справочники.Номенклатура</v8e:MetadataName>
    <v8e:MetadataPresentation>
      Справочники Номенклатура</v8e:MetadataPresentation>
    <v8e:Comment>Комментарий</v8e:Comment>
    <v8e>Data xsi:type="xsd:string">Какие-то данные</v8e>Data>
    <v8e:DataPresentation>Описание данных</v8e:DataPresentation>
  </v8e:Event>
</v8e:EventLog>
```

Копировать в буфер обмена

6.12.2.3. Конвертация журнала регистрации в формат .lgd

После установки «1С:Предприятие» версии 8.3.5 и старше, появляется возможность сменить формат журнала регистрации на формат `.lgd` (см. [здесь](#)). Для этого следует открыть диалог настройки журнала регистрации (**Главное меню – Администрирование – Настройка журнала регистрации**) и нажать кнопку **Новый формат**.

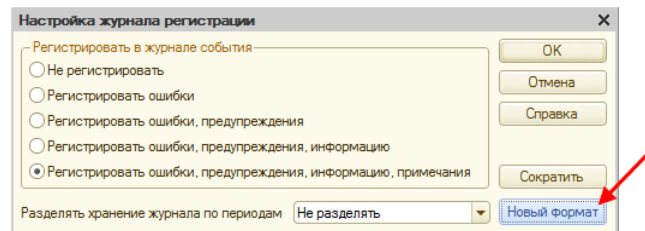


Рис. 52. Конвертация в формат .lgd

Конвертация может быть выполнена только в новый формат (формат `.lgd`). Конвертация в «старый» формат (формат `.lgf`) не поддерживается.

ПРИМЕЧАНИЕ. При большом объеме журнала регистрации, конвертация может занимать существенное время.

6.13. Технологический журнал

Система «1С:Предприятие» обеспечивает возможность ведения технологического журнала, в который помещается информация от всех приложений, относящихся к системе «1С:Предприятие».

Технологический журнал предназначен для выявления ошибок, возникающих при эксплуатации системы, и диагностики работы системы службой технической поддержки фирмы «1С», а также для анализа технологических характеристик работы системы.

Состав и свойства событий технологического журнала могут меняться при выпуске обновлений платформы.

Поскольку технологический журнал представляет собой набор текстовых файлов, хранящихся в различных каталогах, он может быть использован разработчиками прикладных решений для анализа различных режимов работы системы «1С:Предприятие» и прикладных решений.

Технологический журнал может вестись на любом компьютере, на котором установлена система «1С:Предприятие». За ведение технологического журнала отвечает конфигурационный файл, в котором описываются:

- каталог, в котором будут располагаться файлы технологического журнала;
- состав информации, которая будет помещаться в технологический журнал;
- время, в течение которого хранятся файлы технологического журнала;
- параметры дампа, создаваемого при аварийном завершении приложения.

По умолчанию конфигурационный файл отсутствует. Это означает, что технологический журнал включен и настроен на сохранение минимальных дампов при аварийном завершении приложения в каталог:

```
%USERPROFILE%\Local Settings\Application Data\1C\1Cv8\dumps
```

Копировать в буфер обмена

Для ОС Windows Vista и выше каталог будет иметь вид:

```
%LOCALAPPDATA%\1C\1Cv8\dumps
```

Копировать в буфер обмена

При необходимости может быть выполнена произвольная настройка журнала регистрации с помощью отдельного конфигурационного файла. Этот файл должен иметь имя `logcfg.xml` и располагаться в каталоге конфигурационных файлов системы «1С:Предприятие». Подробнее о структуре и возможностях конфигурационного файла см. [здесь](#).

ПРИМЕЧАНИЕ. Для работы технологического журнала в ОС Windows необходимо, чтобы пользователь процесса, который пишет технологический журнал, имел полные права на каталог технологического журнала и права на чтение владельца каталога технологического журнала.

Система «1С:Предприятие» автоматически, с периодичностью 60 секунд, опрашивает каталоги конфигурационных файлов на предмет наличия файла `logcfg.xml` и анализирует его состав. Таким образом, изменение параметров технологического журнала может быть выполнено на ходу, без перезапуска работающих приложений системы «1С:Предприятие».

При определенных настройках объем технологического журнала может быть достаточно большим, поэтому в конфигурационном файле желательно указывать время, в течение которого хранятся файлы журнала. По истечении указанного времени система «1С:Предприятие» удалит устаревшие файлы журнала. Если после удаления устаревших файлов каталог, в котором располагались эти файлы, оказывается пустым, то такой каталог тоже удаляется. Таким образом, все дерево каталогов технологического журнала не содержит устаревших файлов и папок.

ВНИМАНИЕ! Если работа системы выполняется в среде Linux, управление выдачей аварийных дампов выполняется средствами операционной системы. При этом в технологический журнал помещается информация о факте аварийного завершения процесса и о номере сигнала, повлекшего за собой это завершение. Подробнее о настройке журнала для ОС Linux см. [здесь](#).

ВНИМАНИЕ! Необходимо иметь в виду, что каталог технологического журнала не предназначен для хранения в нем файлов, которые не относятся к технологическому журналу. Поэтому не следует размещать в нем дампы или использовать каталог, который может содержать файлы, не относящиеся к технологическому журналу «1С:Предприятия». Если в каталоге, который указан в качестве каталога технологического журнала, имеются посторонние файлы, то указание каталога считается неверным, и технологический журнал не создается.

6.13.1. Конфигурационный файл технологического журнала

В простейшем виде конфигурационный файл может иметь, например, следующее содержимое:

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
  <log location="c:\1c\logs" history="1">
    <event>
      <eq property="name" value="conn" />
    </event>
  </log>
  <dump location="c:\1c\dumps" create="1" type="2" />
</config>
```

[Копировать в буфер обмена](#)

Данный конфигурационный файл указывает на следующее:

- в технологическом журнале регистрируются все события установки и разрыва клиентского соединения с сервером;
- файлы технологического журнала располагаются в каталоге `c:\1c\logs`;
- файлы технологического журнала хранятся в течение одного часа;
- файлы дампа помещаются в каталог `c:\1c\dumps`;
- файлы дампа содержат всю доступную информацию (содержимое всей памяти процесса).

При отсутствии конфигурационного файла используются следующие параметры:

- Технологический журнал – выключен.
- Технологический журнал по умолчанию – включен (см. [здесь](#)).
- Дампы минимального размера.
- Дампы сохраняются в каталог `%USERPROFILE%\Local Settings\Application Data\1C\1Cv8\dumps` профиля текущего пользователя (или `%LOCALAPPDATA%\1C\1Cv8\dumps` для ОС Windows Vista и старше).

Подробнее о структуре и возможностях конфигурационного файла см. [здесь](#).

Работа с конфигурационным файлом в ОС Linux практически не отличается от таковой в ОС Windows, за исключением следующих особенностей:

- Файл должен располагаться в каталоге конфигурационных файлов системы «1С:Предприятие».
- Каталог, в котором будет формироваться технологический журнал, должен быть доступен по записи для пользователя, от имени которого работает приложение (сервер, клиентские приложения, расширения веб-сервера и т. д.), формирующее технологический журнал.

6.13.2. Технологический журнал по умолчанию

Для записи событий, возникающих в критических ситуациях (с точки зрения системы «1С:Предприятие») предназначен технологический журнал по умолчанию. Для этого журнала создается фиксированный фильтр событий, который формирует платформа и который невозможно изменить.

Технологический журнал по умолчанию имеет следующие настройки:

- Каталог файлов технологического журнала по умолчанию:
 - ОС Windows: `%USERPROFILE%\Local Settings\Application Data\1C\1Cv8\logs` (или `%LOCALAPPDATA%\1C\1Cv8\logs` для ОС Windows Vista и старше).
 - ОС Linux: `~/1c8/logs`.
- Информация удаляется из технологического журнала по умолчанию через 24 часа.
- В технологический журнал по умолчанию попадают события `SYSTEM` с уровнем `Error`.

Указанные настройки можно изменить с помощью элемента `<defaultlog>` (см. [здесь](#)). Настройка правил формирования событий, которые регистрируются в технологическом журнале по умолчанию выполняется с помощью элемента `<system>` (см. [здесь](#)).

6.13.3. Структура технологического журнала

Технологический журнал представляет собой каталог, в подкаталогах которого располагаются файлы с собранными технологическими данными. Каталог журнала имеет следующую структуру:

```
<каталог журнала>
  <идентификатор процесса операционной системы>
    <файлы журнала одного процесса>
```

[Копировать в буфер обмена](#)

Каждый файл журнала содержит события за 1 час и имеет имя `yyymmddhh.log`, где:

- `yy` – две последние цифры года;
- `mm` – номер месяца;
- `dd` – номер дня;

- **hh** – номер часа.

Файлы журнала имеют текстовый формат. В файле сведения о завершении каждого события записываются с новой строки.

Например:

```
16:08.8750-9060,CALL,0,process=rphost,p:processName=DebugControlCenter,t:clientID=221,t:applicationName=Debugger,t:computerName=COMP1,Interface=5cf29e71-ec:
16:08.8911-1,DBPOSTGRS,2,process=rphost,p:processName=Database,t:clientID=216,t:applicationName=1CV8,t:computerName=COMP1,t:connectID=125,Usr=User2,Trans:
16:08.8913-1,DBPOSTGRS,2,process=rphost,p:processName=Database,t:clientID=216,t:applicationName=1CV8,t:computerName=COMP1,t:connectID=125,Usr=User2,Trans=
```

[Копировать в буфер обмена](#)

Строка окончания события имеет формат: **mm:ss.ttttt-d**, <наименование>, <уровень>, <ключевые свойства>, где:

- **mm** – номер минуты в текущем часе.
- **ss** – номер секунды в текущей минуте.
- **ttttt** – номер микросекунды текущей секунды.
- **d** – длительность события в микросекундах.
- <наименование> – наименование события.
- <уровень> – уровень события в стеке текущего потока.
- <ключевые свойства> – <ключевое свойство>, <ключевое свойство>, ...
- <Ключевое свойство> – <имя> = <значение>; <наименование>, <имя>, <значение> – произвольный текст. Если в нем присутствуют символы «конец строки» или «запятая», то текст заключается в кавычки или апострофы, в зависимости от того, каких символов в строке меньше, а кавычки или апострофы в тексте удваиваются.

6.13.4. Настройка формирования дампов памяти

6.13.4.1. Для ОС Windows

Данный раздел содержит пример настройки файла конфигурации технологического журнала (**logcfg.xml**), необходимой для создания дампов памяти аварийного завершения.

[Копировать в буфер обмена](#)

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
  <dump location="C:\Program Files\1cv8\dumps" create="1" type="3"/>
</config>
```

При такой настройке дампы памяти будут формироваться в каталоге **C:\Program Files\1cv8\dumps** и построенные дампы будут включать содержимое всей памяти процесса и дополнительный сегмент данных.

Пользователь, от чьего имени выполняется клиентское приложение или сервер, должен иметь **полные права** на каталоги:

- каталог временных файлов,
- каталог технологического журнала,
- каталог дампов.

Пользователь, от чьего имени выполняется клиентское приложение или сервер, должен иметь **право на чтение** каталогов:

- конфигурационных файлов (см. [здесь](#));
- каталога-владельца каталога дампов.

Если в файле **logcfg.xml** выполнена настройка получения планов запросов, то такой файл должен располагаться в каталоге конфигурационных файлов соответствующего приложения:

- для клиент-серверного варианта – в каталоге конфигурационных файлов, доступных серверу «1С:Предприятия»;
- для файлового варианта с прямым подключением – в каталоге конфигурационных файлов, доступных нужной версии клиентского приложения;
- для файлового варианта с подключением через веб-сервер – в каталоге конфигурационных файлов, доступных расширению веб-сервера, которое обслуживает данную информационную базу.

Подробнее о настройке файла **logcfg.xml** см. [здесь](#).

6.13.4.2. Для ОС Linux

Данный раздел описывает шаги по настройке операционной системы Linux для обеспечения создания дампов памяти при аварийном завершении программы.

ПРИМЕЧАНИЕ. Рекомендации, изложенные в данном разделе, в полной мере применимы для ОС Fedora Core 4 и ее аналогов. Для остальных дистрибутивов ОС Linux возможно другое название и синтаксис описываемых здесь команд. За подробностями следует обратиться к справочной системе используемого дистрибутива ОС Linux.

По умолчанию создание дампов аварийного завершения отключено. Поставщики дистрибутивов Linux рекомендуют включать создание дампов только на компьютерах, предназначенных для разработки, но не на компьютерах, используемых для реальной работы программы.

6.13.4.2.1. Включение автоматической генерации дампов

Формирование дампов аварийного завершения настраивается для всех процессов, исполняемых от лица конкретного пользователя. Для того, чтобы включить автоматическую генерацию дампов, необходимо в файл **/etc/security/limits.conf** добавить следующие строки:

[Копировать в буфер обмена](#)

```
<username> soft core unlimited
<username> hard core unlimited
```

Где **<username>** – это имя пользователя, от лица которого выполняется приложение системы «1С:Предприятие».

6.13.4.2.2. Определение имени и местоположения дампов

Для более четкого понимания того, каким процессом был сгенерирован дамп аварийного завершения, а также для размещения дампов в требуемом каталоге диска рекомендуется задать шаблон формирования имени дампа. Задание шаблона может осуществляться как в рамках одной сессии, так и на постоянной основе.

ВНИМАНИЕ! Настройка, выполняемая в данном разделе, оказывает влияние на все процессы всех пользователей операционной системы. Это означает, что дампы аварийного завершения других пользователей (если их генерация включена) будут сохраняться по указанному пути с выбранным шаблоном имени.

ВНИМАНИЕ! Описанные ниже действия необходимо выполнять от имени пользователя **root**.

Для задания шаблона имени и пути расположения дампов аварийного завершения нужно воспользоваться командой:

[Копировать в буфер обмена](#)

```
sysctl -w kernel.core_pattern=/tmp/core.%.%p
```

Эта настройка будет действовать до следующей перезагрузки компьютера. В этом случае дампы будут размещаться в каталоге `/tmp` и имя дампов будет формироваться:

- Из префикса `core`;
- Имени исполняемого файла;
- Идентификатора процесса, для которого был сформирован дамп аварийного завершения.

Для указания шаблона имени и пути на постоянной основе необходимо добавить следующую строку в файл `/etc/sysctl.conf`:

```
kernel.core_pattern=/tmp/core.%.%p
```

[Копировать в буфер обмена](#)

Для того чтобы сделанные в файле изменения вступили в силу, необходимо выполнить команду:

```
sysctl -p
```

[Копировать в буфер обмена](#)

Путь, указанный в настройках, должен быть доступен для записи тем пользователям, от лица которых работают приложения, формирующие дампы аварийного завершения.

6.13.5. Примеры файлов настройки технологического журнала

В приведенных ниже примерах предполагается, что «1С:Предприятие» установлено стандартным способом в каталог `C:\Program Files\1cv8`.

Важно иметь в виду, что в каталог технологического журнала при некоторых его настройках могут выводиться данные очень большого объема. Поэтому на диске, где планируется размещение файлов технологического журнала, должно быть достаточно свободного места.

Далее приведено несколько примеров файлов `logcfg.xml`, описывающих наиболее часто используемые конфигурации технологического журнала.

6.13.5.1. Технологический журнал выключен

Если файл `logcfg.xml` отсутствует в каталоге конфигурационных файлов (см. [здесь](#)) «1С:Предприятия», то технологический журнал не создается. Если файл `logcfg.xml` необходим для правильной настройки дампов, то он не должен содержать ни одного элемента `log`. Следующий пример определяет необходимость построения полного дампа приложения при его аварийном завершении. Дампы помещаются в каталог `C:\v8\dumps`.

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
  <dump location="C:\v8\dumps" create="1" type="3"/>
</config>
```

[Копировать в буфер обмена](#)

6.13.5.2. Полный технологический журнал

Приведенный ниже конфигурационный файл определяет вывод в технологический журнал всех событий вместе со всеми свойствами. Журнал будет сохраняться в течение недели (168 часов). Объем выводимой информации при этом будет очень большим, однако она может быть полезна при анализе сложных нештатных ситуаций. Данную конфигурацию рекомендуется использовать на этапе тестирования и при расследовании ошибок.

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
  <log location="C:\v8\logs" history="168">
    <event>
      <ne property="name" value="" />
    </event>
    <property name="all">
    </property>
  </log>
</config>
```

[Копировать в буфер обмена](#)

6.13.5.3. Обращения к СУБД

Следующий конфигурационный файл определяет, что технологический журнал будет содержать только обращения «1С:Предприятия» СУБД, а также информацию об ошибочных ситуациях. Объем выводимой информации меньше, чем при полном технологическом журнале, но тоже может быть очень большим.

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
  <log location="C:\v8\logs" history="168">
    <event>
      <eq property="name" value="dbmssql"/>
    </event>
    <event>
      <eq property="name" value="dbpostgrs"/>
    </event>
    <event>
      <eq property="name" value="db2"/>
    </event>
    <event>
      <eq property="name" value="dboracle"/>
    </event>
    <event>
      <eq property="name" value="excp"/>
    </event>
    <property name="all">
    </property>
  </log>
</config>
```

[Копировать в буфер обмена](#)

6.13.5.4. Действия администратора и ошибки

Этот конфигурационный файл создает технологический журнал небольшого объема, в котором содержится информация о запуске и завершении приложений, установке и разрыве соединений с кластером серверов «1С:Предприятия», действиях администратора кластера и об ошибочных ситуациях в работе «1С:Предприятия». Такой журнал в большинстве случаев достаточен для расследования ошибочных ситуаций как в конфигурации, так и в технологической платформе «1С:Предприятие».

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
<log location="C:\v8\logs" history="168">
  <event>
    <eq property="name" value="admin"/>
  </event>
  <event>
    <eq property="name" value="conn"/>
  </event>
  <event>
    <eq property="name" value="excp"/>
  </event>
  <event>
    <eq property="name" value="proc"/>
  </event>
  <event>
    <eq property="name" value="gerr"/>
  </event>
</log>
</config>
```

[Копировать в буфер обмена](#)

```

<event>
  <eq property="name" value="scom" />
</event>
<property name="all" />
</log>
</config>

```

6.13.5.5. Ошибки и долгие операции

По сравнению с предыдущим, данный конфигурационный файл добавляет все операции, длительность которых превышает 10 секунд. Это может оказаться полезным для обнаружения действий пользователей, которые выполнялись длительное время, с целью, например, их последующей оптимизации. Длительность событий выражается в сотнях микросекунд.

[Копировать в буфер обмена](#)

```

<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="http://v8.1c.ru/v8/tech-log">
  <dump create="false" />
  <log location="C:\v8\logs" history="168">
    <event>
      <eq property="name" value="admin" />
      <gt property="duration" value="100000" />
    </event>
    <event>
      <eq property="name" value="conn" />
      <gt property="duration" value="100000" />
    </event>
    <event>
      <eq property="name" value="excp" />
      <gt property="duration" value="100000" />
    </event>
    <event>
      <eq property="name" value="proc" />
      <gt property="duration" value="100000" />
    </event>
    <event>
      <eq property="name" value="qerr" />
      <gt property="duration" value="100000" />
    </event>
    <event>
      <eq property="name" value="scom" />
      <gt property="duration" value="100000" />
    </event>
  <property name="all" />
</log>
</config>

```

6.14. Контроль ссылочной целостности

В системе «1С:Предприятие» значительная часть данных хранится в виде **ссылок**. Например, при вводе документов многие реквизиты документа могут заполняться путем выбора значения из списка или документа из списка документов. Такие реквизиты являются **ссылками** на элементы соответствующих списков.

Использование ссылок позволяет избежать многократного исправления одной и той же информации в разных местах. Например, после ввода и распечатки ряда документов выяснилось, что наименование организации-контрагента, на которую были выписаны эти документы, указано неправильно. Так как наименование контрагента вводилось в документы путем выбора из списка контрагентов, достаточно отредактировать наименование контрагента только в списке – измененное наименование будет отражено в документах автоматически, и достаточно будет только заново построить печатные формы.

Однако если удалить организацию-контрагента из списка, то во всех документах, в которых она использовалась, останутся так называемые «неразрешенные ссылки» – ссылки на несуществующий объект.

Для исключения таких ситуаций в системе «1С:Предприятие» существует механизм **контроля ссылочной целостности**, о котором пойдет речь в этом разделе.

6.14.1. Основные понятия

Механизм контроля ссылочной целостности разделяет процесс удаления объектов данных, на которые могут существовать ссылки (списки и документы), на два этапа.

На первом этапе пользователи выполняют пометку объектов на удаление. При этом помеченный на удаление объект практически ничем не отличается в использовании от обычного объекта.

На втором этапе администратор системы или иное лицо, для которого определены соответствующие права (установлено право [Интерактивное удаление помеченных](#) для соответствующих видов списков и документов), выполняет специальную процедуру – удаление помеченных объектов, которая реализована в виде стандартной функции [Удаление помеченных объектов](#) (подробнее см. [здесь](#)). В ходе выполнения этой процедуры происходит полный анализ всех ссылок на помеченные объекты, и могут быть удалены только те объекты, на которые ссылки либо отсутствуют, либо располагаются в объектах, которые также помечены для удаления.

Фактически процедура удаления помеченных объектов является регламентной. Ее рекомендуется выполнять периодически по мере накопления помеченных объектов.

6.14.2. Включение режима контроля ссылочной целостности

Система «1С:Предприятие» позволяет удалять лишнюю или устаревшую информацию в двух режимах:

- **Непосредственное удаление объектов** – не производится анализ использования удаляемого объекта в других объектах базы данных.
- **Использование контроля ссылочной целостности** – объекты сначала помечаются на удаление, а затем производится контроль наличия ссылок на эти объекты в других объектах.

ВНИМАНИЕ! Установка прав удаления (непосредственное удаление или использование контроля ссылочной целостности) производится для каждой роли, назначаемой пользователям, по каждому виду объектов (списков и документов) на этапе проектирования прикладного решения.

Если пользователь работает в режиме непосредственного удаления, то дополнительная ответственность ложится и на пользователя, выполняющего удаление объектов, и на администратора системы, определяющего права пользователей и действия системы при неразрешенных ссылках. Работа системы без контроля ссылочной целостности может быть, например, использована специалистами в процессе отладки прикладного решения. Если контроль ссылочной целостности не используется, то удаление объектов происходит непосредственно (без пометки на удаление), и появляется возможность образования неразрешенных ссылок.

Самым радикальным способом установки режима контроля ссылочной целостности является отключение в конфигурации в целом прав непосредственного удаления объектов. Таким способом полностью исключается возможность в пределах данного прикладного решения непосредственно удалять объекты. Пользователи будут иметь возможность только помечать объекты на удаление.

Предоставление прав на непосредственное удаление, а также установку и снятие пометки удаления объектов производится для каждого вида объектов конфигурации. Если по данному виду для выбранного набора прав (роли) установлено право [Интерактивное удаление](#), то пользователи, для которых определена эта роль, имеют возможность непосредственного удаления объектов данного вида. Установка прав производится при разработке прикладного решения.

Аналогично предоставляется право установки и снятия пометки удаления объектов, а также удаления помеченных объектов.

Разумеется, только отключение в конфигурации права [Интерактивное удаление](#) обеспечивает согласованное использование механизма ссылочной целостности всеми пользователями.

ВНИМАНИЕ! Заметим, что также существует возможность непосредственного удаления объектов средствами встроенного языка. Поэтому элементы конкретной конфигурации могут выполнять непосредственное удаление в обход механизма контроля ссылочной целостности. В этом случае ответственность за целостность данных лежит на специалисте, выполняющем разработку конкретного механизма системы.

6.14.3. Непосредственное удаление объектов

Если режим контроля ссылочной целостности не используется (в конфигурации установлено право [Интерактивное удаление](#) у конкретного пользователя для конкретного вида объекта конфигурации), в списках списков и в журналах документов пользователь имеет возможность использовать пункт меню [Удалить непосредственно](#) (клавишу **Shift + Del** или соответствующую кнопку панели инструментов) для удаления данного вида объектов. При этом данный объект будет удален без проверки ссылок на него в других объектах.

6.14.4. Установка и снятие пометки удаления

При использовании механизма контроля ссылочной целостности в списках списков и журналах документов в меню [Еще \(Все действия\)](#) присутствует пункт [Пометить на удаление/Снять пометку](#). При выборе данного пункта меню выполняется пометка объекта на удаление. Помеченный на удаление объект отмечается в левой колонке списка значком, изображающим перечеркнутый образ объекта.

ВНИМАНИЕ! При пометке на удаление проведенного документа он становится непроведенным.

Выбор пункта меню [Еще – Пометить на удаление/Снять пометку удаления \(Все действия – Пометить на удаление/Снять пометку удаления\)](#) помечает объект на удаление, а для помеченного на удаление объекта снимает у него пометку удаления.

ВНИМАНИЕ! При снятии пометки удаления у документа он не становится проведенным. Для того чтобы он стал проведенным, документ следует провести.

Возможность установки и снятия пометки удаления конкретным пользователем также регламентируется правами доступа (отдельно пометка и снятие пометки).

6.14.5. Особенности использования объектов, помеченных на удаление

В основном помеченные на удаление объекты используются так же, как и обычные. Они также показываются в списках, по ним может выполняться поиск и т. д. Помеченные на удаление объекты могут быть открыты и изменены.

Помеченный на удаление документ не может быть проведен. При попытке выполнить проведение помеченного на удаление документа выдается соответствующее сообщение, и проведение документа не выполняется.

6.15. Стандартные функции

Стандартные функции – это набор инструментов системы, предназначенный для выполнения различных сервисных операций, которые могут потребоваться при выполнении действий по администрированию информационной базы.

Доступ к стандартным функциям возможен только в режиме 1С:Предприятие. Для получения доступа к стандартным функциям необходимо включить соответствующий параметр в окне настроек ([Сервис – Параметры – Отображать команду "Все функции"](#)).

ПРИМЕЧАНИЕ. Для окон стандартных функций не поддерживается получение навигационных ссылок, и они не могут быть добавлены в избранные работы пользователя.

Ниже приведен полный список стандартных функций с кратким описанием.

Название	Краткое описание
Активные пользователи	Отображает список пользователей, работающих в данный момент с системой «1С:Предприятие». Доступность функции определяется правом Активные пользователи
Журнал регистрации	Позволяет просматривать журнал регистрации. Доступность функции определяется правом Журнал регистрации
Поиск ссылок на объекты	Позволяет найти объекты, ссылающиеся на какой-либо выбранный объект
Проведение документов	Позволяет выполнять проведение и перепроведение документов за выбранный период, а также восстанавливать последовательности, существующие в конфигурации
Удаление помеченных объектов	Позволяет выполнить удаление объектов, помеченных на удаление
Управление итогами	Позволяет выполнять регламентные операции с регистрами
Управление полнотекстовым поиском	Позволяет управлять полнотекстовым поиском

Для вызова необходимой стандартной функции следует открыть окно [Все функции](#), выбрать ветвь [Стандартные функции](#) и в открывшемся списке выбрать нужную стандартную функцию (если она доступна).

Далее подробно описаны все стандартные функции.

6.15.1. Список активных пользователей

На экран выводится окно со списком пользователей, работающих в данный момент с базой данных.

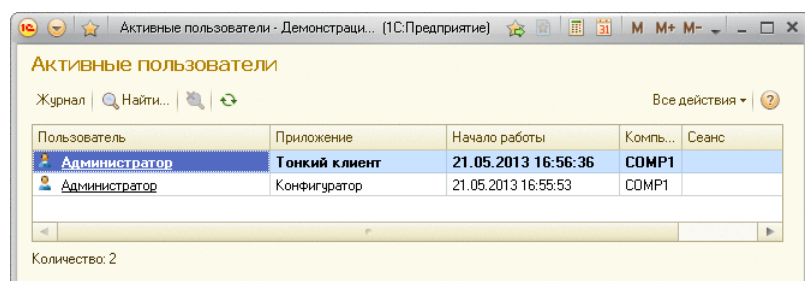


Рис. 53. Список активных пользователей

Данные пользователя, вызвавшего окно (текущего соединения), отображаются полужирным начертанием.

В нижней части окна отображается общее количество пользователей, работающих с данной информационной базой.

[Открыть журнал регистрации событий](#) – открывает журнал регистрации.

[Работа пользователя](#) – открывает журнал регистрации с установленным отбором по выбранному пользователю. Это действие можно выполнить также с помощью нажатия на гиперссылку с именем пользователя (колонка [Пользователь](#)).

6.15.2. Журнал регистрации

Для выполнения административных обязанностей часто требуется выяснить, какие события происходили в определенный момент времени или какие действия выполнял тот или иной пользователь.

Для этих целей предназначен журнал регистрации. В нем могут фиксироваться различные события. С его помощью администратор может получить историю работы пользователей с системой.

При работе пользователей система «1С:Предприятие» фиксирует в журнале основные действия, выполняемые пользователем по модификации данных информационной базы, выполнению регламентных операций, подключению и отключению от системы и т. д.

6.15.2.1. Просмотр журнала регистрации

Просмотр журнала регистрации выполняется в соответствующей форме.

Дата, время	Пользователь	Компьютер	Приложение	Сеанс	Событие	Комментарий	Статус транзакции	Метаданные
21.05.2013 16:58:09	СОМР1	Фоновое задание		11	Сеанс. Завершение			Представление данных
21.05.2013 16:58:21	Администратор	СОМР1	Тонкий клиент	8	Данные. Изменение		Зафиксирована 21.05.2013 16:58:21 (10943)	Документ. Приход товара Поступление товара 000000024 от 01.0...
21.05.2013 16:58:21	Администратор	СОМР1	Тонкий клиент	8	Данные. Проведение		Зафиксирована 21.05.2013 16:58:21 (10943)	Документ. Приход товара Поступление товара 000000024 от 01.0...
21.05.2013 16:58:23	Администратор	СОМР1	Тонкий клиент	8	Данные. Изменение		Зафиксирована 21.05.2013 16:58:23 (11697)	Документ. Приход товара Поступление товара 000000033 от 12.0...
21.05.2013 16:58:23	Администратор	СОМР1	Тонкий клиент	8	Данные. Проведение		Зафиксирована 21.05.2013 16:58:23 (11697)	Документ. Приход товара Поступление товара 000000033 от 12.0...

Рис. 54. Журнал регистрации

Каждое событие фиксируется в отдельной строке журнала. В левой колонке **Дата, время** пиктограммой отображается вид события (см. [рис. 55](#)). Для просмотра события следует выбрать пункт **Еще – Просмотр текущего события в отдельном окне** (**Все действия – Просмотр текущего события в отдельном окне**).

При работе с системой могут возникать события следующих видов:

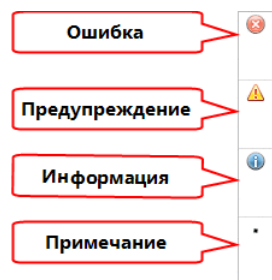


Рис. 55. Виды событий журнала регистрации

Если событие связано с данными, то становится доступным пункт **Еще – Открыть данные для просмотра** (**Все действия – Просмотр текущего события в отдельном окне**). С его помощью можно просмотреть данные, с которыми связано событие.

Событие может быть либо **транзакционным** либо **независимым** (определяется программно). По умолчанию установлен независимый режим записи событий.

Следует учитывать, что есть набор предопределенных событий, которые формируются на уровне системы. Для таких событий транзакционность устанавливается также на уровне системы. Так, события изменения данных, проведения документов являются транзакционными, а начало и завершение сеанса – независимыми. Ниже приведен полный список предопределенных событий.

- Независимые:
 - Сеанс:
 - Начало,
 - Завершение,
 - Аутентификация,
 - Ошибка аутентификации.
 - Провайдер Open-ID:
 - Подтверждено,
 - Отклонено.
 - Информационная база:
 - Изменение конфигурации,
 - Изменение конфигурации базы данных,
 - Изменение главного узла,
 - Изменение параметров журнала регистрации,
 - Изменение параметров информационной базы,
 - Изменение региональных установок,
 - Удаление данных информационной базы,
 - Запуск фонового обновления конфигурации базы данных,

- Завершение фонового обновления конфигурации базы данных.
- Отмена фонового обновления конфигурации базы данных,
- Приостановка фонового обновления конфигурации базы данных,
- Продолжение фонового обновления конфигурации базы данных,
- Тестирование и исправление:
 - Предупреждение,
 - Ошибка,
 - Сообщение.
- Фоновое задание:
 - Запуск,
 - Успешное завершение,
 - Принудительное завершение,
 - Ошибка выполнения,
 - Отмена.
- Доступ:
 - Отказ в доступе,
 - Доступ.
- Пользователи:
 - Добавление,
 - Изменение,
 - Удаление.
- Ошибка выполнения.
- Транзакционные:
 - Данные:
 - Изменение максимального периода рассчитанных итогов,
 - Изменение минимального периода рассчитанных итогов,
 - Добавление,
 - Изменение,
 - Удаление,
 - Проведение,
 - Отмена проведения,
 - Изменение состава стандартного интерфейса OData.
 - Транзакция:
 - Начало,
 - Фиксация,
 - Отмена.

При начале транзакции в журнал регистрации записывается событие начала транзакции [Транзакция.Начало](#), которому присваивается идентификатор транзакции. По завершении транзакции в случае ее фиксации в журнал записывается событие [Транзакция.Фиксация](#), статус транзакции записи [Транзакция.Начало](#) обновляется на [Зафиксирована](#). В случае отмены транзакции в журнал записывается событие [Транзакция.Отмена](#), статус транзакции для записи [Транзакция.Начало](#) обновляется на [Отменена](#). В случае аварийного завершения выполнения статус транзакции остается [Не завершена](#).

ВНИМАНИЕ! При открытии журнала регистрации по умолчанию устанавливается отбор по событиям, исключающий события, связанные с транзакциями.

Записи, соответствующие отмененным транзакциям и транзакциям с неопределенным статусом, выводятся бледным шрифтом.

Кроме просмотра журнала регистрации текущей информационной базы, имеется возможность просмотреть фрагмент журнала регистрации, ранее сохраненного в формате [LGD](#) или [LGF](#). Для этого нужно воспользоваться командой [Еще – Просмотреть из файла](#) ([Все действия – Просмотреть из файла](#)).

6.15.2.2. Установка интервала

С помощью пункта меню [Еще – Установить интервал дат для просмотра](#) ([Все действия – Установить интервал дат для просмотра](#)) можно управлять интервалом показа событий журнала.

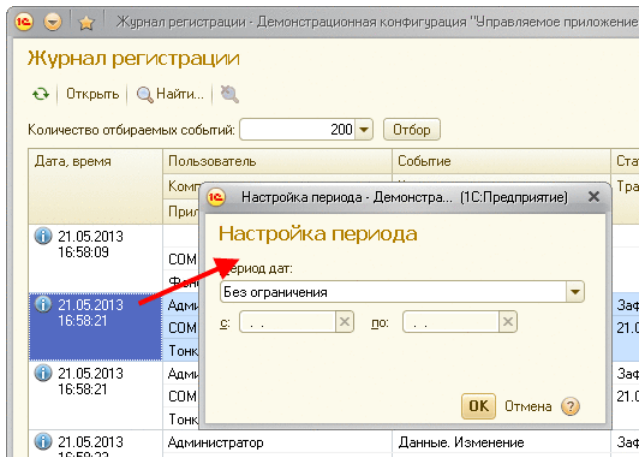


Рис. 56. Диалог настройки периода

В диалоге настройки нужно выбрать требуемый интервал и нажать кнопку **OK**.

Этот же диалог можно вызвать двойным щелчком по содержимому колонки **Дата, время**.

6.15.2.3. Установка отбора

С помощью кнопки **Отбор** или пункта меню **Еще – Отбор (Все действия – Отбор)** можно управлять отбором событий журнала. На экран выводится окно настройки отбора.

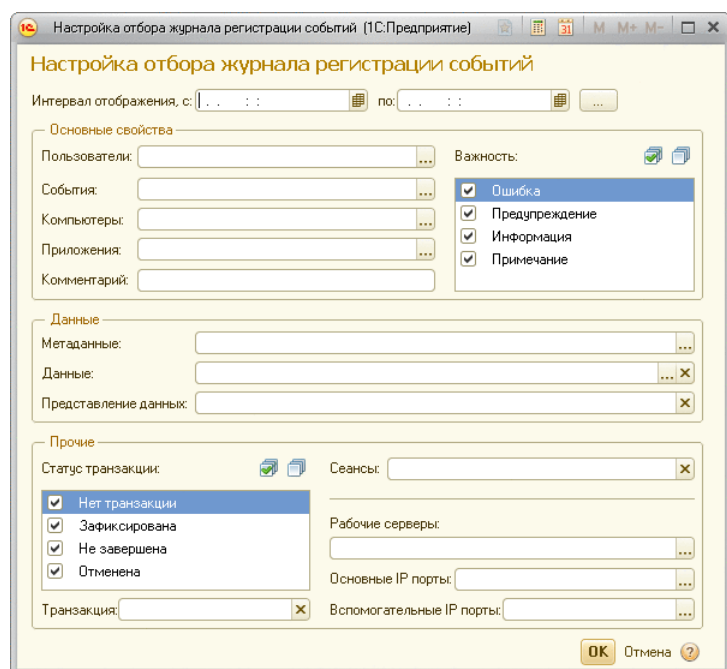


Рис. 57. Диалог настройки отбора в журнале регистрации

В диалоге производится установка фильтров отбора по периоду, пользователю, событию, наименованию компьютера, номеру соединения, степени важности событий, комментарию. При установке отбора по периоду следует помнить о следующих особенностях:

- отбор устанавливается с учетом времени;
- при ручном редактировании начальной или конечной даты следует также указывать время;
- при выборе начальной или конечной даты выбором из календаря, время устанавливается автоматически: при выборе в поле **с:** время устанавливается в 0:00:00, при выборе в поле **по:** время устанавливается в 23:59:59;
- при выборе интервала, с помощью кнопки выбора **...**, время устанавливается в начало дня начала периода и в конец дня окончания периода.

Если исполнялись несколько видов приложений, то в списке приложений можно указать, события каких именно приложений следует отбирать.

В списке событий указывается, какие виды событий нужно включить в отбор.

В группе **Данные** указываются данные, по которым будет производиться отбор событий. Информация о событиях представлена в колонках **Метаданные**, **Данные** и **Представление данных журнала регистрации**.

В поле **Метаданные** содержится список метаданных, представленных в конфигурации. Нужно установить флажки для тех метаданных, по которым требуется произвести отбор.

В поле **Данные** выбирается объект информационной базы, по которому требуется отобрать события.

В поле **Представление данных** указывается строковое представление.

В группе **Прочие** указываются дополнительные параметры отбора:

- **Статус транзакции** – выбираются статусы транзакции.
- **Транзакция** – указывается конкретная транзакция.
- **Сеансы** – указываются номера сеансов (через запятую).
- **Рабочие серверы** – выбираются центральные серверы кластеров (для клиент-серверного варианта работы).

- **Основные IP порты** – выбираются IP-порты менеджеров кластера (для клиент-серверного варианта работы).
- **Вспомогательные IP порты** – выбираются вспомогательные порты менеджеров кластера (для клиент-серверного варианта работы).

Для установки отбора нужно нажать кнопку **OK**.

Представление установленного отбора отображается правее кнопки **Отбор**. Представление отбора предваряется гиперссылкой **Отключить**. Нажатие на эту гиперссылку отключает отбор.

6.15.3. Удаление помеченных объектов

Процесс удаления помеченных объектов имеет несколько стадий. Стадии следуют строго последовательно. Перед каждой очередной стадией можно прервать процесс, закрыв окно режима. Ниже подробно описаны действия системы и пользователя на каждой стадии.

6.15.3.1. Выбор варианта удаления

На первой стадии система предложит выбрать вариант удаления: полное или выборочное удаление.

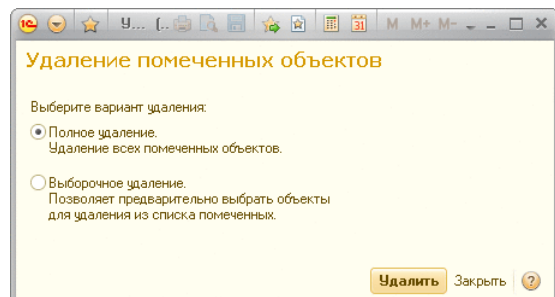


Рис. 58. Удаление помеченных объектов

6.15.3.2. Полное удаление

При выборе варианта **Полное удаление** система выполняет попытку удаления всех помеченных объектов. Удаление происходит с контролем ссылочной целостности, по окончании операции не все объекты могут оказаться удаленными, т. к. на часть из них могут оказаться ссылки в не удаляемых объектах.

Перечень не удаленных объектов отображается после окончания процесса удаления (если остались не удаленные объекты). Подробнее см. [здесь](#).

6.15.3.3. Выборочное удаление

При выборе варианта **Выборочное удаление** система выполняет формирование списка объектов, помеченных на удаление. В конце этой стадии пользователю выдается окно. Оно содержит список объектов, помеченных на удаление, которые обнаружены в информационной базе.

В этом списке можно выбрать те объекты, которые должны быть удалены.

Если напротив объекта в списке объектов установлен флажок, то это означает, что объект будет удален.

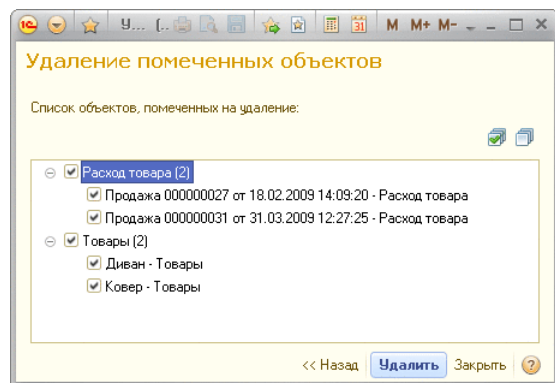


Рис. 59. Список объектов, помеченных на удаление

Установка пометки в данном диалоге имеет смысл только в пределах режима удаления помеченных объектов и не влияет на пометки объектов в самой системе. Если данная пометка была снята в диалоге, то после выхода из режима удаления помеченных объектов объект все равно останется помеченным на удаление.

Двойным щелчком мыши на объекте можно открыть форму этого объекта. Это позволяет просмотреть объекты и принять решение о правомерности их удаления.

На этой стадии пользователь может, не закрывая окно режима удаления помеченных объектов, переходить в другие окна и режимы, а также вносить любые исправления.

Для удаления объектов следует нажать кнопку **Удалить**. В этом случае система переходит к удалению объектов, разрешенных для удаления. Удаление происходит с контролем ссылочной целостности, по окончании операции не все объекты могут оказаться удаленными, т. к. на часть из них могут оказаться ссылки в не удаляемых объектах.

6.15.3.4. Список не удаленных объектов

Если в информационной базе есть ссылки на выбранные объекты в списке помеченных на удаление, то система выведет предупреждение: **Невозможно удалить объектов: <количество>, т. к. в информационной базе на них ссылаются другие объекты**. Такие объекты удалены не будут.

По нажатию кнопки **Далее** система отобразит список не удаленных объектов, который содержит список обнаруженных ссылок. Ссылки выводятся для выбранного объекта.

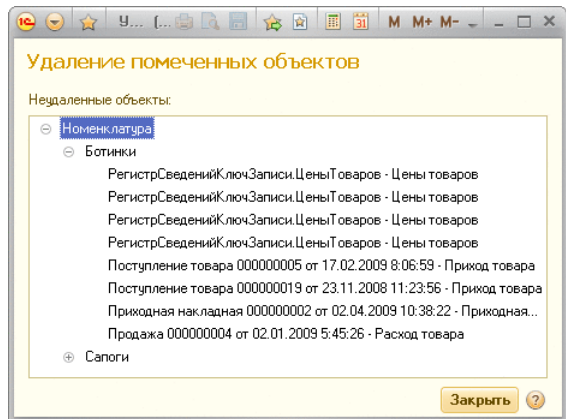


Рис. 60. Список не удаленных объектов

При выборе в списке нужной ссылки ее можно открыть для просмотра и редактирования. Это позволяет внести изменения в объект (выбрать другую ссылку), чтобы помеченный объект можно было удалить.

Чтобы выйти из режима удаления помеченных объектов, следует нажать кнопку **Закреть**.

6.15.4. Поиск ссылок на объекты

Данный режим предоставляет администратору системы возможность найти объекты, ссылающиеся на выбранный объект.

В данном режиме пользователь может выбрать объект и получить список ссылок на него, содержащихся в других объектах информационной базы.

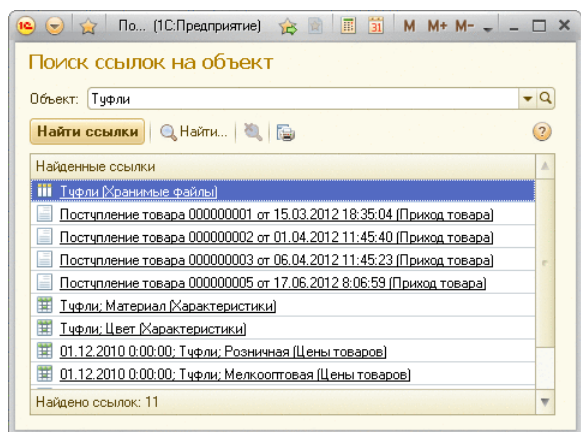


Рис. 61. Поиск ссылок на объект

Необходимо выбрать объект в поле **Объект** и нажать кнопку **Найти ссылки**. Система производит поиск ссылок на указанный объект во всех объектах информационной базы, в которых он может встречаться (определяется прикладным решением). После выполнения поиска можно проанализировать найденные ссылки. Чтобы открыть форму интересующей ссылки, следует нажать кнопку **Открыть** (если это допустимо) или нажать на гиперссылку. В том случае, если необходимо выполнить поиск ссылок на один из найденных элементов списка **Найденные ссылки**, можно открыть контекстное меню (на выбранной строке) и выполнить команду **Найти ссылки**. При этом будет открыто новое окно поиска ссылок на объекты и выполнен поиск ссылок на выбранный объект.

При работе с окном можно переходить в другие окна и режимы, не закрывая окно поиска.

6.15.5. Проведение документов

С помощью данного сервиса имеется возможность выполнять пакетное проведение или перепроведение документов, а также восстановление последовательностей.

6.15.5.1. Проведение документов

С помощью сервиса **Проведение документов** осуществляется проведение документов выбранных видов в указанном интервале.

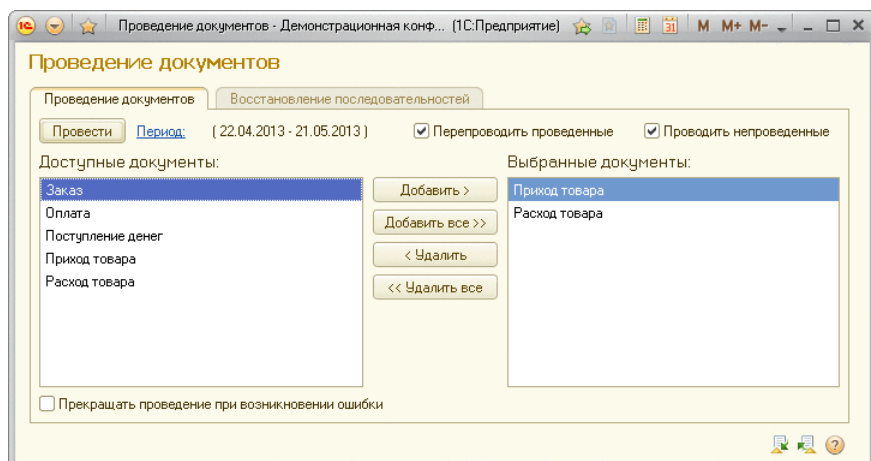


Рис. 62. Проведение документов

В верхней части диалога, в поле **Период** задается интервал, в котором будут проводиться документы. Для установки периода следует выбрать вариант стандартного периода или выбрать **Произвольный период** и установить период вручную. Если в окне задания произвольного периода очистить обе границы интервала, то проведение будет выполняться без ограничения периода, о чем будет свидетельствовать соответствующая надпись справа от поля выбора периода.

Окно диалога содержит список видов документов, которые могут проводиться. В список доступных документов попадают только те виды документов, для которых у текущего пользователя имеется право **Интерактивное проведение**.

Список выбранных документов, которые следует проводить, редактируется двойным щелчком мыши или с помощью кнопок **Добавить >** (доступно множественное выделение) и **Добавить все >>** и обратно, **< Исключить** (доступно множественное выделение) и **<< Исключить все**.

Над списком типов документов расположено поле **Режим проведения**, в котором следует выбрать, какие документы будут проводиться: проведенные, не проведенные или и те и другие.

После установки всех необходимых параметров для выполнения проведения следует нажать кнопку **Провести**. Перед проведением определяется дата первого и последнего проводимого документа (исходя из режима проведения и списка проводимых документов).

При групповом проведении документы, помеченные на удаление, не проводятся, даже если они подходят по условиям, выбранным в системном диалоге группового проведения. Если в процессе проведения документа возникла ошибка, то поведение системы зависит от значения флажка **Прекращать проведение при возникновении ошибки**. Если флажок установлен, то проведение будет завершено. Если флажок снят (значение по умолчанию), то проведение будет продолжено, а документы, проведение которых было выполнено с ошибками, будут сохранены.

После завершения процесса проведения будет выведена информация о количестве проведенных документов. Если в процессе проведения были обнаружены ошибки, то будет открыта форма, содержащая список документов с ошибками.

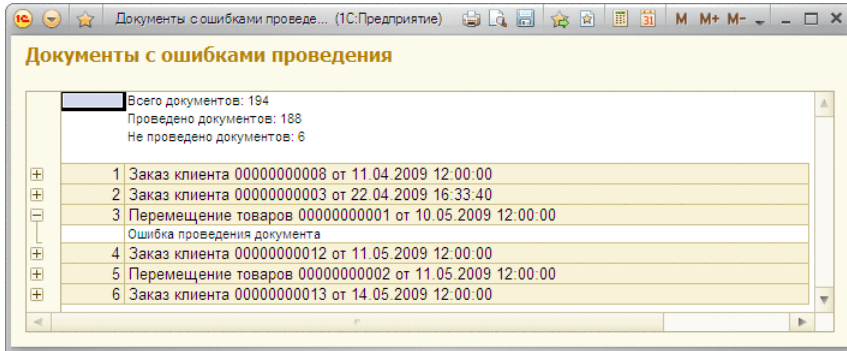


Рис. 63. Документы с ошибками проведения

Если в списке ошибок присутствует только одна строка **Ошибка проведения документа**, то это означает, что во время проведения документа произошла ошибка, но при этом документ не сформировал собственных сообщений об ошибке.

Двойной щелчок по строке с именем документа приведет к его открытию для просмотра.

Во время процесса проведения в панель состояния выводится информация о реальном интервале проведения документов, текущей дате проведения и общем количестве проведенных документов.

Прервать процесс проведения документов можно с помощью комбинации **Ctrl + Break**.

6.15.5.2. Восстановление последовательностей

Все документы в системе «1С:Предприятие» образуют единую хронологическую последовательность. Для этого каждый документ имеет дату и время. Даже если два документа имеют одинаковую дату и одинаковое время, они все равно располагаются в определенной последовательности, определяемой порядком их ввода в систему. Дата и время документа могут быть изменены. Таким образом, независимо от порядка ввода документы могут быть расположены в последовательности, которая отражает реальный порядок событий, происходивших в хозяйственной жизни предприятия.

В системе «1С:Предприятие» документ выполняет в процессе проведения некоторые действия, которые отражают данный документ в различных механизмах учета, поддерживаемых «1С:Предприятием».

Алгоритм проведения документа, как правило, отражает в учете данные, записанные в самом документе (в его реквизитах). Однако в некоторых случаях алгоритм проведения документа анализирует также и текущие итоги, используя их при проведении. Например, если документ списывает товары или материалы по средней себестоимости, то для определения суммы списания алгоритм проведения будет анализировать остатки товаров (материалов) на момент документа. Если списание выполняется по методу LIFO или FIFO, то алгоритм проведения будет анализировать существующие остатки товаров (материалов) в разрезе партий на момент (позицию) документа.

Очевидно, что документы, использующие при проведении данные итогов (например, в разрезе партий), должны проводиться строго последовательно. Однако на практике из-за ошибок при вводе информации и несвоевременного поступления документов часто приходится вводить или исправлять документы задним числом. Разумеется, в этом случае движения регистров, сформированные всеми последующими документами (расположенными после того, который был исправлен), становятся некорректными. Например, если выяснилось, что в одной из приходных накладных, введенных в начале месяца, было неверно указано количество товара, то во всех последующих расходных накладных, списывающих имеющиеся в наличии партии, необходимо заново проанализировать остатки с учетом внесенных изменений и заново записать движения регистров. То есть все документы, анализирующие остатки, расположенные после измененного документа, должны быть проведены заново.

Для автоматического контроля необходимости перепроведения документов используются последовательности документов. Каждая введенная в конфигурации последовательность документов обеспечивает контроль порядка проведения документов указанных видов. Таким образом, в системе может существовать несколько независимых последовательностей.

Режим восстановления последовательностей позволяет автоматически выполнить перепроведение всех документов, относящихся к последовательности, от текущей позиции границы последовательности до указанного момента. Текущая позиция границы последовательности определяется по дате, начиная с которой последовательность проведения документов необходимо восстановить.

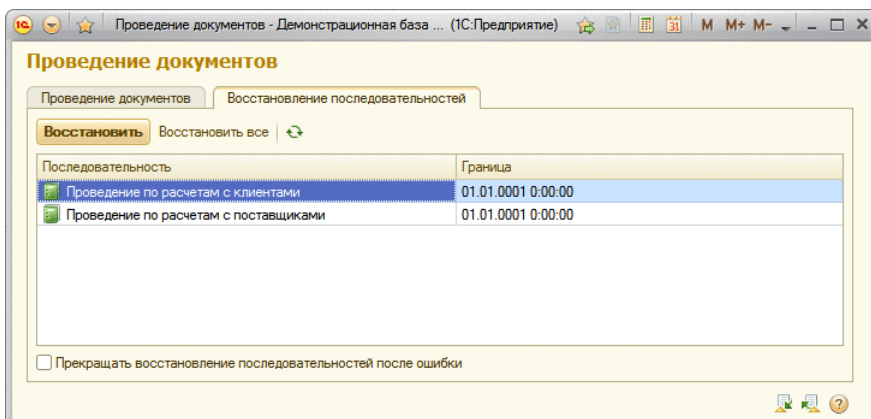


Рис. 64. Восстановление последовательности документов

В таблице отображается список существующих в конфигурации последовательностей, на которые у текущего пользователя имеется право **Изменение**. В колонке **Граница**

списка для каждой последовательности выводится текущая позиция границы последовательности. Для восстановления всех последовательностей можно нажать кнопку [Восстановить все](#).

Для выполнения восстановления последовательностей следует нажать кнопку [Восстановить](#). При этом система будет перепроводить все документы, относящиеся к выбранным последовательностям, начиная с позиции наиболее ранней границы из выбранных последовательностей и до указанной позиции включительно. Если выбраны несколько последовательностей (с помощью множественного выделения), то будут восстанавливаться выбранные последовательности в последовательности их отображения в списке. Если выбрана единственная последовательность, будет восстановлена именно она.

Флажок [Прекращать восстановление последовательностей после ошибки](#) определяет поведение системы в том случае, если во время восстановления последовательности будет обнаружена ошибка. Если флажок снят (значение по умолчанию), то ошибка не приведет к остановке всего процесса, т. е. будет продолжено восстановление других выбранных последовательностей. В противном случае процесс будет остановлен при обнаружении любой ошибки.

Прервать процесс восстановления последовательностей можно с помощью комбинации [Ctrl + Break](#).

6.15.6. Управление итогами

Данный сервис предоставляет возможность выполнять необходимые регламентные действия с регистрами, имеющимися в прикладном решении. К таким действиям относятся включение и выключение использования итогов, пересчет итогов, работа с агрегатами и т. д.

Вся работа с итогами делится на два набора возможностей:

- **Часто используемые возможности** (открывается по умолчанию) – этот режим предоставляет простые средства для выполнения наиболее часто используемых действий с итогами регистров.
- **Полные возможности** – предоставляет полный доступ к возможностям управления итогами и агрегатами прикладного решения.

В список попадают только те регистры накопления и бухгалтерии, на которые у текущего пользователя имеется право [Управление итогами](#), и для которых в текущем сеансе используются все разделители, в состав которых они входят (если разделители существуют в прикладном решении). С этим списком оперируют оба режима работы с итогами.

Для переключения режима использования служит гиперссылка в правой нижней части окна. При закрытии окно запоминает, в каком режиме оно находилось в момент закрытия, и при повторном открытии откроется в запомненном режиме.

Далее подробнее опишем оба режима.

6.15.6.1. Часто используемые возможности

К списку часто используемых возможностей относятся операции по установке периода рассчитанных итогов, включению использования итогов, перестройке и заполнению агрегатов, а также операция получения оптимальных агрегатов.

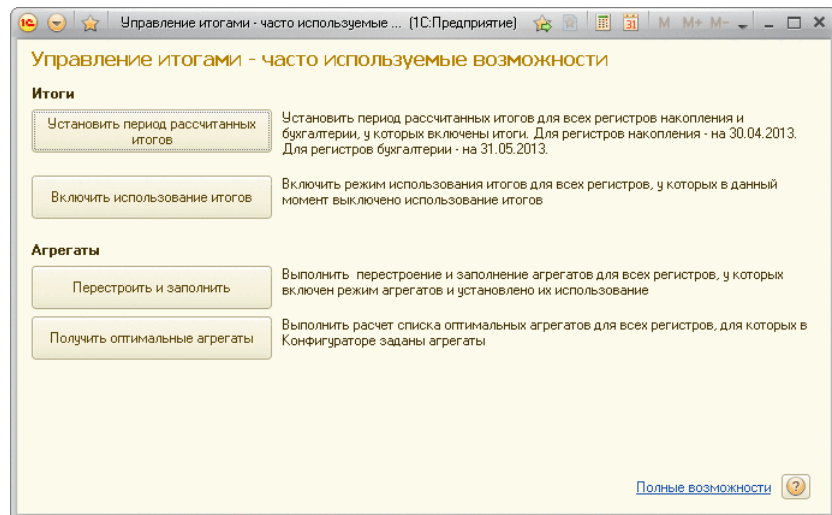


Рис. 65. Управление итогами – часто используемые возможности

6.15.6.1.1. Установить период рассчитанных итогов

Данная операция позволяет установить период рассчитанных итогов для всех регистров накопления и бухгалтерии, у которых включены итоги. Для регистров накопления период будет установлен на **дату окончания предыдущего месяца**, т. к. наиболее типовым использованием регистра накопления является получение текущих остатков. Для регистра бухгалтерии период будет установлен на **дату окончания текущего месяца**, т. к. наиболее типовым использованием регистра бухгалтерии является получение оборотов за текущий месяц.

СОВЕТ. Операцию можно использовать в начале каждого месяца для повышения производительности регистров.

6.15.6.1.2. Включить использование итогов

Данная операция позволяет включить использование для всех регистров, у которых выключено использование итогов, кроме оборотных регистров накопления, находящихся в режиме агрегатов.

СОВЕТ. Операция может понадобиться, например, в том случае, если произойдет аварийное завершение операции массового изменения данных регистров, которая отключает использование итогов для ускорения работы.

6.15.6.1.3. Перестроить и заполнить

Данная операция выполняет операции перестроения и заполнения для всех оборотных регистров накопления, у которых включен режим агрегатов и установлено их использование.

Подробнее о работе с агрегатами можно прочитать в книге «1С:Предприятие 8.3. Руководство разработчика».

СОВЕТ. Операцию можно использовать в качестве регламентной операции при использовании агрегатов.

6.15.6.1.4. Получить оптимальные агрегаты

Выполняет расчет оптимальных агрегатов для всех оборотных регистров накопления, у которых в конфигураторе заданы агрегаты.

СОВЕТ. Операцию можно использовать как перед включением использования агрегатов, так и в процессе эксплуатации системы.

6.15.6.2. Полные возможности

Режим полных возможностей позволяет получить полный доступ ко всем инструментам работы с итогами (закладка [Итоги](#)) и агрегатами (закладка [Агрегаты](#)) регистров

6.15.6.2.1. Работа с итогами

На закладке **Итоги** представлен список доступных данному пользователю регистров накопления, бухгалтерии и сведений (для которых включено использование итогов).

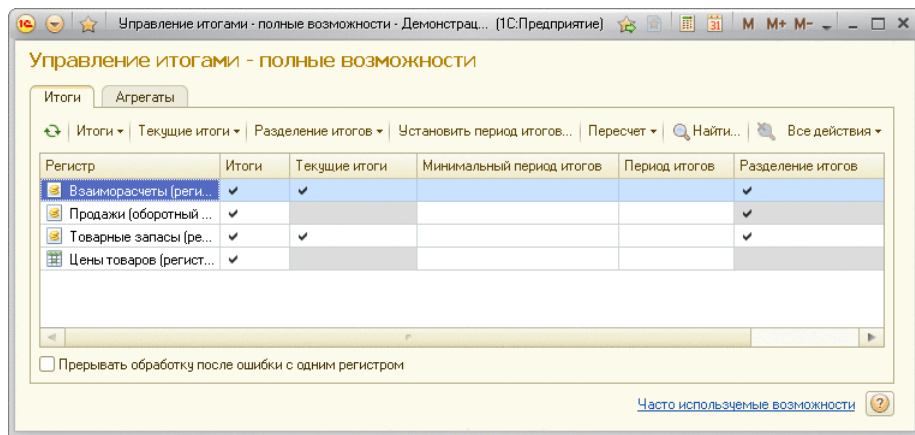


Рис. 66. Полные возможности управления итогами

Список показывает текущее состояние регистров системы. Галочками отмечены те режимы, которые в данный момент включены для каждого регистра:

- **Итоги** – состояние использования итогов;
- **Текущие итоги** – состояние использования текущих итогов;
- **Период итогов** – текущая дата актуальности итогов;
- **Разделение итогов** – состояние режима разделения итогов;
- **Агрегаты/итоги** – текущий режим использования агрегатов или итогов для оборотных регистров накопления, для которых в конфигураторе заданы агрегаты.

Серым цветом отмечены те режимы, которые невозможно изменить при текущем состоянии системы. Так, например, серый цвет в колонке **Разделение итогов** означает, что для выбранного регистра в конфигураторе запрещено разделение итогов.

Используя необходимые команды, можно включать или выключать соответствующий режим или выполнять расчет тех или иных итогов.

При выполнении всех команд доступен режим множественного выделения. Т. е. исполняемая команда будет выполнена для всех выделенных регистров. Если во время выполнения команды будет обнаружена ошибка, то дальнейшее поведение системы зависит от состояния флажка **Прерывать обработку после ошибки с одним регистром**. Если флажок снят (значение по умолчанию), то исполнение команды не будет прервано (в случае обнаружения ошибки) и будут обработаны все выделенные регистры, в противном случае обработка будет прервана.

Если у регистра есть возможность работать в режиме агрегатов, то двойной щелчок по содержимому колонки **Агрегаты/Итоги** приведет к переходу на закладку **Агрегаты** и установке курсора на регистр с тем же именем, что и на закладке **Итоги**.

6.15.6.2.2. Работа с агрегатами

Инструменты, собранные на закладке **Агрегаты**, предназначены для управления агрегатами оборотных регистров накопления (подробнее об агрегатах можно прочитать в книге «1С:Предприятие 8.3. Руководство разработчика»).

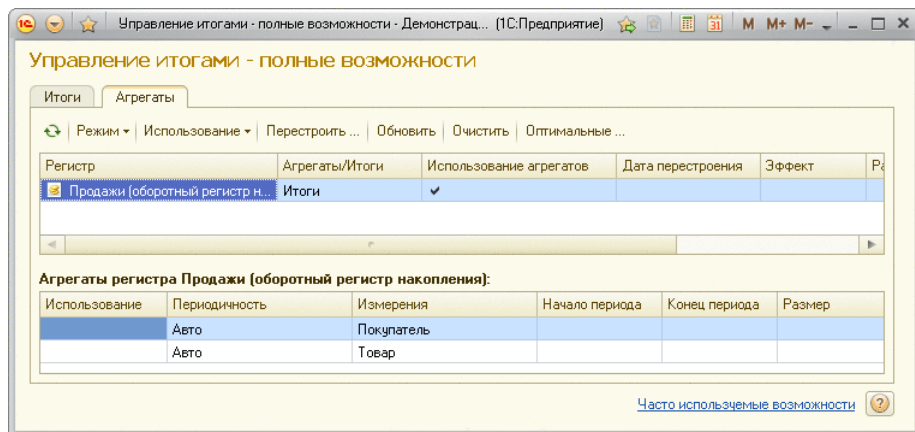


Рис. 67. Полные возможности управления агрегатами

Верхний список содержит перечень оборотных регистров накопления текущей конфигурации, для которых в конфигураторе заданы агрегаты. Нижний список (**Агрегаты регистра...**) содержит перечень агрегатов, заданных для регистра, признак использования того или иного агрегата и статистическую информацию об агрегате.

Имеется возможность переключать режим использования регистра, изменять признак использования агрегатов и выполнять основные операции с агрегатами.

При расчете оптимальных агрегатов будет запрошен каталог, в который будет помещен файл со списком оптимальных агрегатов для выбранного регистра. Регистр будет отмечен полужирным шрифтом, если рекомендуется заменить существующие в системе агрегаты на рассчитанный список оптимальных агрегатов.

При сохранении оптимальных агрегатов имя файла будет сформировано следующим образом: **ИмяАгрегата.xml**. Так, для регистра **Продажи** на [рис. 67](#) имя файла оптимальных агрегатов будет иметь вид **Продажи.xml**.

6.15.7. Управление полнотекстовым поиском

Система «1С:Предприятие» предоставляет возможность организации полнотекстового поиска по данным. Возможность поиска, формы для ввода условий поиска проектируются при создании конфигурации. Система предоставляет возможности для управления полнотекстовым поиском.

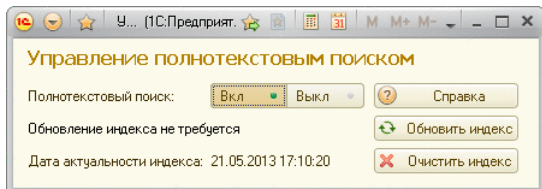


Рис. 68. Режим управления полнотекстовым поиском

Включение или выключение полнотекстового поиска выполняется с помощью тумблера **Полнотекстовый поиск**. Для выполнения этой операции требуется монополярный доступ к информационной базе. Это означает, что включить (выключить) полнотекстовый поиск можно только тогда, когда с информационной базой работает только один пользователь.

Индекс поиска формируется системой после нажатия кнопки **Обновить индекс**. Для оптимизации процесса формирования индекса используется основной индекс и дополнительный. Дополнительный индекс формируется при вводе данных и содержит информацию по данным, введенным после последнего обновления основного индекса.

Очистка индексов (запускается нажатием кнопки **Очистить индекс**) нужна для того, чтобы удалить индекс, например, чтобы освободить дисковое пространство, занятое файлами с индексом. После очистки индекса нужно выполнить индексирование (если требуется).

Кнопки в диалоге доступны только в том случае, если пользователь имеет право **Административные функции**.

В поле **Дата актуальности индекса** указывается дата начала последнего выполнения индексирования.

6.15.8. Управление расширениями конфигурации

Диалог предназначен для управления расширениями конфигурации в режиме «1С:Предприятия».

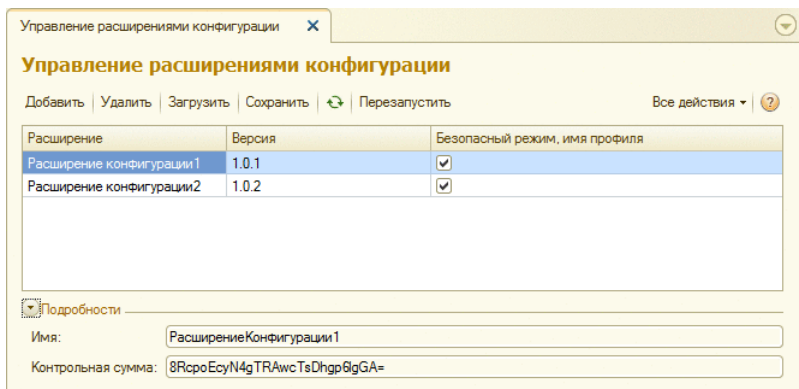


Рис. 69. Диалог управления расширениями конфигурации

Диалог управления расширениями доступен пользователю, который обладает правом **АдминистрированиеРасширенийКонфигурации**. Пользователю может потребоваться право **Администрирование** для указания профиля безопасности для подключаемого расширения.

В диалоге можно выполнить стандартные действия по работе с расширениями:

- Добавить новое расширение конфигурации из файла (кнопка **Добавить**);
- Удалить подключенное расширение (кнопка **Удалить**);
- Заменить существующую версию расширения новой версией (кнопка **Загрузить**);
- Сохранить расширение конфигурации в файл (кнопка **Сохранить**);
- Обновить список расширений.
- Перезапустить клиентское приложение, чтобы изменения в расширениях вступили в силу (кнопка **Перезапустить**).

В том случае, если необходимо настраивать профили безопасности (на сервер «1С:Предприятия»), то следует воспользоваться группой **Подробности**, которая содержит значение контрольной суммы (поле **Контрольная сумма**), необходимое для заполнения одноименного свойства в описании доступного внешнего модуля.

Глава 7. Настройка веб-серверов для работы с «1С:Предприятием»

Данная глава описывает механизм настройки веб-серверов для работы с веб-клиентом и Web-сервисами, а также настройка поддержки OpenID-аутентификации. После публикации обращение к опубликованным компонентам будет выполняться следующим образом:

- **Обращение к веб-клиенту.** Для запуска веб-клиента необходимо использовать адрес, который формируется по следующим правилам: `<Имя хоста веб-сервера>/<Имя виртуального каталога>`. Если имя виртуального каталога `DemoCfg`, то для запуска веб-клиента следует набрать следующий URL (для получения доступа с локальной машины): <http://localhost/DemoCfg>.

- **Обращение в Web-сервису.** Для получения доступа к Web-сервису необходимо использовать адрес, который формируется следующим образом: `<Имя хоста веб-сервера>/<Имя виртуального каталога>/ws/<Имя Web-сервиса>` или `<Имя хоста веб-сервера>/<Имя виртуального каталога>/ws/<Адрес Web-сервиса>`.

Так, если виртуальный каталог имеет имя `DemoWS`, имя Web-сервиса в конфигураторе указано как `ДемонстрацияРаботыWS`, а в качестве адреса указано `DemoWorkWS`, то обращение к Web-сервису можно выполнять одновременно по двум адресам (для получения доступа с локальной машины): <http://localhost/DemoWS/ws/ДемонстрацияРаботыWS> или <http://localhost/DemoWS/ws/DemoWorkWS>.

- **Обращение в HTTP-сервису.** Для получения доступа к HTTP-сервису необходимо использовать адрес, который формируется следующим образом: `<Имя хоста веб-сервера>/<Имя виртуального каталога>/hs/<путь к ресурсу>` или `<Имя хоста веб-сервера>/<Имя виртуального каталога>/hs/<путь к ресурсу>`.

- **OpenID-аутентификация** выполняется системой автоматически.

Веб-серверы семейства Internet Information Services (далее IIS) поставляются вместе с операционной системой, и для упрощения понимания, какой веб-сервер вы используете, приведем таблицу соответствия версии веб-сервера и операционной системы:

Версия IIS	Версия операционной системы
IIS 5.1	Windows XP Professional
IIS 6.0	Windows Server 2003 или Windows XP Professional x64 Edition
IIS 7.0	Windows Vista или Windows Server 2008
IIS 7.5	Windows 7 или Windows Server 2008 R2
IIS 8.0	Windows 8 или Windows Server 2012
IIS 8.5	Windows Server 2012 R2

ПРИМЕЧАНИЕ. Дистрибутив веб-сервера Apache (как для ОС Windows, так и для ОС Linux) можно получить на веб-сайте проекта: <http://httpd.apache.org/download>.

7.1. Общие требования

На компьютере, где выполняется публикация, должен быть установлен и настроен поддерживаемый веб-сервер. Для установки веб-сервера Internet Information Services может потребоваться дистрибутив используемой операционной системы. При установке веб-сервера необходимо **обязательно** установить поддержку **ISAPI расширений**.

Для установки и настройки веб-серверов необходимы административные права на компьютере, где выполняется публикация:

- Для ОС Windows Vista и старше, чтобы выполнить публикацию, следует запустить конфигуратор с помощью пункта **Запуск от имени администратора** контекстного меню приложения или программы запуска. Если публикация выполняется с помощью утилиты `webinst`, то от имени администратора должна быть запущена или сама утилита, или командная строка Windows.
- Для ОС Linux, чтобы выполнить публикацию, следует получить права суперпользователя (пользователь `root`) с помощью команды `su` или запускать приложение, которое выполняет публикацию, с помощью команды `sudo`.

Публикация возможна только в том случае, если «1С:Предприятие» расположено на компьютере с веб-сервером.

Для веб-сервера IIS 7.x и старше не поддерживается публикация в том случае, если свойство **Каталог** (или параметр `dir` утилиты `webinst`) указывает на каталог `%SYSTEMDRIVE%\Inetpub\wwwroot`.

ПРИМЕЧАНИЕ. Для работы с конфигурацией через веб-сервер конфигурация не должна быть пустой.

Публикация может выполняться двумя способами:

- С помощью диалога публикации на веб-сервере, если на компьютере с веб-сервером имеется возможность запустить конфигуратор необходимой разрядности.
- С помощью утилиты `webinst`.

7.2. Виды публикации

7.2.1. Общая схема публикации

Общая схема публикации выглядит следующим образом:

- выполняется регистрация модуля обработки запросов (модуля расширения веб-сервера), соответствующего веб-серверу;
- на веб-сервере регистрируется виртуальное приложение;
- создается каталог виртуального приложения, и в нем размещается файл `default.vrd` (см. [здесь](#)) и выполняется его настройка;
- для пользователей назначаются права на каталог с файлом базы данных (только для файлового варианта);
- перезапуск веб-сервера потребуется в следующих случаях:
 - изменилась версия «1С:Предприятия» из которого выполняется публикация, по отношению к версии, из которой выполнялась предыдущая публикация;
 - изменился путь каталога установки «1С:Предприятия»;
 - для веб-сервера Apache создана новая публикация.

Для публикации веб-клиента следует использовать версию «1С:Предприятия», использующуюся для работы с информационной базой, к которой планируется получить доступ с помощью веб-клиента. Если на компьютере установлены две версии, например, `8.3.3.100` и `8.3.3.150` и запущен сервер «1С:Предприятия» версии `8.3.3.150`, то для публикации следует использовать конфигуратор или утилиту `webinst` ровно этой же версии.

При выполнении публикации следует помнить, что разрядность регистрируемого расширения веб-сервера должна совпадать с разрядностью самого веб-сервера. Чтобы определить, каким способом выполнять публикацию в том или ином случае, следует использовать следующую таблицу:

	32-разрядный веб-сервер	64-разрядный веб-сервер
32-разрядное «1С:Предприятие»	Конфигуратор или	Частично

	webinst	
64-разрядное «1С:Предприятие»	Не поддерживается	Конфигуратор или webinst

Конфигуратор – публикация с помощью конфигуратора возможна только в ОС Linux.

Частично – имеется возможность выполнить публикацию 32-разрядного приложения «1С:Предприятие» для использования с 64-разрядным веб-сервером IIS. Подробности настройки IIS см. [здесь](#). Вызывать утилиту [webinst](#) следует из каталога `bin` 32-разрядной версии «1С:Предприятия». Под ОС Linux такая публикация не поддерживается.

При публикации на веб-сервере IIS следует помнить, что:

- Публикация всегда выполняется для веб-сайта по умолчанию ([Default Web Site](#));
- Публикация всегда выполняется для пула приложений по умолчанию ([DefaultAppPool](#));
- Для пула приложений, используемого для работы «1С:Предприятия», должна быть отключена поддержка .NET. Для этого следует установить свойство пула приложений [Версии среды .NET Framework](#) в значение [Без управляемого кода](#).

Для выполнения публикации из конфигуратора, необходимо воспользоваться диалогом публикации ([Администрирование – Публикация на веб-сервере...](#)).

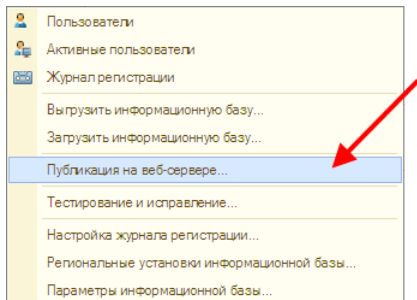


Рис. 70. Публикация на веб-сервере

Затем следует выполнить следующие действия:

- Ввести имя виртуального каталога в поле [Имя](#), при этом имя виртуального каталога может состоять только из символов латинского алфавита.
- В поле [Веб-сервер](#) указать тип веб-сервера, для которого выполняется публикация.
- В поле [Каталог](#) указать физическое местоположение каталога, в котором будут расположены файлы, описывающие виртуальный каталог. При использовании веб-сервера Apache имя каталога должно состоять только из символов латинского алфавита.
- В зависимости от необходимости установить флажки [Публиковать тонкий и веб-клиент](#) и [Публиковать Web-сервисы](#).
- Для веб-сервера IIS можно указать необходимость выполнять аутентификацию на веб-сервере средствами ОС.
- При необходимости выбирать Web-сервисы, которые необходимо опубликовать. Колонка [Адрес](#) может быть изменена. В данной колонке задается синоним Web-сервиса. Обращение к Web-сервису возможно как по имени, так и по синониму.
- При необходимости – выполнить настройки остальных параметров публикации.
- Нажатие кнопки [Опубликовать](#) запускает процесс публикации. Нажатие кнопки [Отключить](#) выполняет удаление публикации с выбранного веб-сервера.

После выполнения публикации будет предложено перезапустить веб-сервер в следующих случаях:

- изменилась версия «1С:Предприятия»;
- изменился путь к модулю расширения веб-сервера;
- выполнена новая публикация для веб-сервера Apache.

При использовании анонимной аутентификации и файловой информационной базы, при выполнении публикации происходит проверка наличия прав доступа на каталог информационной базы у пользователя, от лица которого выполняется анонимный доступ. Если пользователь не обладает необходимыми правами, выдается предупреждение о невозможности работы с этой информационной базой через веб-сервер. Рекомендуется или дать права на каталог с информационной базой, или установить флажок [Использовать аутентификацию операционной системы на веб-сервере](#).

Если публикация из конфигуратора недоступна (например, при использовании 64-разрядной ОС Windows), публикацию можно выполнить с помощью утилиты командной строки [webinst](#), которая доступна в ОС Windows и Linux в обеих разрядностях. Далее будет подробно описан диалог публикации и ключи командной строки утилиты [webinst](#).

7.2.2. Диалог публикации

Диалог публикации служит для создания публикации или подготовки файла-шаблона для публикации с помощью утилиты [webinst](#) (с использованием параметра командной строки `-descriptor`).

Все параметры, которые можно редактировать при создании публикации, размещены на двух закладках. Рассмотрим их подробнее.

7.2.2.1. Кнопки диалога

Кнопка [Опубликовать](#) выполняет публикацию на веб-сервере. При публикации происходит создание каталога на диске и настройка указанного веб-сервера для работы с «1С:Предприятием». Следует помнить, что публикация на веб-сервере IIS всегда выполняется для веб-сайта по умолчанию ([Default Web Site](#)) и для пула приложений по умолчанию ([DefaultAppPool](#)).

Под ОС Linux выполняются следующие действия:

- Для каталога, в котором расположен файл [default.vrd](#), в качестве группы-владельца устанавливается группа пользователя, от имени которого работает веб-сервер;
- Для файла [default.vrd](#) устанавливается доступ по чтению для группы, включающей пользователя, от имени которого работает веб-сервер.

В случае публикации файловой информационной базы, для каталога с файлом информационной базы в качестве группы-владельца устанавливается группа пользователя, от имени которого работает веб-сервер, а также настраивается наследование группы-владельца для обеспечения работы с информационной базой (подробнее см. [здесь](#)).

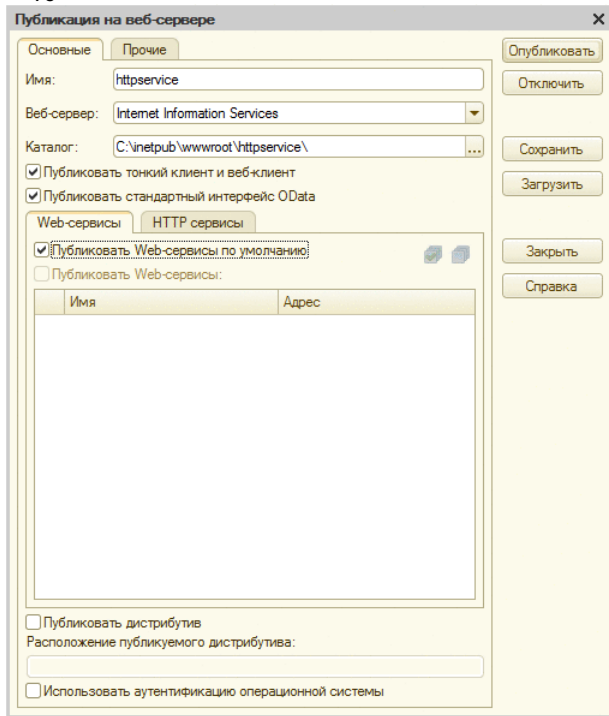


Рис. 71. Публикация на веб-сервере

Кнопка **Отключить** выполняет удаление приложения из веб-сервера и каталога публикации, если это необходимо.

Кнопка **Сохранить** выполняет сохранение параметров, указанных в диалоге публикации на веб-сервере, в файл. При сохранении система запрашивает имя и место расположения файла, в который будет выполнено сохранение. Сохранение будет выполнено в формате файла `default.vrd`. С помощью этой команды можно создавать файлы шаблонов, которые будут использоваться в качестве параметра `-descriptor` утилиты `webinst`. В качестве значений атрибутов `ib` и `base` элемента `point` будут записаны параметры той информационной базы, из которой выполняется сохранение файла.

Кнопка **Загрузить** позволяет загрузить для редактирования произвольный файл `default.vrd`. При загрузке игнорируются атрибуты `ib` и `base` элемента `point` загружаемого файла.

Кнопка **Закреть** выполняется закрытие диалога.

Кнопка **Справка** открывает окно со справочной информацией о диалоге публикации.

7.2.2.2. Основные

На данной закладке можно задать основные параметры публикации.

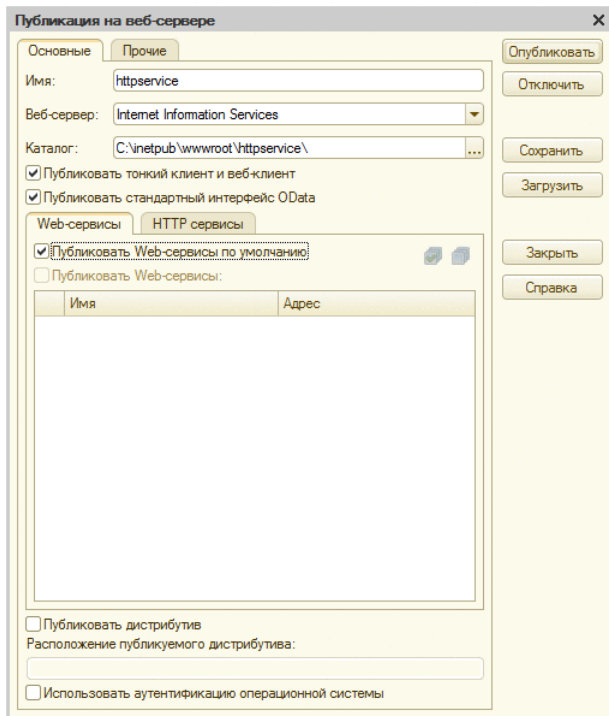


Рис. 72. Публикация на веб-сервере. Основные

Имя. Указывает имя публикации. При публикации с помощью утилиты `webinst` описывается параметром `-wsdir`. В файле `default.vrd` соответствует атрибуту `base` элемента `point`.

Веб-сервер. Указывает, для какого веб-сервера выполняется публикация. При публикации с помощью утилиты `webinst` описывается указанием какого-либо из параметров `iis`, `apache2` или `apache22`. При работе в ОС Linux возможна публикация только для веб-сервера Apache.

Каталог. Указывает физический каталог на диске, в котором будет находиться файл `default.vrd` и куда будет отображен виртуальный каталог веб-сервера. Каталог должен существовать. При публикации с помощью утилиты `webinst` описывается параметром `-dir`.

Публиковать тонкий и веб-клиент. Отвечает за возможность работы с опубликованной информационной базой тонким и веб-клиентами. Если флажок установлен, работа с

опубликованной информационной базой с помощью тонкого и веб-клиентов возможна. В файле `default.vrd` соответствует атрибуту `enable` элемента `point`.

Публиковать стандартный интерфейс OData. Отвечает за возможность получения доступа к стандартному интерфейсу OData прикладного решения. Подробнее механизм стандартного интерфейса OData описан в книге «1С:Предприятие 8.3. Руководство разработчика». В файле `default.vrd` соответствует атрибуту `enableStandardOData` элемента `point`.

Публиковать дистрибутив. Определяет возможность получения и установки клиентского приложения в том случае, если не совпадают версии клиентского приложения и сервера. В качестве дистрибутива используется zip-архив, полное имя которого указывается в качестве значения свойства **Расположение публикуемого дистрибутива**. В файле `default.vrd` эти свойства соответствуют атрибуту `pubdst` элемента `point`. В архиве должен располагаться дистрибутив клиентского приложения (без каталогов). При установке будут использоваться параметры установки, указанные в файле `1cestart.cfg` (аналогично обычной установке клиентского приложения).

Использовать аутентификацию операционной системы. Разрешает системе установить возможность аутентификации ОС на веб-сервере IIS.

7.2.2.2.1. Закладка Web-сервисы

Публиковать Web-сервисы. Установка данного флажка приведет к тому, что будут опубликованы Web-сервисы, созданные в конфигурации и перечисленные в таблице, расположенной ниже флажка. В файле `default.vrd` соответствует атрибуту `enable` элемента `ws`. Если флажок сброшен, то это эквивалентно отсутствию элементу `ws` в файле `default.vrd` или наличие элемента `ws` с атрибутом `enable`, установленным в значение `true`.

Публиковать Web-сервисы по умолчанию. Отвечает за возможность использования в данной информационной базе Web-сервисов, которые опубликованы без явного указания разрешения использования. В файле `default.vrd` соответствует атрибуту `pointEnableCommon` элемента `ws`.

Таблица ниже флажка **Публиковать Web-сервисы** содержит перечень публикуемых Web-сервисов и позволяет управлять публикацией каждого Web-сервиса. Первая колонка управляет публикацией конкретного Web-сервиса. Если флажок сброшен, то данный Web-сервис будет запрещен к использованию (его нельзя будет вызвать). В файле `default.vrd` соответствует атрибуту `enable` элемента `point`.

Вторая колонка (с именем **Имя**) содержит имя Web-сервиса, как оно задано при создании (описание Web-сервисов приведено в книге «1С:Предприятие 8.3. Руководство разработчика»). В файле `default.vrd` соответствует атрибуту `name` элемента `point`.

Последняя колонка таблицы (с именем **Адрес**) содержит псевдоним имени публикуемого Web-сервиса. Обращение к Web-сервису возможно как по имени, так и по псевдониму. Псевдоним Web-сервиса можно редактировать в окне публикации. В файле `default.vrd` соответствует атрибуту `alias` элемента `point`.

7.2.2.2.2. Закладка HTTP-сервисы

Закладка **HTTP сервисы** предназначена для управления возможностью доступа к прикладному решению с помощью HTTP-сервисов.

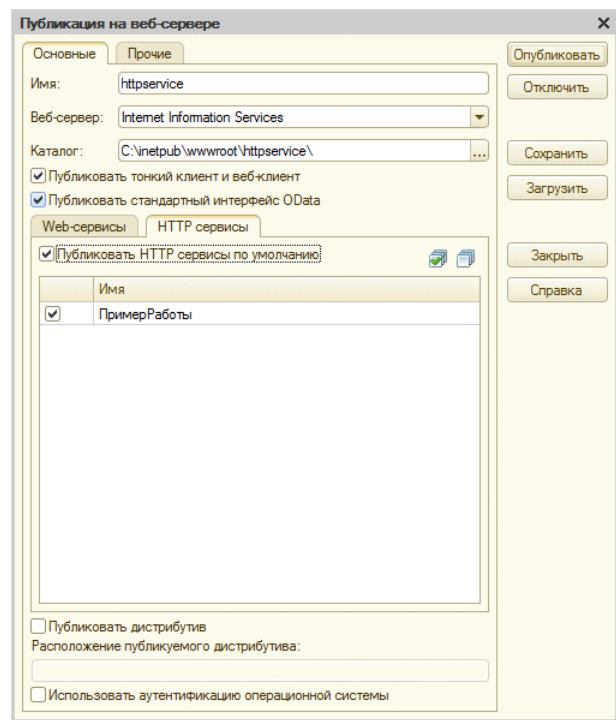


Рис. 73. Публикация HTTP-сервисов

Публиковать HTTP сервисы по умолчанию. Установка данного флажка приведет к тому, что будут опубликованы HTTP-сервисы, созданные в конфигурации и перечисленные в таблице, расположенной ниже флажка. В файле `default.vrd` соответствует атрибуту `publishByDefault` элемента `httpServices`. Если флажок сброшен, то это эквивалентно отсутствию элементу `httpServices` в файле `default.vrd` или наличие элемента `httpServices` с атрибутом `publishByDefault`, установленным в значение `false`.

Таблица ниже флажка **Публиковать HTTP сервисы по умолчанию** содержит перечень публикуемых HTTP-сервисов и позволяет управлять публикацией каждого HTTP-сервиса. Первая колонка управляет публикацией конкретного HTTP-сервиса. Если флажок сброшен, то данный HTTP-сервис будет запрещен к использованию (его нельзя будет вызвать). В файле `default.vrd` соответствует атрибуту `enable` элемента `service`.

Вторая колонка (с именем **Имя**) содержит имя HTTP-сервиса, как оно задано при создании (описание HTTP-сервисов приведено в книге «1С:Предприятие 8.3. Руководство разработчика»). В файле `default.vrd` соответствует атрибуту `name` элемента `service`.

7.2.2.3. Прочие

На данной закладке можно изменить вспомогательные параметры публикации.

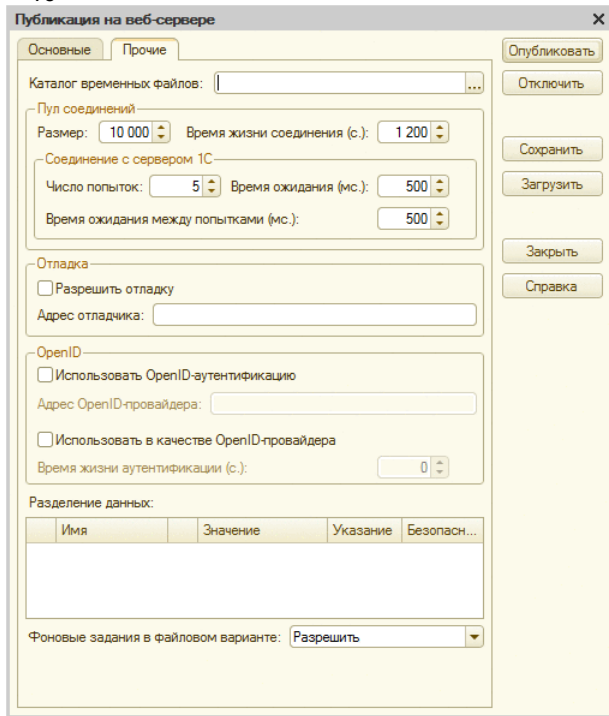


Рис. 74. Прочие параметры публикации на веб-сервере

Каталог временных файлов. Позволяет указать каталог временных файлов для работы расширения веб-сервера или файлового варианта информационной базы. В файле `default.vrd` соответствует атрибуту `temp` элемента `point`. Подробнее см. [здесь](#).

Группа Пул соединений. Описывает элемент `pool` файла `default.vrd`. Подробнее см. [здесь](#).

Группа Отладка. Описывает элемент `debug` файла `default.vrd`. Подробнее см. [здесь](#).

Группа OpenID. Описывает элемент `openid` файла `default.vrd`. Подробнее см. [здесь](#).

Разделение данных. Описывает элемент `zones` файла `default.vrd`. Подробнее см. [здесь](#). Подробнее остановимся на структуре таблицы с разделителями.

В таблицу попадают все независимые разделители, существующие в конфигурации или загруженном файле. Первая колонка (без имени) определяет необходимость создания элемента `zone` для выбранного разделителя. Следует помнить, что сопоставление элемента выполняется не по имени разделителя, а по его порядковому расположению в списке. Если отключается первый разделитель, то имеет смысл отключить и все остальные, т. к. параметры элемента `zones` будут применены системой к другим разделителям.

Колонка **Имя** содержит имя разделителя, как оно задано в свойствах общего реквизита. Флажок в следующей колонке определяет, будет задано значение разделителя в элементе `zone` или нет. Если флажок установлен, то в качестве значения атрибута `value` будет использовано значение из колонки **Значение**.

Флажки в колонках **Указание** и **Безопасное** отвечают за атрибуты `safe` и `specify` (соответственно) элемента `zone` файла `default.vrd`.

Параметр **Фоновые задания в файловом варианте** описывает возможность использования фоновых заданий в файловом варианте информационной базы (атрибут `allowexecutescheduledjobs` корневого элемента `point`). Подробнее см. [здесь](#).

7.2.3. Утилита webinst

Утилита предназначена для настройки веб-серверов с целью поддержки работы веб-клиента. Утилита работает как в среде ОС Windows, так и в среде ОС Linux и входит в состав дистрибутива системы.

[Копировать в буфер обмена](#)

```
webinst [-publish] | -delete <веб-сервер>
                    -wsdir <виртуальный каталог>
                    -dir <физический каталог>
                    -connstr <строка соединения>
                    -confpath <путь к файлу httpd.conf>
                    -descriptor <путь к файлу default.vrd>
                    [-osauth]
```

ВНИМАНИЕ! Название и значение параметра должны разделяться символом пробел. Если параметр содержит пробелы, он должен быть заключен в кавычки (""). Если внутри параметра есть символ кавычки, то внутренние кавычки должны быть удвоены.

ВНИМАНИЕ! При запуске утилиты допустимо указание только одного из параметров: `iis`, `apache2` или `apache22`.

ВНИМАНИЕ! Для выполнения публикации утилита должна запускаться от имени администратора. При работе в ОС Windows будет выполняться запрос на повышение привилегий.

`-publish` по умолчанию

Выполняется публикация веб-клиента на веб-сервер.

`-delete`

Выполняется удаление публикации из указанного каталога (включая удаление физического каталога с публикацией).

ПРИМЕЧАНИЕ. При удалении публикации достаточно указания только параметра `-wsdir`. Остальные параметры могут указываться для контроля работы.

`<веб-сервер>`

Указывает, для какого веб-сервера будет выполняться действие (публикация или удаление публикации):

- `-iis` – веб-сервер семейства Microsoft Internet Information Services версий 5.1, 6.0, 7.x, 8.x (только при использовании с ОС Windows);
- `-apache2` – веб-сервер Apache 2.0;
- `-apache22` – веб-сервер Apache 2.2.

`-wsdir`

Имя виртуального каталога.

`-dir`

Имя физического каталога, куда будет отображен виртуальный каталог веб-сервера. Каталог должен существовать.

Для веб-сервера IIS 7.x и старше не поддерживается публикация в том случае, если значение этого параметра указывает на каталог `%SYSTEMDRIVE%\inetpub\wwwroot`.

ПРИМЕЧАНИЕ. Имя каталога не должно завершаться символом «\», если оно заключено в кавычки. Правильно: `"c:\my path"`, неправильно: `"c:\my path\"`.

`-connstr`

Строка соединения с информационной базой. Подробности см. описание строки соединения встроенной справки конфигурирования.

`-confpath` только для Apache

Полный путь к конфигурационному файлу (`httpd.conf`) веб-сервера Apache. Данный параметр применяется только при использовании веб-серверов Apache.

`-descriptor`

Позволяет выполнить публикацию по шаблону, заданному существующим файлом, который задается в данном параметре (включая путь к файлу). Имя файла-шаблона не обязательно должно быть `default.vrd`. При выполнении публикации существующий файл `default.vrd` полностью замещается файлом-шаблоном. Если одновременно с данным параметром указаны параметры `-wsdir` или `-connstr`, то значения этих параметров замещают собой значения атрибутов `base` и `ib` (соответственно) элемента `point`.

В том случае, если параметр `-descriptor` указывается одновременно с параметром `-delete`, то из файла-шаблона используется имя виртуального каталога (атрибут `base` элемента `point`) и строка соединения информационной базы (атрибут `ib` элемента `point`). Публикация будет удалена только в случае совпадения обоих значений у удаляемой публикации и файла-шаблона.

`-osauth` только для IIS

При публикации настраивает использование аутентификации ОС на веб-сервере. Данный параметр применяется только при использовании веб-серверов IIS.

Пример команды публикации для IIS 7.0 и старше:

```
webinst -publish -iis -wsdir demo -dir "c:\inetpub\demo" -connstr "Srvr=server:1741;Ref=demo;"
```

[Копировать в буфер обмена](#)

В данном примере выполняется публикация веб-клиента со следующими параметрами:

- Виртуальный каталог: `demo` (параметр `-wsdir demo`);
- Физический каталог, в который отображается виртуальный каталог: `c:\inetpub\demo` (параметр `-dir "c:\inetpub\demo"`);
- Строка соединения с информационной базой: `Srvr=server:1741;Ref=demo;` (параметр `-connstr "Srvr=server:1741;Ref=demo;"`, клиент-серверный вариант информационной базы).

Пример команды публикации для Apache 2.2:

```
webinst -publish -apache22 -wsdir DemoWS -dir "c:\apache.www\demows" -connstr "File=""c:\my db\demows"";" -confpath "C:\Program Files\Apache Software Found"
```

[Копировать в буфер обмена](#)

В данном примере выполняется публикация веб-клиента со следующими параметрами:

- Виртуальный каталог: `DemoWS` (параметр `-wsdir demoWS`);
- Физический каталог, в который отображается виртуальный каталог: `c:\apache.www\demows` (параметр `-dir "c:\apache.www\demows"`);
- Строка соединения с информационной базой: `File=""c:\my db\demows"";` (параметр `-connstr "File=""c:\my db\demows"";"`, файловый вариант информационной базы);
- Конфигурационный файл веб-сервера Apache: `C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf` (параметр `-confpath "C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf"`).

Пример публикации с использованием файла-шаблона:

```
webinst -publish -iis -wsdir demoMA -dir "c:\inetpub\wwwroot\demoMA" -connstr "Srvr=server:1741;Ref=demo;" -descriptor template.vrd
```

[Копировать в буфер обмена](#)

В данном примере:

- Выполняется публикация на веб-сервере IIS (параметры `-publish -iis`);
- Виртуальный каталог: `demoMA` (параметр `-wsdir demoMA`);
- Физический каталог, в который отображается виртуальный каталог: `c:\inetpub\wwwroot\demoMA` (параметр `-dir "c:\inetpub\wwwroot\demoMA"`);
- Строка соединения с информационной базой `Srvr=server:1741;Ref=demo;` (параметр `-connstr "Srvr=server:1741;Ref=demo;"`);
- Остальные параметры публикации будут получены из файла-шаблона `template.vrd` (параметр `-descriptor template.vrd`).

Пример команды удаления публикации для IIS:

```
webinst -delete -iis -wsdir DemoWS
```

[Копировать в буфер обмена](#)

В данном примере выполняется удаление публикации, сделанной в виртуальном каталоге:

- Виртуальный каталог: `DemoWS` (параметр `-wsdir DemoWS`). Остальные параметры определяются автоматически по этому имени.

7.3. Настройка поддержки веб-клиента

Данный раздел содержит инструкции по настройке различных веб-серверов для работы с помощью веб-клиента. Будут описаны как действия, необходимые для публикации из конфигурирования, так и действия, необходимые для выполнения публикации с помощью утилиты `webinst`.

При описании публикации будут описываться значения, которые являются ключевыми для выполнения публикации. Остальные параметры необходимо настраивать при необходимости.

7.3.1. Для ОС Windows

Данный раздел содержит описание настройки веб-серверов, работающих под управлением ОС Windows, для работы веб-клиента.

Для публикации веб-клиента следует установить флажок [Публиковать тонкий клиент и веб-клиент](#).

7.3.1.1. Internet Information Services

Кроме указания параметров публикации (описаны ниже), следует дополнительно выполнить следующие настройки:

- дать права на чтение для пользователя, от лица которого выполняются запросы (пользователь `IUSR_<PC_NAME>` для IIS версий 5.1 и 6.0 или группа `IIS_IUSRS` для IIS версий 7.0 и старше), на каталог `bin` файлов конкретной версии системы «1С:Предприятие»;
- дать права на модификацию пользователю, от лица которого выполняются запросы (пользователь `IUSR_<PC_NAME>` для IIS версий 5.1 и 6.0 или группа `IIS_IUSRS` для IIS версий 7.0 и старше), на каталог информационной базы (только в случае файлового варианта).

ПРИМЕЧАНИЕ. Текст `<PC_NAME>` в имени пользователя `IUSR_<PC_NAME>` означает имя компьютера, на котором установлен IIS. Так, для компьютера с именем `IIS-COMP` имя пользователя будет выглядеть следующим образом: `IUSR_IIS-COMP`.

Дополнительные особенности настройки IIS см. [здесь](#).

7.3.1.1.1. Диалог публикации

В поле **Веб-сервер** нужно указать [Internet Information Services](#). Если необходима аутентификация операционной системы на веб-сервере, установить соответствующий флажок ([Использовать аутентификацию операционной системы на веб-сервере](#)).

При необходимости следует настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

7.3.1.1.2. Утилита webinst

Для настройки веб-сервера IIS с помощью утилиты `webinst` (см. [здесь](#)) следует выполнить следующую команду (параметры даны для примера, их следует заменить реальными значениями).

Пример:

```
webinst -publish -iis -wsdir demo -dir "c:\inetpub\demo" -connstr "Srvr=server:1741;Ref=demo;" -osauth
```

[Копировать в буфер обмена](#)

7.3.1.2. Apache версии 2.0

Кроме указания параметров публикации (описаны ниже), следует дополнительно выполнить следующие настройки:

- дать права на чтение для пользователя, от лица которого работает веб-сервер, на каталог `bin` файлов конкретной версии системы «1С:Предприятие»;
- дать права на модификацию пользователю, от лица которого работает веб-сервер, на каталог информационной базы (только в случае файлового варианта).

7.3.1.2.1. Диалог публикации

В поле **Веб-сервер** нужно указать [Apache 2.0](#).

При необходимости следует настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

7.3.1.2.2. Утилита webinst

Для настройки веб-сервера Apache версии 2.0 с помощью утилиты `webinst` (см. [здесь](#)) следует выполнить следующую команду (параметры даны для примера, их следует заменить реальными значениями).

Пример:

```
webinst -publish -apache2 -wsdir demo -connstr "Srvr=server:1741;Ref=demo;" -dir "c:\apache.www\demows" -confpath "C:\Program Files\Apache Software Foundat
```

[Копировать в буфер обмена](#)

7.3.1.3. Apache версии 2.2

Кроме указания параметров публикации (описаны ниже), следует дополнительно выполнить следующие настройки:

- дать права на чтение для пользователя, от лица которого работает веб-сервер, на каталог `bin` файлов конкретной версии системы «1С:Предприятие»;
- дать права на модификацию пользователю, от лица которого работает веб-сервер, на каталог информационной базы (только в случае файлового варианта).

7.3.1.3.1. Диалог публикации

В поле **Веб-сервер** нужно указать [Apache 2.2](#).

При необходимости следует настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

7.3.1.3.2. Утилита webinst

Для настройки веб-сервера Apache версии 2.2 с помощью утилиты `webinst` (см. [здесь](#)) следует выполнить следующую команду (параметры даны для примера, их следует заменить реальными значениями).

Пример:

```
webinst -publish -apache22 -wsdir demo -connstr "Srvr=server:1741;Ref=demo;" -dir "c:\apache.www\demows" -confpath "C:\Program Files\Apache Software Foundat
```

[Копировать в буфер обмена](#)

7.3.1.4. Настройка автоматической аутентификации для веб-сервера Apache

Подробно описание этой настройки см. [здесь](#).

7.3.2. Для ОС Linux

Данный раздел содержит описание настройки веб-серверов, работающих под управлением ОС Linux, для работы веб-клиента. После выполнения публикации необходимо предоставить пользователю, от лица которого запускается Apache, права на каталог исполняемых файлов (`/opt/1C/v8.3/i386/` для 32-разрядной версии или `/opt/1C/v8.3/x86_64/` для 64-разрядной версии) конкретной версии системы «1С:Предприятие» (чтение и выполнение). В случае файлового варианта информационной базы необходимо дать права на модификацию каталога информационной базы пользователю, от лица которого работает веб-сервер.

7.3.2.1. Apache версии 2.0

7.3.2.1.1. Диалог публикации

В поле **Веб-сервер** нужно указать [Apache 2.0](#).

При необходимости следует настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

7.3.2.1.2. Утилита webinst

Для настройки веб-сервера Apache версии 2.0 с помощью утилиты `webinst` (см. [здесь](#)) следует выполнить следующую команду (параметры даны для примера, их следует заменить реальными значениями).

Пример:

```
webinst -apache2 -wsdir DemoWS -dir /var/www/DemoWS -connstr "Srvr=server:1741;Ref=demo;" -confpath /etc/apache2/httpd.conf
```

7.3.2.2. Apache версии 2.2

7.3.2.2.1. Диалог публикации

В поле **Веб-сервер** нужно указать **Apache 2.2**.

При необходимости следует настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

7.3.2.2.2. Утилита webinst

Для настройки веб-сервера Apache версии 2.0 с помощью утилиты **webinst** (см. [здесь](#)) следует выполнить следующую команду (параметры даны для примера, их следует заменить реальными значениями).

Пример:

```
webinst -apache22 -wsdir DemoWS -dir /var/www/DemoWS -connstr "Srvr=server:1741;Ref=demo;" -confpath /etc/apache2/apache.conf
```

Копировать в буфер обмена

7.3.2.3. Настройка автоматической аутентификации для веб-сервера Apache

Подробно описание этой настройки см. [здесь](#).

7.4. Настройка поддержки Web-сервисов

Настройка поддержки Web-сервисов заключается в настройке используемого веб-сервера на работу с Web-сервисами и в установке прав доступа к каталогам исполняемых файлов и базы данных (для файлового варианта работы).

Для публикации Web-сервисов следует на закладке **Web-сервисы** установить флажок **Публиковать Web-сервисы**, а также в таблице, расположенной под флажком, выбрать публикуемые сервисы.

7.4.1. Для ОС Windows

Данный раздел содержит описание публикации Web-сервисов для веб-серверов, работающих под управлением ОС Windows. Считается, что веб-сервер уже установлен.

ПРИМЕЧАНИЕ. Для установки веб-сервера IIS может потребоваться дистрибутив используемой операционной системы.

7.4.1.1. Internet Information Services

Кроме указания параметров публикации (описаны ниже), следует дополнительно выполнить следующие настройки:

- дать права на чтение для пользователя, от лица которого выполняются запросы (**IUSR_<PC_NAME>** для IIS версий 5.1 или 6.0 или группу **IIS_IUSRS** для IIS версий 7.0 и старше), на каталог **bin** файлов конкретной версии системы «1С:Предприятие»;
- дать права на модификацию пользователю, от лица которого выполняются запросы (**IUSR_<PC_NAME>** для IIS версий 5.1 или 6.0 или группу **IIS_IUSRS** для IIS версий 7.0 и старше), на каталог информационной базы (только в случае файлового варианта).

ПРИМЕЧАНИЕ. Текст **<PC_NAME>** в имени пользователя **IUSR_<PC_NAME>** означает имя компьютера, на котором установлен IIS. Так, для компьютера с именем **IIS-COMP** имя пользователя будет выглядеть следующим образом: **IUSR_IIS-COMP**.

Дополнительные особенности настройки IIS см. [здесь](#).

7.4.1.1.1. Диалог публикации

В поле **Веб-сервер** нужно указать **Internet Information Services**.

При необходимости следует настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

7.4.1.1.2. Утилита webinst

Перед публикацией необходимо создать файл-шаблон:

- В поле **Веб-сервер** указать **Internet Information Services**.
- Установить флаг **Публиковать Web-сервисы**.
- Выбрать параметры публикуемых Web-сервисов.
- При необходимости настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).
- С помощью кнопки **Сохранить** выполнить сохранение файла-шаблона. Укажем имя файла-шаблона **iis-template.vrd**.

Затем выполнить публикацию с использованием файла-шаблона.

Пример:

```
webinst -publish -iis -wsdir demo-ws -dir "c:\inetpub\demo-ws" -connstr "Srvr=server:1741;Ref=demo;" -descriptor iis-template.vrd
```

Копировать в буфер обмена

7.4.1.2. Apache версии 2.0

Необходимо дать права пользователю, от лица которого запускается Apache, на каталог **bin** файлов конкретной версии системы «1С:Предприятие» (чтение и выполнение) и каталог информационной базы (чтение и запись, в случае файлового варианта).

7.4.1.2.1. Диалог публикации

В поле **Веб-сервер** нужно указать **Apache 2.0**.

При необходимости следует настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

7.4.1.2.2. Утилита webinst

Перед публикацией необходимо создать файл-шаблон:

- В поле **Веб-сервер** указать **Apache 2.0**.
- Установить флаг **Публиковать Web-сервисы**.
- Выбрать параметры публикуемых Web-сервисов.

- При необходимости настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

- С помощью кнопки **Сохранить** выполнить сохранение файла-шаблона. Укажем имя файла-шаблона `apache-template.vrd`.

Затем выполнить публикацию с использованием файла-шаблона.

Пример:

```
webinst -publish -apache2 -wsdir demo-ws -dir "c:\inetpub\demo-ws" -connstr "Srvr=server:1741;Ref=demo;" -descriptor apache-template.vrd
```

[Копировать в буфер обмена](#)

7.4.1.3. Apache версии 2.2

Необходимо дать права пользователю, от лица которого запускается Apache, на каталог `bin` файлов конкретной версии системы «1С:Предприятие» (чтение и выполнение) и каталог информационной базы (чтение и запись, в случае файлового варианта).

7.4.1.3.1. Диалог публикации

В поле **Веб-сервер** нужно указать **Apache 2.2**.

При необходимости следует настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

7.4.1.3.2. Утилита webinst

Перед публикацией необходимо создать файл-шаблон:

- В поле **Веб-сервер** указать **Apache 2.2**.
- Установить флаг **Публиковать Web-сервисы**.
- Выбрать параметры публикуемых Web-сервисов.
- При необходимости настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).
- С помощью кнопки **Сохранить** выполнить сохранение файла-шаблона. Укажем имя файла-шаблона `apache-template.vrd`.

Затем выполнить публикацию с использованием файла-шаблона.

Пример:

```
webinst -publish -apache22 -wsdir demo-ws -dir "c:\inetpub\demo-ws" -connstr "Srvr=server:1741;Ref=demo;" -descriptor apache-template.vrd
```

[Копировать в буфер обмена](#)

7.4.2. Для ОС Linux

Данный раздел содержит описание публикации Web-сервисов для веб-серверов, работающих под управлением ОС Linux. Считается, что веб-сервер уже установлен.

Для публикации менеджера сервисов следует на закладке **Web-сервисы** установить флажок **Публиковать Web-сервисы**, а также в таблице, расположенной под флажком, выбрать публикуемые сервисы.

После выполнения публикации необходимо предоставить пользователю, от лица которого запускается Apache, права на каталог исполняемых файлов (`/opt/1C/v8.3/i386/` для 32-разрядной версии или `/opt/1C/v8.3/x86_64/` для 64-разрядной версии) конкретной версии системы «1С:Предприятие» (чтение и выполнение). В случае файлового варианта информационной базы необходимо дать права на модификацию каталога информационной базы пользователю, от лица которого работает веб-сервер.

7.4.2.1. Apache версии 2.0

7.4.2.1.1. Диалог публикации

В поле **Веб-сервер** нужно указать **Apache 2.0**.

При необходимости следует настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

7.4.2.1.2. Утилита webinst

Перед публикацией необходимо создать файл-шаблон (из конфигулятора):

- В поле **Веб-сервер** указать **Apache 2.0**.
- Установить флаг **Публиковать Web-сервисы**.
- Выбрать параметры публикуемых Web-сервисов.
- При необходимости настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).
- С помощью кнопки **Сохранить** выполнить сохранение файла-шаблона. Укажем имя файла-шаблона `apache20-template.vrd`.

Затем выполнить публикацию с использованием файла-шаблона.

Пример:

```
webinst -publish -apache2 -wsdir demo-ws -dir /var/www/demo-ws -connstr "Srvr=server:1741;Ref=demo;" -confpath /etc/apache2/httpd.conf -descriptor apache20-1
```

[Копировать в буфер обмена](#)

7.4.2.2. Apache версии 2.2

7.4.2.2.1. Диалог публикации

В поле **Веб-сервер** нужно указать **Apache 2.2**.

При необходимости следует настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

7.4.2.2.2. Утилита webinst

Перед публикацией необходимо создать файл-шаблон (из конфигулятора):

- В поле **Веб-сервер** указать **Apache 2.2**.
- Установить флаг **Публиковать Web-сервисы**.
- Выбрать параметры публикуемых Web-сервисов.
- При необходимости настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

- С помощью кнопки [Сохранить](#) выполнить сохранение файла-шаблона. Укажем имя файла-шаблона [apache22-template.vrd](#).

Затем выполнить публикацию с использованием файла-шаблона.

Пример:

```
webinst -publish -apache22 -wsdir demo-ws -dir /var/www/demo-ws -connstr "Srvr=server:1741;Ref=demo;" -confpath /etc/apache2/httpd.conf -descriptor apache22
```

[Копировать в буфер обмена](#)

7.5. Настройка поддержки стандартного интерфейса OData

Настройка поддержки стандартного интерфейса OData заключается в настройке используемого веб-сервера и в установке прав доступа к каталогам исполняемых файлов и базы данных (для файлового варианта работы).

Для публикации стандартного интерфейса OData следует на закладке [Основные](#) установить флажок [Публиковать стандартный интерфейс OData](#).

7.5.1. Для ОС Windows

Данный раздел содержит описание публикации стандартного интерфейса OData для веб-серверов, работающих под управлением ОС Windows. Считается, что веб-сервер уже установлен.

ПРИМЕЧАНИЕ. Для установки веб-сервера IIS может потребоваться дистрибутив используемой операционной системы.

7.5.1.1. Internet Information Services

Кроме указания параметров публикации (описаны ниже), следует дополнительно выполнить следующие настройки:

- дать права на чтение для пользователя, от лица которого выполняются запросы ([IUSR_<PC_NAME>](#) для IIS версий 5.1 или 6.0 или группу [IIS_IUSR](#) для IIS версий 7.0 и старше), на каталог [bin](#) файлов конкретной версии системы «1С:Предприятие»;
- дать права на модификацию пользователю, от лица которого выполняются запросы ([IUSR_<PC_NAME>](#) для IIS версий 5.1 или 6.0 или группу [IIS_IUSR](#) для IIS версий 7.0 и старше), на каталог информационной базы (только в случае файлового варианта).

ПРИМЕЧАНИЕ. Текст [<PC_NAME>](#) в имени пользователя [IUSR_<PC_NAME>](#) означает имя компьютера, на котором установлен IIS. Так, для компьютера с именем [IIS-COMP](#) имя пользователя будет выглядеть следующим образом: [IUSR_IIS-COMP](#).

Дополнительные особенности настройки IIS см. [здесь](#).

7.5.1.1.1. Диалог публикации

В поле [Веб-сервер](#) нужно указать [Internet Information Services](#). Если необходима аутентификация операционной системы на веб-сервере, установить соответствующий флажок ([Использовать аутентификацию операционной системы на веб-сервере](#)).

При необходимости следует настроить остальные параметры публикации на вкладке [Прочие](#) диалога публикации на веб-сервере (см. [здесь](#)).

7.5.1.1.2. Утилита webinst

Перед публикацией необходимо создать файл-шаблон:

- В поле [Веб-сервер](#) указать [Internet Information Services](#).
- Установить флаг [Публиковать стандартный интерфейс OData](#).
- При необходимости указать возможность [Использовать аутентификацию операционной системы на веб-сервере](#);
- При необходимости настроить остальные параметры публикации на вкладке [Прочие](#) диалога публикации на веб-сервере (см. [здесь](#)).
- С помощью кнопки [Сохранить](#) выполнить сохранение файла-шаблона. Укажем имя файла-шаблона [iis-template.vrd](#).

Затем выполнить публикацию с использованием файла-шаблона.

Пример:

```
webinst -publish -iis -wsdir demo-ws -dir "c:\inetpub\demo-ws" -connstr "Srvr=server:1741;Ref=demo;" -descriptor iis-template.vrd
```

[Копировать в буфер обмена](#)

7.5.1.2. Apache 2.0

Кроме указания параметров публикации (описаны ниже), следует дополнительно выполнить следующие настройки:

- дать права на чтение для пользователя, от лица которого работает веб-сервер, на каталог [bin](#) файлов конкретной версии системы «1С:Предприятие»;
- дать права на модификацию пользователю, от лица которого работает веб-сервер, на каталог информационной базы (только в случае файлового варианта).

7.5.1.2.1. Диалог публикации

В поле [Веб-сервер](#) нужно указать [Apache 2.0](#).

При необходимости следует настроить остальные параметры публикации на вкладке [Прочие](#) диалога публикации на веб-сервере (см. [здесь](#)).

7.5.1.2.2. Утилита webinst

Перед публикацией необходимо создать файл-шаблон:

- В поле [Веб-сервер](#) указать [Apache 2.0](#).
- Установить флаг [Публиковать стандартный интерфейс OData](#).
- При необходимости настроить остальные параметры публикации на вкладке [Прочие](#) диалога публикации на веб-сервере (см. [здесь](#)).
- С помощью кнопки [Сохранить](#) выполнить сохранение файла-шаблона. Укажем имя файла-шаблона [apache-template.vrd](#).

Затем выполнить публикацию с использованием файла-шаблона.

Пример:

```
webinst -publish -apache2 -wsdir demo-ws -dir "c:\inetpub\demo-ws" -connstr "Srvr=server:1741;Ref=demo;" -descriptor apache-template.vrd
```

[Копировать в буфер обмена](#)

7.5.1.3. Apache 2.2

Кроме указания параметров публикации (описаны ниже), следует дополнительно выполнить следующие настройки:

- дать права на чтение для пользователя, от лица которого работает веб-сервер, на каталог `bin` файлов конкретной версии системы «1С:Предприятие»;
- дать права на модификацию пользователю, от лица которого работает веб-сервер, на каталог информационной базы (только в случае файлового варианта).

7.5.1.3.1. Диалог публикации

В поле **Веб-сервер** нужно указать **Apache 2.2**.

При необходимости следует настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

7.5.1.3.2. Утилита `webinst`

Перед публикацией необходимо создать файл-шаблон:

- В поле **Веб-сервер** указать **Apache 2.2**.
- Установить флаг **Публиковать стандартный интерфейс OData**.
- При необходимости настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).
- С помощью кнопки **Сохранить** выполнить сохранение файла-шаблона. Укажем имя файла-шаблона `apache-template.vrd`.

Затем выполнить публикацию с использованием файла-шаблона.

Пример:

[Копировать в буфер обмена](#)

```
webinst -publish -apache22 -wsdir demo-ws -dir "c:\inetpub\demo-ws" -connstr "Srvr=server:1741;Ref=demo;" -descriptor apache-template.vrd
```

7.5.2. Для ОС Linux

7.5.2.1. Apache 2.0

7.5.2.1.1. Диалог публикации

В поле **Веб-сервер** нужно указать **Apache 2.0**.

При необходимости следует настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

7.5.2.1.2. Утилита `webinst`

Перед публикацией необходимо создать файл-шаблон:

- В поле **Веб-сервер** указать **Apache 2.0**.
- Установить флаг **Публиковать стандартный интерфейс OData**.
- При необходимости настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).
- С помощью кнопки **Сохранить** выполнить сохранение файла-шаблона. Укажем имя файла-шаблона `apache-template.vrd`.

Затем выполнить публикацию с использованием файла-шаблона.

Пример:

[Копировать в буфер обмена](#)

```
webinst -publish -apache2 -wsdir demo-ws -dir /var/www/demo-ws -connstr "Srvr=server:1741;Ref=demo;" -confpath /etc/apache2/httpd.conf -descriptor apache20-
```

7.5.2.2. Apache 2.2

7.5.2.2.1. Диалог публикации

В поле **Веб-сервер** нужно указать **Apache 2.2**.

При необходимости следует настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

7.5.2.2.2. Утилита `webinst`

Перед публикацией необходимо создать файл-шаблон:

- В поле **Веб-сервер** указать **Apache 2.2**.
- Установить флаг **Публиковать стандартный интерфейс OData**.
- При необходимости настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).
- С помощью кнопки **Сохранить** выполнить сохранение файла-шаблона. Укажем имя файла-шаблона `apache-template.vrd`.

Затем выполнить публикацию с использованием файла-шаблона.

Пример:

[Копировать в буфер обмена](#)

```
webinst -publish -apache22 -wsdir demo-ws -dir /var/www/demo-ws -connstr "Srvr=server:1741;Ref=demo;" -confpath /etc/apache2/httpd.conf -descriptor apache20-
```

7.6. Настройка поддержки HTTP-сервисов

Настройка поддержки HTTP-сервисов заключается в настройке используемого веб-сервера на работу с HTTP-сервисами и в установке прав доступа к каталогам исполняемых файлов и базы данных (для файлового варианта работы).

Для публикации HTTP-сервисов следует на закладке **HTTP сервисы** установить флажок **Публиковать HTTP-сервисы по умолчанию**, а также в таблице, расположенной под флажком, выбрать публикуемые сервисы.

7.6.1. Для ОС Windows

Данный раздел содержит описание публикации HTTP-сервисов для веб-серверов, работающих под управлением ОС Windows. Считается, что веб-сервер уже установлен.

ПРИМЕЧАНИЕ. Для установки веб-сервера IIS может потребоваться дистрибутив используемой операционной системы.

7.6.1.1. Internet Information Services

Кроме указания параметров публикации (описаны ниже), следует дополнительно выполнить следующие настройки:

- дать права на чтение для пользователя, от лица которого выполняются запросы (`IUSR_<PC_NAME>` для IIS версий 5.1 или 6.0 или группу `IIS_IUSRS` для IIS версий 7.0 и старше), на каталог `bin` файлов конкретной версии системы «1С:Предприятие»;
- дать права на модификацию пользователю, от лица которого выполняются запросы (`IUSR_<PC_NAME>` для IIS версий 5.1 или 6.0 или группу `IIS_IUSRS` для IIS версий 7.0 и старше), на каталог информационной базы (только в случае файлового варианта).

ПРИМЕЧАНИЕ. Текст `<PC_NAME>` в имени пользователя `IUSR_<PC_NAME>` означает имя компьютера, на котором установлен IIS. Так, для компьютера с именем `IIS-COMP` имя пользователя будет выглядеть следующим образом: `IUSR_IIS-COMP`.

Дополнительные особенности настройки IIS см. [здесь](#).

7.6.1.1.1. Диалог публикации

В поле **Веб-сервер** нужно указать [Internet Information Services](#).

При необходимости следует настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

7.6.1.1.2. Утилита webinst

Перед публикацией необходимо создать файл-шаблон:

- В поле **Веб-сервер** указать [Internet Information Services](#).
- Установить флаг **Публиковать HTTP-сервисы**.
- Выбрать параметры публикуемых HTTP-сервисов.
- При необходимости настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).
- С помощью кнопки **Сохранить** выполнить сохранение файла-шаблона. Укажем имя файла-шаблона `iis-template.vrd`.

Затем выполнить публикацию с использованием файла-шаблона.

Пример:

```
webinst -publish -iis -wsdir demo-hs -dir "c:\inetpub\demo-ws" -connstr "Srvr=server:1741;Ref=demo;" -descriptor iis-template.vrd
```

[Копировать в буфер обмена](#)

7.6.1.2. Apache версии 2.0

Необходимо дать права пользователю, от лица которого запускается Apache, на каталог `bin` файлов конкретной версии системы «1С:Предприятие» (чтение и выполнение) и каталог информационной базы (чтение и запись, в случае файлового варианта).

7.6.1.2.1. Диалог публикации

В поле **Веб-сервер** нужно указать [Apache 2.0](#).

При необходимости следует настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

7.6.1.2.2. Утилита webinst

Перед публикацией необходимо создать файл-шаблон:

- В поле **Веб-сервер** указать [Apache 2.0](#).
- Установить флаг **Публиковать HTTP-сервисы**.
- Выбрать параметры публикуемых HTTP-сервисов.
- При необходимости настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).
- С помощью кнопки **Сохранить** выполнить сохранение файла-шаблона. Укажем имя файла-шаблона `apache-template.vrd`.

Затем выполнить публикацию с использованием файла-шаблона.

Пример:

```
webinst -publish -apache2 -wsdir demo-ws -dir "c:\inetpub\demo-ws" -connstr "Srvr=server:1741;Ref=demo;" -descriptor apache-template.vrd
```

[Копировать в буфер обмена](#)

7.6.1.3. Apache версии 2.2

Необходимо дать права пользователю, от лица которого запускается Apache, на каталог `bin` файлов конкретной версии системы «1С:Предприятие» (чтение и выполнение) и каталог информационной базы (чтение и запись, в случае файлового варианта).

7.6.1.3.1. Диалог публикации

В поле **Веб-сервер** нужно указать [Apache 2.2](#).

При необходимости следует настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).

7.6.1.3.2. Утилита webinst

Перед публикацией необходимо создать файл-шаблон:

- В поле **Веб-сервер** указать [Apache 2.2](#).
- Установить флаг **Публиковать HTTP-сервисы**.
- Выбрать параметры публикуемых HTTP-сервисов.
- При необходимости настроить остальные параметры публикации на вкладке **Прочие** диалога публикации на веб-сервере (см. [здесь](#)).
- С помощью кнопки **Сохранить** выполнить сохранение файла-шаблона. Укажем имя файла-шаблона `apache-template.vrd`.

Затем выполнить публикацию с использованием файла-шаблона.

Пример:

```
webinst -publish -apache22 -wsdir demo-ws -dir "c:\inetpub\demo-ws" -connstr "Srvr=server:1741;Ref=demo;" -descriptor apache-template.vrd
```

[Копировать в буфер обмена](#)

7.6.2. Для ОС Linux

Данный раздел содержит описание публикации Web-сервисов для веб-серверов, работающих под управлением ОС Linux. Считается, что веб-сервер уже установлен.

Для публикации менеджера сервисов следует на закладке [HTTP-сервисы](#) установить флажок [Публиковать HTTP-сервисы по умолчанию](#), а также в таблице, расположенной под флажком, выбрать публикуемые сервисы.

После выполнения публикации необходимо предоставить пользователю, от лица которого запускается Apache, права на каталог исполняемых файлов (/opt/1C/v8.3/i386/ для 32-разрядной версии или /opt/1C/v8.3/x86_64/ для 64-разрядной версии) конкретной версии системы «1С:Предприятие» (чтение и выполнение). В случае файлового варианта информационной базы необходимо дать права на модификацию каталога информационной базы пользователю, от лица которого работает веб-сервер.

7.6.2.1. Apache версии 2.0

7.6.2.1.1. Диалог публикации

В поле [Веб-сервер](#) нужно указать [Apache 2.0](#).

При необходимости следует настроить остальные параметры публикации на вкладке [Прочие](#) диалога публикации на веб-сервере (см. [здесь](#)).

7.6.2.1.2. Утилита webinst

Перед публикацией необходимо создать файл-шаблон (из конфигуратора):

- В поле [Веб-сервер](#) указать [Apache 2.0](#).
- Установить флаг [Публиковать HTTP-сервисы](#).
- Выбрать параметры публикуемых HTTP-сервисов.
- При необходимости настроить остальные параметры публикации на вкладке [Прочие](#) диалога публикации на веб-сервере (см. [здесь](#)).
- С помощью кнопки [Сохранить](#) выполнить сохранение файла-шаблона. Укажем имя файла-шаблона [apache20-template.vrd](#).

Затем выполнить публикацию с использованием файла-шаблона.

Пример:

```
webinst -publish -apache2 -wsdir demo-ws -dir /var/www/demo-ws -connstr "Srvr=server:1741;Ref=demo;" -confpath /etc/apache2/httpd.conf -descriptor apache20-1
```

[Копировать в буфер обмена](#)

7.6.2.2. Apache версии 2.2

7.6.2.2.1. Диалог публикации

В поле [Веб-сервер](#) нужно указать [Apache 2.2](#).

При необходимости следует настроить остальные параметры публикации на вкладке [Прочие](#) диалога публикации на веб-сервере (см. [здесь](#)).

7.6.2.2.2. Утилита webinst

Перед публикацией необходимо создать файл-шаблон (из конфигуратора):

- В поле [Веб-сервер](#) указать [Apache 2.2](#).
- Установить флаг [Публиковать HTTP-сервисы](#).
- Выбрать параметры публикуемых HTTP-сервисов.
- При необходимости настроить остальные параметры публикации на вкладке [Прочие](#) диалога публикации на веб-сервере (см. [здесь](#)).
- С помощью кнопки [Сохранить](#) выполнить сохранение файла-шаблона. Укажем имя файла-шаблона [apache22-template.vrd](#).

Затем выполнить публикацию с использованием файла-шаблона.

Пример:

```
webinst -publish -apache22 -wsdir demo-ws -dir /var/www/demo-ws -connstr "Srvr=server:1741;Ref=demo;" -confpath /etc/apache2/httpd.conf -descriptor apache22
```

[Копировать в буфер обмена](#)

7.7. Безопасность при работе через Интернет

7.7.1. Аутентификация

В общем случае, схема обращения клиента к интернет-сервису, выглядит следующим образом:

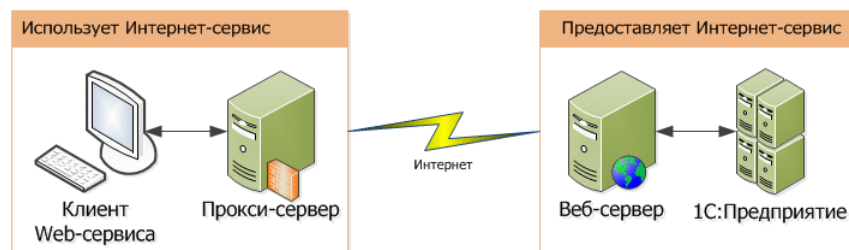


Рис. 75. Соединения интернет-сервиса

При этом можно говорить о трех различных видах аутентификации:

- На прокси-сервере – эта аутентификация не имеет прямого отношения к использованию веб-сервера, но про нее необходимо помнить, если необходимо использовать интернет-сервис из сети, находящейся за прокси-сервером.
- На веб-сервере – в этом случае могут быть использованы следующие виды аутентификаций:

- **Анонимная аутентификация** – в этом случае все запросы, поступающие от веб-сервера, выполняются под специальным пользователем, который олицетворяет «анонимное» соединение.

В этом случае аутентификация в «1С:Предприятии» выполняется с использованием имени пользователя и пароля, переданного в HTTP-запросе.

- **Basic аутентификация** – в этом случае клиент интернет-сервиса передает для аутентификации на веб-сервер имя пользователя и пароль в HTTP-запросе, который формируется при обращении к веб-серверу.

Для успешного выполнения данного типа аутентификации, имя пользователя и пароль, которые используются для доступа к «1С:Предприятию», должны использоваться и для доступа к веб-серверу. Если пользователь, параметры которого передаются в HTTP-запросе, не может получить доступ к веб-серверу, значит, он не сможет пользоваться интернет-сервисом.

• **Аутентификация ОС** – в этом случае веб-сервер определяет, от имени какого пользователя ОС интернет-сервис выполняет обращение к «1С:Предприятию» и дальше используются именно эти данные.

В этом случае веб-сервер определяет пользователя ОС, который пытается получить доступ к веб-серверу и затем передает «1С:Предприятию» и характеристики пользователя ОС и данные, переданные в HTTP-запросе к интернет-сервису. Если в HTTP-запросе указаны имя пользователя и пароль, то для аутентификации используются именно они и данные пользователя ОС не используются. Если имя пользователя и пароль в HTTP-запросе не указаны – используются данные определенного пользователя ОС.

• **Аутентификация «1С:Предприятия»**. Для выполнения этой аутентификации расширение веб-сервера использует имя и пароль пользователя, которые переданы веб-сервером (в случае использования на веб-сервере Basic аутентификации или аутентификации ОС). При использовании на веб-сервере анонимной аутентификации, «1С:Предприятие» запросит у вызывающей стороны Basic аутентификацию. «1С:Предприятие» ожидает, что логин и пароль пользователя будет передан в кодировке UTF-8.

Если обращение к интернет-сервису выполняется из веб-браузера Microsoft Internet Explorer, то не рекомендуется для задания имени пользователя и пароля использовать нелатинские символы.

При взаимодействии с веб-сервером имеется возможность организовать работу по защищенному каналу (см. [здесь](#)).

При использовании файлового варианта информационной базы, пользователи, от имени которых выполняется доступ к системе, должны иметь доступ к исполнению файлов нужной версии «1С:Предприятия» и права на чтение и изменение данных в каталоге информационной базы.

7.7.2. Работа по защищенному каналу

При взаимодействии клиента и сервера интернет-сервисов обмен данными может вестись по защищенному каналу. Защищенные каналы связи позволяют предотвратить несанкционированный просмотр и изменение данных. Защищенный канал работает по протоколу обмена SSL 3.0/TLS 1.0.

SSL (Secure Socket Layer) – протокол, использующийся для обеспечения защищенного взаимодействия между клиентом и сервером. SSL базируется:

- на взаимной аутентификации клиента и сервера для того, чтобы и клиент, и сервер были уверены в том, что они те, за кого себя выдают;
- цифровых подписях, для обеспечения целостности данных (защиты данных от несанкционированного изменения);
- шифровании, для обеспечения конфиденциальности данных (защиты данных от несанкционированного просмотра).

SSL-протокол поддерживает различные варианты шифров, цифровых подписей, сертификатов и т. д. для того, чтобы наиболее адекватно обеспечить защищенный канал требуемой стойкости.

SSL-протокол использует SSL-сессию для установки защищенного соединения между клиентом и сервером. Сессия устанавливается путем обмена между клиентом и сервером последовательностью сообщений. При установке сессии могут выполняться такие действия, как:

- установка алгоритмов шифрования, которые будут использоваться для шифрования и цифровой подписи передаваемых данных;
- установка сессионного ключа;
- аутентификация сервера на клиенте;
- аутентификация клиента на сервере.

Для выполнения аутентификации клиента на сервере и сервера на клиенте, SSL использует сертификаты. Сертификат является документом, в котором описывается набор параметров того, для кого выполняется аутентификация. Например, сертификат может содержать имя пользователя или имя веб-сайта сервера. Сертификат также имеет цифровую подпись, которая используется для проверки его действительности. Для предотвращения возможности неконтролируемой выдачи сертификатов используются цепочки сертификатов. Началом цепочки сертификатов выступает Центр Сертификации – организация, занимающаяся выдачей сертификатов. Если определенному пользователю требуется сертификат, он посылает запрос в Центр Сертификации на выдачу сертификата. Центр Сертификации выдает готовый сертификат, который подписан собственным закрытым ключом. Пользователь, которому выдан сертификат, может, в свою очередь, выступать в качестве Центра Сертификации для других пользователей. Таким образом, образуется цепочка сертификатов, корнем которой является корневой Центр Сертификации, как правило, общеизвестная организация. Чтобы клиент принял данный сертификат, он должен быть в списке сертификатов, которым доверяет этот клиент. В списке может быть как сам сертификат, так и любой сертификат из цепочки сертификатов данного сертификата. Как правило, это сертификат корневого Центра Сертификации. Следует помнить, что «1С:Предприятие» корректно работает с сертификатами, только если поля сертификатов содержат данные в US ASCII или символы, закодированные при помощи Punycode. Поля сертификатов не должны содержать данные в Unicode.

Одним из наиболее общепотребимых применений протокола SSL является его использование для передачи HTTP-запросов (HTTPS-протокол). В этом случае URL-схема адресации для таких ресурсов – https, а порт по умолчанию – 443.

Клиентская часть механизма web-сервисов автоматически по URL-схеме (https) расположения web-сервиса определяет, что взаимодействие с таким web-сервисом должно вестись по защищенному каналу связи. Клиентская часть также требует, чтобы с сервером был связан действительный сертификат, выданный известным клиенту Центром Сертификации.

Сертификат сервера является действительным, если его цифровая подпись соответствует содержанию сертификата, его дата действия не просрочена и веб-сайт, на который выдан данный сертификат, соответствует веб-сайту сервера. Если сертификат не является действительным, например, веб-сайт сертификата не соответствует веб-сайту сервера, то клиент не сможет взаимодействовать по протоколу SSL с web-сервисами этого сайта.

Для того чтобы включить возможность работы по протоколу SSL, нужно:

- Получить серверный сертификат для веб-сайта, для которого планируется использовать SSL. Сертификат выдается Центром Сертификации и привязывается к этому веб-сайту.
- Для веб-сервера должна быть включена поддержка SSL.
- Для того чтобы приложение, использующее Web-сервис, использовало защищенное соединение, необходимо явно указать это при подключении к Web-сервису. Для этого при создании объектов **WSОпределения** и **WSПрокси** необходимо указать параметр **ЗащищенноеСоединение**. При использовании защищенного соединения следует указать в качестве значения этого параметра один из объектов: **ЗащищенноеСоединениеOpenSSL** или **ЗащищенноеСоединениеNSS**.

7.8. Настройка поддержки OpenID-аутентификации

7.8.1. Настройки для использования OpenID

Если информационная база использует OpenID-аутентификацию, то в файле **default.vrd** (с помощью которого выполнена публикация информационной базы на веб-сервере) необходимо указать адрес OpenID-провайдера, с помощью которого происходит аутентификация. Для этого предназначены элементы **<openid>** и **<rely>**.

Пример:

```
<?xml version="1.0" encoding="UTF-8"?>
<point xmlns="http://v8.1c.ru/8.2/virtual-resource-system"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  base="/demo"
  id="Srvr=&quot;tcp://Server&quot;;Ref=&quot;demo&quot;;"
  enable="false">
  <openid>
    <rely url="https://myserver.org/users-ib/e1cib/oid2op"/>
  </openid>
</point>
```

```
</openid>
</point>
```

Эти элементы описывают URL к провайдеру OpenID, который выполняет аутентификацию пользователя для информационной базы, использующей OpenID-аутентификацию. В данном примере в качестве провайдера OpenID выступает информационная база, опубликованная по адресу <https://myserver.org/users-ib>.

Подробное описание файла `default.vrd` см. [здесь](#).

Настройку этого параметра можно выполнить с помощью диалога публикации на веб-сервере (закладка [Прочие](#), группа [OpenID](#)) (см. [здесь](#)).

7.8.2. Настройки для провайдера OpenID

Если информационная база выступает в роли OpenID-провайдера, то в файл `default.vrd` (с помощью которого выполнена публикация информационной базы на веб-сервере) необходимо указать, что данная информационная база выступает в роли OpenID-провайдера. Для этого предназначены элементы `<openid>` и `<provider>`.

Пример:

```
<?xml version="1.0" encoding="UTF-8"?>
<point xmlns="http://v8.1c.ru/8.2/virtual-resource-system"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  base="/users-ib"
  ib="Srvr=&quot;tcp://Server&quot;;Ref=&quot;oidusers&quot;;" enable="false">
  <openid>
    <provider>
      <lifetime>432000</lifetime>
    </provider>
  </openid>
</point>
```

[Копировать в буфер обмена](#)

Эти элементы указывают:

- Что информационная база выступает в роли OpenID-провайдера.
- Что время жизни признака аутентифицированности составляет 432 000 секунд (или 5 суток).
- URL, который следует указать в элементе `<rely>` файла `default.vrd` (адрес OpenID-провайдера), может выглядеть следующим образом: <https://myserver.org/users-ib/e1cib/oid2op>. Такой вид URL будет иметь в том случае, если имя хоста, на котором опубликована информационная база, является myserver.org.

Подробное описание файла `default.vrd` см. [здесь](#).

Настройку этого параметра можно выполнить с помощью диалога публикации на веб-сервере (закладка [Прочие](#), группа [OpenID](#)) (см. [здесь](#)).

7.8.3. Дополнительный интерфейс для использования внешними ресурсами

К OpenID-провайдеру, реализованный «1С:Предприятием», можно обращаться по стандартному протоколу OpenID 2.0 с учетом некоторых особенностей:

- В запросах на интерактивную и неинтерактивную аутентификации (параметр `openid.mode` равен `checkid_immediate` или `checkid_setup`) параметрам `openid.claimed_id` и `openid.identity` должно быть установлено значение http://specs.openid.net/auth/2.0/identifier_select. Установка этого значения означает, что идентификатор пользователя определяет провайдер.
- Запросы на неинтерактивную аутентификацию с прочими значениями параметров `openid.claimed_id` и `openid.identity` приводят к запросу на интерактивную аутентификацию, во время которой значения `openid.claimed_id` и `openid.identity` определяет провайдер.

В OpenID-провайдере реализована форма для ввода имени пользователя и пароля при интерактивной аутентификации.

Также система предоставляет ряд команд, упрощающих использование OpenID-провайдера сторонними системами, описание которых приведено ниже. При описании команд используются следующие сокращения:

- **ProviderIB** – информационная база OpenID-провайдера;
- **RPID** – информационная база зависимой стороны OpenID.

Параметры запросов передаются в кодировке UTF-8.

Запрос XRDS-документа провайдера OpenID

Описание:

Выполняет получение XRDS-документа, описывающего свойства OpenID-провайдера.

Синтаксис:

```
https://hostname/ProviderIB/e1cib/oid2op
```

Возвращаемое значение:

XRDS-документ, описывающий свойства OpenID-провайдера.

Запрос XRDS-документа зависимой стороны OpenID

Описание:

Выполняет получение XRDS-документа, описывающего свойства зависимой стороны OpenID.

Синтаксис:

```
https://hostname/RPIB/e1cib/oid2rp
```

Возвращаемое значение:

XRDS-документ, описывающий свойства зависимой стороны OpenID.

Запрос аутентификации

Описание:

Выполняет запрос аутентификации.

Синтаксис:

```
https://hostname/ProviderIB/e1cib/oid2op?cmd=auth
```

Параметры:*openid.auth.user* **обязательный**

Имя пользователя, как оно задано в базе OpenID-провайдера.

openid.auth.pwd **обязательный**

Пароль пользователя.

opeind.auth.short **необязательный**Если параметр установлен в значение `true`, то аутентификация выполняется в рамках сессии веб-браузера, но не больше, чем значение параметра `lifetime` файла `default.vrd`, описывающего публикацию информационной базы OpenID-провайдера.*openid.auth.check* **необязательный**Требуется выполнить проверку ответа на данный запрос (параметр установлен в значение `true`). Имеет смысл только в том случае, если указан параметр `openid.return_to`.*openid.return_to* **необязательный**

Содержит URL, на который выполняется переход после обработки запроса.

Возвращаемое значение:Если не указан параметр `openid.return_to`, то возвращается пустой документ с кодом состояния HTTP:

- 200 – аутентификация прошла успешно;
- 400 – аутентификация не выполнена.

Если параметр `openid.return_to` указан, то выполняется переадресация на адрес, указанный в параметре. В случае успешной аутентификации к URL добавляются следующие параметры:

- `openid.auth.user` с именем пользователя в качестве значения;
- `openid.auth.uid` с одноразовым идентификатором в качестве значения, для проверки данного ответа. Данный параметр указывается в том случае, если в запросе на аутентификацию указан параметр `openid.auth.check`.

В случае неудачной аутентификации выполняется переход по указанному URL без добавления каких-либо параметров.

Запрос OpenID-провайдера для проверки активной аутентификации**Описание:**

Выполняется проверка аутентификации.

Синтаксис:<https://hostname/ProviderIB/elcib/oid2op?cmd=lookup>**Параметры:***openid.return_to* **обязательный**

Содержит URL, на который выполняется переход после обработки запроса.

openid.auth.check **необязательный**Требуется выполнить проверку ответа на данный запрос (параметр установлен в значение `true`). Имеет смысл только в том случае, если указан параметр `openid.return_to`.**Возвращаемое значение:**Выполняется переадресация на URL, указанный в параметре `openid.return_to`. В случае успешной аутентификации к URL добавляются следующие параметры:

- `openid.auth.user` с именем пользователя в качестве значения;
- `openid.auth.uid` с одноразовым идентификатором в качестве значения, для проверки данного ответа. Данный параметр указывается в том случае, если в запросе на аутентификацию указан параметр `openid.auth.check`.

В случае неудачной аутентификации выполняется переход по указанному URL без добавления каких-либо параметров.

Выполнить проверку ответа OpenID-провайдера**Описание:**Выполняет проверку ответа OpenID-провайдера для запросов `cmd=auth` и `cmd=lookup` в случае, если в запросе передан параметр `openid.auth.check` со значением `true`.**Синтаксис:**<https://hostname/ProviderIB/e1cib/oid2op?cmd=check>**Параметры:***openid.auth.user* **обязательный**

Имя пользователя, которое получено из одноименного параметра запроса.

openid.auth.uid **обязательный**

Значение одноразового идентификатора ответа OpenID-провайдера, полученный из одноименного параметра запроса.

Возвращаемое значение:Возвращается документ типа `text/plain` следующего содержимого:

- `is_valid:true` – ответ действительно сформирован используемым OpenID-провайдером. При этом код состояния HTTP будет равен 200.
- `is_valid:false` – используемый OpenID-провайдер не формировал проверяемый ответ. При этом код состояния HTTP будет равен 400.

Запрос отмены аутентификации для зависимой стороны

Выполняет отмену аутентификации в том случае, если не известен URL OpenID-провайдера. Завершает текущий сеанс, отменяет аутентификацию на OpenID-провайдере, перезапускает веб-клиент. Веб-клиент выполнит запрос отмены аутентификации для OpenID-провайдера.

Синтаксис:

<https://hostname/RPIB/e1cib/oid2op?cmd=logout>

Запрос отмены аутентификации для OpenID-провайдера

Описание:

Выполняет отмену аутентификации на указанном OpenID-провайдере.

Синтаксис:

<https://hostname/ProviderIB/e1cib/oid2op?cmd=logout>

Параметры:

`openid.return_to` *необязательный*

Содержит URL, на который выполняется переход после обработки запроса.

Возвращаемое значение:

Если указан параметр `openid.return_to`, то выполняется переадресация на указанный URL, в противном случае возвращается пустой ответ с кодом состояния HTTP равным 200.

7.9. Особенности настройки веб-серверов

7.9.1. Internet Information Services

Если используется 32-разрядный вариант расширения веб-сервера на 64-разрядной версии операционной системы, то необходимо указать веб-серверу, что он может выполнять 32-разрядные приложения. Для этого нужно выполнить следующие операции:

- Для IIS 5.1, IIS 6.0 – необходимо запустить командный интерпретатор и выполнить в нем следующую команду:

```
cscript %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET W3SVC/AppPools/Enable32bitAppOnWin64 1
```

[Копировать в буфер обмена](#)

- Для IIS 7.0 и старше – открыть диалог основных настроек пула приложения: **Панель управления – Администрирование – Диспетчер служб IIS – <Конкретный сервер> – Пулы приложений – <Нужный пул приложений> – Дополнительные параметры**. Установить параметр **Разрешены 32-разрядные приложения** в значение **Истина (True)**.

При настройке IIS следует помнить, что в рамках одного пула приложений не могут исполняться более одного модуля расширения веб-сервера, которые различаются только третьей и четвертой цифрами версии. Для организации такой работы следует использовать столько пулов приложений, сколько различных версии модулей расширения планируется использовать, и вручную привязать каждое виртуальное приложение веб-клиента к нужному пулу приложений.

В тех случаях, когда ошибки «1С:Предприятия» (при работе с веб-сервером IIS версии 7.0 или 7.5) отображаются текстом вида **500 – внутренняя ошибка сервера. Проблема с запрашиваемым ресурсом; ресурс не может быть отображен**, следует изменить параметр, управляющий представлением ошибок. Для этого необходимо открыть диалог настройки параметров страниц ошибок: **Панель управления – Администрирование – Диспетчер служб IIS – <Конкретный сервер> – сайты – <Default Web Site> – <Имя виртуального приложения> – Страницы ошибок – Изменить параметры...** В открывшемся диалоге следует установить параметр **Если сервер обнаружил ошибку, вернуть** в значение **Подробные сообщения об ошибках**. Затем нажать кнопку **OK**.

При обращении к стандартному интерфейсу OData URL может быть достаточно большой длины. Стандартно IIS ограничивает URL 260 символами. Для изменения этого ограничения необходимо (с правами администратора) в системном реестре, в разделе `HKKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\HTTP\Parameters`, создать параметр `UrlSegmentMaxLength` типа `DWORD`. Данный параметр необходимо установить в значение, большее максимальной длины используемых запросов, или в значение 0, чтобы настроить неограниченную длину URL. Затем следует перезагрузить компьютер, на котором установлен IIS.

7.9.2. Apache

В случае публикации на веб-сервере Apache 2.2 (работающего под управление ОС Windows) файлового варианта информационной базы, рекомендуется дополнять конфигурационный файл веб-сервера Apache (<httpd.conf>) следующим фрагментом:

```
<IfModule mpm_winnt_module>
    ThreadStackSize 8388608
</IfModule>
```

[Копировать в буфер обмена](#)

Если во время работы информационной базы возникают проблемы, связанные с исчерпанием стека на стороне веб-сервер, рекомендуется увеличить значение параметра `ThreadStackSize`.

Подробное описание параметра `ThreadStackSize`: http://httpd.apache.org/docs/2.2/mod/mpm_common.html#ThreadStackSize (на английском языке).

При использовании веб-сервера Apache под управлением ОС Linux, рекомендуется использовать модуль мультипроцессной обработки `worker`. Подробное описание данного модуля: <http://httpd.apache.org/docs/2.2/mod/worker.html> (на английском языке).

7.9.3. Reverse Proxy

Reverse proxy (обратный прокси-сервер) – прокси-сервер, который ретранслирует запросы клиентов из внешней сети на один или несколько серверов, расположенных во внутренней сети. Может использоваться для балансировки нагрузки и повышения безопасности.

Если доступ к веб-серверам, на которых опубликованы информационные базы «1С:Предприятия», осуществляется через **reverse proxy**, то при ненадлежащей настройке **reverse proxy**, это может привести к неработоспособности некоторых механизмов. Это может происходить из-за того, что запрос, который поступает веб-сервер «1С:Предприятия», поступает не от внешнего клиента, а от компьютера, на котором установлен **reverse proxy**.

Для того чтобы избежать этих проблем, следует настроить **reverse proxy** таким образом, чтобы при переадресации HTTP-запроса соответствующим образом формировались заголовки запроса `X-Forwarded-Port`, `X-Forwarded-Host` и `X-Forwarded-Proto`. В этом случае «1С:Предприятие» сможет корректно «восстановить» внешний HTTP-запрос.

Подробное описание настройки обратного прокси-сервера следует искать в документации к веб-серверу, используемого в этой роли.

Глава 8. Настройка веб-браузеров для работы в веб-клиенте

Данная глава описывает настройки веб-браузеров, которые необходимо выполнить для работы в веб-клиенте.

8.1. Общие настройки

Если на компьютере, на котором используется веб-клиент, используется программное обеспечение, блокирующее открытие окон веб-браузеров или отправку HTTP-запросов, то следует включить необходимые веб-сайты (адреса информационных баз) в список исключений.

8.2. Mozilla Firefox

8.2.1. Настройки подключения

Для начала работы необходимо выполнить следующие настройки:

- Запустить веб-браузер;
- Открыть окно настроек: [Firefox – Настройки – Настройки](#);
- В открывшемся окне перейти в раздел [Содержимое](#) и выполнить следующие настройки:
 - Установить флажок [Использовать JavaScript](#);
 - Установить флажок [Автоматически загружать изображения](#). При этом следует убедиться, что необходимые веб-сайты (адреса информационных баз) не находятся в списке исключений в состоянии [Блокировать](#).
- Перейти в раздел [Приватность](#) и выполнить следующие настройки:
 - В группе [История](#) установить параметр [Firefox](#): в значение [будет запоминать историю](#) или [будет использовать ваши настройки хранения истории](#).
 - Если выбрано значение [будет использовать ваши настройки хранения истории](#), то следует установить флажок [Принимать cookies с сайтов](#). При этом следует убедиться, что необходимые веб-сайты (адреса информационных баз) не находятся в списке исключений в состоянии [Блокировать](#).

В случае если в качестве веб-браузера используется не локализованная версия Mozilla Firefox, то для правильной работы веб-клиента требуется выполнить следующее (наименования пунктов меню приводятся по английской версии):

- Запустить веб-браузер;
- Открыть окно настроек: [Firefox – Options – Options](#);
- В разделе [Contents](#) в группе [Fonts & Colors](#) нажмите кнопку [Advanced](#);
- В параметре [Default Character Encoding](#) выберите [Unicode](#).

Рекомендуется включить вопрос местоположения сохраняемого файла. Для этого необходимо:

- Запустить веб-браузер;
- Открыть окно настроек: [Firefox – Настройки – Настройки](#).
- В разделе [Основные](#) установить переключатель в значение [Всегда выдавать запрос на сохранение файла](#).

Для работы системы в веб-браузере Mozilla Firefox рекомендуется включить расширение браузера [1C:Enterprise Extension](#), установка которого предлагается при первой попытке доступа к информационной базе «1С:Предприятия», а в дальнейшем – помощью команды [Сервис – Настройки веб-браузера](#).

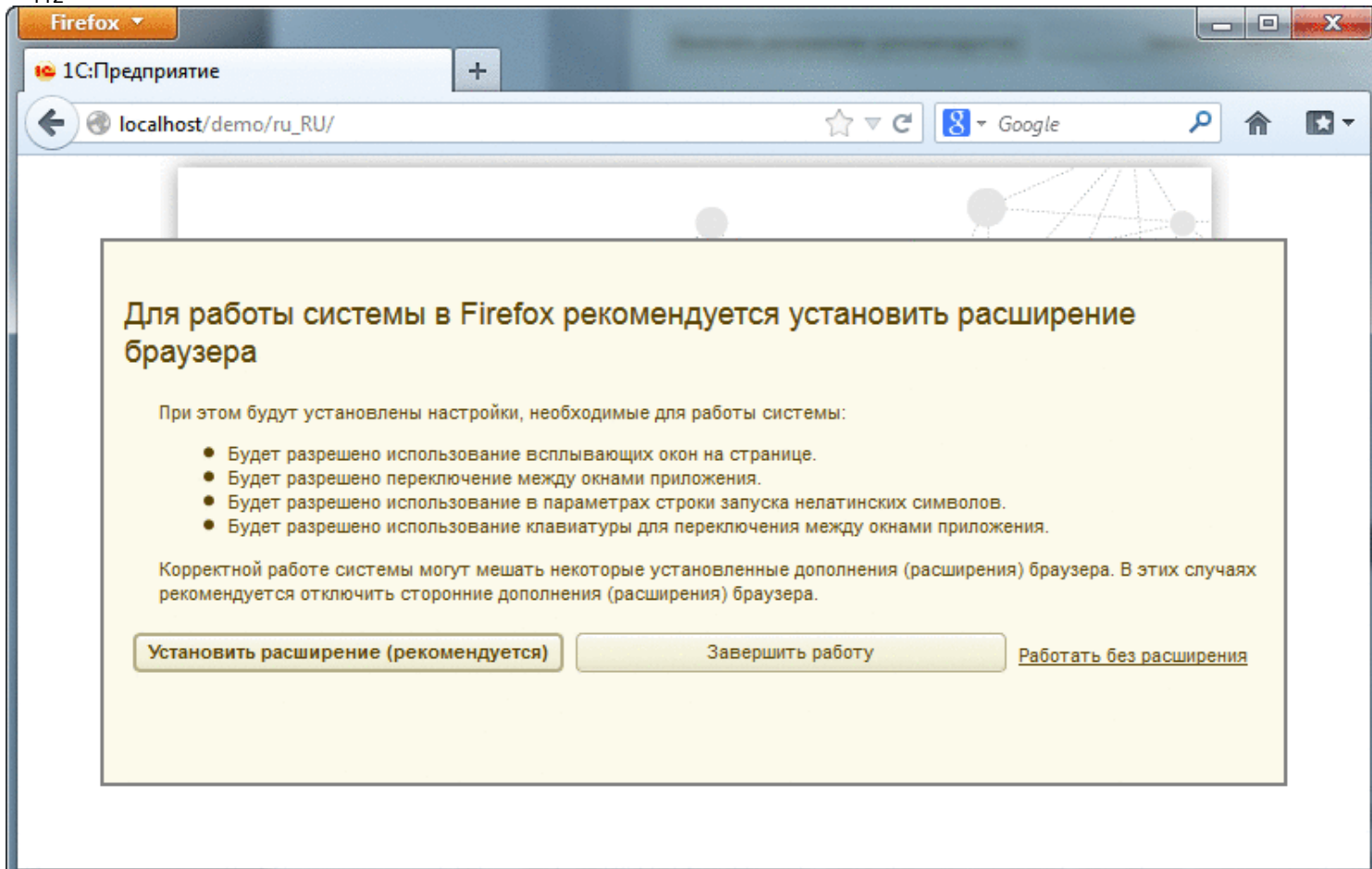


Рис. 76. Запрос установки расширения

При этом будут установлены настройки, необходимые для работы системы:

- Будет разрешено использование всплывающих окон на странице;
- Будет разрешено переключение между окнами приложения;
- Будет разрешено использование в параметрах строки запуска нелатинских символов;
- Будет разрешено использование клавиатуры для переключения между окнами приложения;
- Будет возможно использовать расширения работы с файлами, расширения работы с криптографией, внешних компонент. Установки расширений и внешних компонент будет доступна только при установленном расширении веб-браузера [1C:Enterprise Extension](#).

В случае работы с отключенным расширением, некоторые настройки можно выполнить вручную:

- Запустить веб-браузер;
- Разрешение всплывающих окон:
 - Открыть окно настроек: [Firefox – Настройки – Настройки](#);
 - в открывшемся окне перейдите в раздел [Содержимое](#);
 - снимите флажок [Блокировать всплывающие окна](#).
- Разрешение переключения между окнами приложения с помощью клавиатуры:
 - в адресной строке браузера набрать [about:config](#);
 - найти параметр [dom.disable_window_flip](#);
 - установить параметра в значение [false](#).

- найти параметр [dom.popup_allowed_events](#);
- добавьте к значению этой настройки событие [keydown](#) (через пробел от остальных параметров).
- Разрешение использования нелатинских символов в строке запуска:
 - в адресной строке браузера набрать [about:config](#);
 - найти параметр [network.standard-url.encode-query-utf8](#) или [browser.fixup.use-utf8](#), если параметр не найден [network.standard-url.encode-query-utf8](#);
 - установите параметр в значение [true](#).

8.2.2. Автоматическая аутентификация

Для браузера Mozilla Firefox имеется возможность сконфигурировать веб-браузер для использования автоматической аутентификации ОС.

Также эти настройки можно выполнить вручную:

- Запустить веб-браузер.
- В адресной строке браузера набрать [about:config](#).
- Затем на странице настроек в строке поиска ввести название параметра.

Данная настройка может осуществляться для трех параметров:

- [network.automatic-ntlm-auth.trusted-uris](#).
- В конкретной конфигурации сети и веб-сервера, возможно, потребуется задание значений для параметров [network.negotiate-auth.trusted-uris](#) и [network.negotiate-auth.delegation-uris](#).
- Далее задать список веб-серверов, через которые будет осуществляться работа с базой «1С:Предприятия».

Подробнее об этой возможности можно прочитать в статье:

https://developer.mozilla.org/En/Integrated_Authentication (на английском языке).

Далее приведено описание, за что отвечают вышеприведенные параметры при разных способах аутентификации:

- Веб-сервер поддерживает **NTLM**-аутентификацию.

Если имя веб-сервера, к которому осуществляется попытка доступа, перечислено в списке имен, содержащихся в параметре [network.automatic-ntlm-auth.trusted-uris](#), то будет осуществлена попытка автоматической аутентификации. Если же имени веб-сервера там нет, то браузер покажет диалог, в котором необходимо указать логин и пароль пользователя для доступа к веб-серверу.

- Веб-сервер поддерживает **Kerberos**-аутентификацию.

Для того чтобы получить доступ к веб-серверу с данным типом аутентификации, в параметр [network.negotiate-auth.trusted-uris](#) нужно добавить имя этого веб-сервера. При работе с файловой информационной базой этого будет достаточно. В случае необходимости обеспечить автоматическую аутентификацию пользователей веб-клиента при использовании клиент-серверного варианта «1С:Предприятия» необходимо добавить DNS-имя этого веб-сервера в параметр [network.negotiate-auth.delegation-uris](#).

Если имя веб-сервера, к которому осуществляется доступ, не обнаружено в параметре [network.negotiate-auth.trusted-uris](#), аутентификация производиться не будет, и пользователь увидит сообщение об ошибке **401 Unauthorized**. Для информирования пользователя о действиях, которые ему необходимо предпринять, администратор может модифицировать страницу сообщения об ошибке **401** (см. документацию к используемому веб-серверу).

8.3. Microsoft Internet Explorer

Для работы с веб-клиентом в операционной системе Windows XP SP 2 требуется наличие системной библиотеки [jscript.dll](#) версии 5.6.0.8834 или выше. Библиотека находится в каталоге [system32](#) операционной системы. Если

на вашем компьютере установлена более ранняя версия библиотеки, то возможно значительное замедление работы веб-клиента. Для обновления версии необходимо скачать пакет обновлений, находящийся по адресу <http://support.microsoft.com/kb/942840/en-us>.

ПРИМЕЧАНИЕ. При первом запуске веб-клиента под Microsoft Internet Explorer 6.0 выдается предупреждение о том, что работа в этом веб-браузере может быть неэффективной. Рекомендуется использовать Microsoft Internet Explorer 7 и выше или другой веб-браузер (см. [здесь](#)).

8.3.1. Настройка подключения

Для начала работы необходимо выполнить следующие настройки:

- Запустить веб-браузер.
- В меню [Сервис](#) выбрать пункт [Свойства обозревателя](#).
- В открывшемся окне перейти на закладку [Безопасность](#).
- Нажать кнопку [Другой](#).
- В открывшемся окне:
 - В разделе [Сценарии](#) для параметра [Активные сценарии](#) установить [Разрешить](#) или [Включить](#).
 - В разделе [Прочее](#) для параметра [Отображение разнородного содержимого](#) установить значение [Включить](#) (только для Microsoft Internet Explorer версий 6.0, 7 и 8).
 - В разделе [Загрузка](#) для параметра [Автоматические запросы на загрузку файлов](#) установить значение [Включить](#) (только для Microsoft Internet Explorer версий 6.0, 7 и 8).
 - Нажать [ОК](#).
- Перейти на закладку [Конфиденциальность](#) и с помощью регулятора выбрать уровень использования cookies [Умеренный](#) или [Средний](#) (рекомендуется не выше [Умеренно высокий](#)).
- На этой же закладке, в разделе [Блокирование всплывающих окон](#) снять флажок [Блокировать всплывающие окна](#) или [Включить блокирование всплывающих окон](#).
- Перейти на закладку [Дополнительно](#) и установить параметр [Отображать рисунки](#) или [Показывать изображения](#).

8.3.2. Использование внешних компонентов, расширения работы с файлами, расширения работы с криптографией

Если предполагается использование внешних компонентов (сканера штрихкодов, электронных весов и т. д.), расширения для работы с файлами или расширения работы с криптографией, на закладке [Безопасность](#) необходимо выбрать подходящую зону: [Надежные узлы](#) или [Местная интрасеть](#), а затем выполнить следующие действия:

- На вкладке [Безопасность](#) нажать кнопку [Другой](#).
- В открывшемся окне установить [Разрешить](#) или [Включить](#) для следующих параметров:
 - [Автоматические запросы элементов управления ActiveX](#);
 - [Работа с ActiveX объектам и плагинами](#) или [Запуск элементов ActiveX и модулей подключения](#);
 - [Работа с объектами, помеченными как безопасные](#) или [Выполнять сценарии элементов ActiveX, помеченные как безопасные](#).
- В этом же окне ([Параметры безопасности](#)) установить [Предлагать](#) для следующего параметра:
 - [Загрузка подписанных ActiveX объектов с разрешения пользователя](#) или [Загрузка подписанных элементов ActiveX](#).

Настройку использования ActiveX также можно выполнить с помощью службы установки ActiveX (для ОС

Windows Vista, Windows 7, Windows Server 2008, Windows Server 2008 R2).

Подробное описание настройки можно получить:

- для Windows Vista, Windows Server 2008:
[http://technet.microsoft.com/ru-ru/library/cc721964\(WS.10\).aspx](http://technet.microsoft.com/ru-ru/library/cc721964(WS.10).aspx);
- для Windows 7, Windows Server 2008 R2:
[http://technet.microsoft.com/ru-ru/library/dd631688\(WS.10\).aspx](http://technet.microsoft.com/ru-ru/library/dd631688(WS.10).aspx).

8.4. Google Chrome

Для начала работы необходимо выполнить следующие настройки:

- Запустить веб-браузер.
- В меню выбрать пункт **Параметры**.
- На вкладке **Расширенные** нажать кнопку **Настройки содержания**.
- На вкладке **Файлы cookie** установить флажок **Разрешить сохранять локальные данные (рекомендуется)** или добавить адреса информационных баз в список исключений.
- На вкладке **Безопасность JavaScript** установить флажок **Разрешить всем сайтам использовать JavaScript (рекомендуется)** или добавить адреса информационных баз в список исключений.
- На вкладке **Всплывающие окна** установить флажок **Разрешить всплывающие окна для всех сайтов** или добавить адреса информационных баз в список исключений.

Рекомендуется включить вопрос местоположения сохраняемого файла. Для этого необходимо:

- Открыть веб-браузер.
- В меню выбрать пункт **Параметры**.
- На закладке **Расширенные**, в разделе **Загрузка** установить флажок **Запрашивать место для сохранения каждого файла перед загрузкой**.

8.5. Safari

Для начала работы необходимо выполнить следующие настройки:

- Запустить веб-браузер.
- В меню выбрать пункт **Настройки**.
- На вкладке **Безопасность** установить флажок **Подключить JavaScript**.
- На вкладке **Безопасность**, в группе **Принимать Cookies** установить флажок **Только от посещаемых сайтов**.
- На вкладке **Безопасность** снять флажок **Блокировать всплывающие окна**.
- На вкладке **Вкладки** снять флажок **Ctrl-нажатие мыши открывает ссылку в новой вкладке**.

Рекомендуется включить вопрос местоположения сохраняемого файла. Для этого необходимо:

- Открыть веб-браузер.
- В меню выбрать пункт **Параметры**.
- На закладке **Основные** установить флажок **Всегда выполнять запрос перед загрузкой**.

ПРИМЕЧАНИЕ. Настройка вопроса о местоположении сохраняемого файла недоступна в веб-браузере Safari для ОС Mac OS.

8.6. Особенности использования веб-клиента

Особенности поведения веб-клиента описаны в книге «1С:Предприятие 8.3. Руководство разработчика».

Глава 9. Защита от несанкционированного использования: особенности и настройка

Защита от несанкционированного использования системы «1С:Предприятие» может быть построена на использовании сетевой системы защиты HASP4 Net или системы программного лицензирования (далее СПЛ). Любая из этих систем обеспечивает одновременную работу определенного количества пользователей (сеансов) с системой «1С:Предприятие». При этом пользователи могут находиться как в рамках локальной сети, так и за ее пределами (при использовании веб-клиентов или тонких клиентов, подключенных через веб-сервер).

Выбор той или иной системы защиты определяется поставкой программного продукта.

Далее будет подробно рассмотрена каждая схема защиты.

9.1. Система защиты HASP4

Для защиты от несанкционированного использования системы «1С:Предприятие» может использоваться сетевая система защиты HASP4 Net. С помощью данной системы защиты организовано обеспечение одновременной работы определенного количества пользователей (сеансов) с системой «1С:Предприятие». При этом пользователи могут находиться как в рамках локальной сети, так и за ее пределами – веб-клиенты и тонкие клиенты, подключенные к информационной базе с помощью веб-сервера. При этом подсчет пользователей производится либо специальной программой – HASP License Manager, либо серверной частью «1С:Предприятия».

ВНИМАНИЕ! Взаимодействие с HASP License Manager возможно только по протоколу IPv4.

При любом способе подсчета пользователей в сети должен находиться один или несколько компьютеров, к USB-портам которых подключены клиентские аппаратные ключи защиты HASP4 Net. Общее число пользователей, которые могут работать с системой, определяется как сумма лицензий, доступных в каждом из подключенных клиентских ключей (с учетом некоторых особенностей, описание которых см. [здесь](#)).

ВНИМАНИЕ! К USB-портам одного компьютера не имеет смысла присоединять несколько аппаратных ключей HASP4 Net одной серии, предназначенных для защиты «1С:Предприятия», так как эти ключи неразличимы и фактически будет задействован только один из них (выбранный произвольно).

HASP License Manager может быть запущен как обычное приложение Windows, как служба (только в операционных системах Windows 2000/XP/Server 2003/Vista/Server 2008), а также в операционной системе Linux. При использовании многопользовательской системы защиты нет необходимости устанавливать HASP Device Driver на пользовательских компьютерах, на которых запускается «1С:Предприятие» и к USB-порту которых не присоединен клиентский ключ защиты «1С:Предприятия».

СОВЕТ. Загрузить последние версии HASP Device Driver и HASP License Manager можно с сайта компании Aladdin по адресу: <http://www3.safenet-inc.com/support/hasp.aspx>.

9.1.1. Маркировка ключей защиты

Назначение ключа защиты	Маркировка
Локальный клиентский ключ	ORGL8
Многопользовательский клиентский ключ (количество пользователей указано после «NET»: 5, 10, 20, 50, 100)	NET5 ORGL8
	NET10 ORGL8
	NET20 ORGL8
	NET50 ORGL8
	NET100 ORGL8

Многопользовательский клиентский ключ на 300 пользователей	NET250+ ORG8A
Многопользовательский клиентский ключ на 500 пользователей	NET250+ ORG8B
Локальный ключ 32-разрядного сервера	ENSR8
Локальный ключ 64-разрядного сервера	EN8SA

Ключ аппаратной защиты 32-разрядного сервера обеспечивает возможность работы произвольного количества 32-разрядных рабочих процессов на одном физическом компьютере.

Ключ аппаратной защиты 64-разрядного сервера обеспечивает возможность работы произвольного количества 32-разрядных и 64-разрядных рабочих процессов на одном физическом компьютере.

На одном компьютере могут одновременно работать по одному ключу серий ORGL8, ORGL8A и ORGL8B.

Лицензии ищутся в следующем порядке:

- в ключе ORGL8,
- в ключе ORG8A,
- в ключе ORG8B.

Также реализовано запоминание ключа защиты, чья лицензия была получена пользователем при последнем подключении. Именно из этого ключа будет выполняться попытка получения лицензии при следующем обращении. Если из запомненного ключа лицензию получить не удалось, порядок поиска свободной лицензии соответствует описанному выше.

9.1.2. Защита от несанкционированного использования

Для предотвращения возможности незаконного использования система «1С:Предприятие» предоставляется пользователям в защищенном виде.

Возможность использования программного продукта на одном или нескольких рабочих местах, а также возможность использования сервера «1С:Предприятия» определяются имеющимися лицензионными соглашениями.

Одной из составных частей используемой системы защиты является аппаратный ключ защиты от несанкционированного использования.

Для работы продукта, использование которого регулируется Лицензионным соглашением на одно рабочее место или на одно дополнительное рабочее место, к USB-порту компьютера необходимо присоединить аппаратный ключ (подробности установки драйвера защиты (см. [здесь](#)). Если использование продукта регулируется дополнительной многопользовательской лицензией, необходимо присоединить аппаратный ключ защиты к USB-порту компьютера, на котором запущена программа HASP License Manager (подробности установки и использования см. [здесь](#)).

Информация о последних изменениях в системе защиты размещается в файле [readme.htm](#).

9.1.3. Особенности учета клиентских лицензий

В зависимости от вида клиента и места расположения ключа с клиентскими лицензиями (локального или сетевого) возможны несколько вариантов учета лицензий. Рассмотрим их более подробно.

9.1.3.1. Файловый вариант

В данном случае возможны следующие варианты получения лицензий.

9.1.3.1.1. Локальный ключ

Обеспечивает запуск на компьютере с ключом произвольного количества экземпляров системы в режиме 1С:Предприятие или Конфигуратор.

9.1.3.1.2. Многопользовательский клиентский ключ, доступный по сети через HASP License Manager

Обеспечивает одновременную работу столько компьютеров, на сколько пользователей имеется ключ. На одном компьютере возможен запуск произвольного количества экземпляров системы в режиме 1С:Предприятие или Конфигуратор.

Количество лицензий ограничено общим количеством доступных лицензий со всех компьютеров в сети, на которых установлен и настроен HASP License Manager.

9.1.3.2. Клиент-серверный вариант

В данном случае возможны следующие варианты получения лицензий.

9.1.3.2.1. Локальный ключ

Обеспечивает запуск на компьютере с ключом произвольного количества экземпляров системы в режиме 1С:Предприятие или Конфигуратор.

9.1.3.2.2. Многопользовательский клиентский ключ, доступный по сети через HASP License Manager

Обеспечивает одновременную работу столько компьютеров, на сколько пользователей имеется ключ. На одном компьютере возможен запуск произвольного количества экземпляров системы в режиме 1С:Предприятие или Конфигуратор.

Количество лицензий ограничено общим количеством доступных лицензий со всех компьютеров в сети, на которых установлен и настроен HASP License Manager.

9.1.3.2.3. Локальный многопользовательский клиентский ключ без установленного HASP License Manager и многопользовательский клиентский ключ, доступный по сети через HASP License Manager

В этом случае ключ может располагаться как на компьютере, где установлен сервер «1С:Предприятия» (локальный многопользовательский клиентский ключ), так и в сети. Подсчет лицензий занимается менеджер кластера, на который назначен сервис сеансовых данных. В этом случае лицензии расходуются из расчета «один сеанс – одна лицензия». Таким образом, если на одном компьютере запущено два экземпляра «1С:Предприятия» (в любом режиме запуска и с любым видом клиента), то на это будет израсходовано две лицензии.

ВНИМАНИЕ! Если подсчетом клиентских лицензий занимается сервер «1С:Предприятия», то клиентские лицензии, у которых в программе Aladdin Monitor (скачать можно по адресу <http://www3.safenet-inc.com/support/hasp/hasp4/vendor.aspx#testtools>) значение колонки Timeout равно 0, будут считаться занятыми. В связи с этим **не рекомендуется** одни и те же многопользовательские ключи HASP использовать для **одновременного** получения клиентских лицензий с помощью HASP License Manager и сервера «1С:Предприятия».

Необходимо учитывать следующую особенность: если в сети будет обнаружено несколько многопользовательских клиентских ключей серии ORGL8, то сервером будет выбран **один** произвольный ключ. После исчерпания лицензий этого ключа возможно использование **одного** многопользовательского ключа ORG8A, и затем возможно использование **одного** многопользовательского ключа ORG8B. При выборе клиентским приложением того же сетевого клиентского ключа, который выбран сервером, клиентское приложение также может прекратить поиск лицензии в других доступных по сети ключах той же серии.

9.1.3.3. Веб-клиент

В зависимости от варианта информационной базы (файловый или клиент-серверный) подсчетом лицензий занимается либо модуль расширения веб-сервера (в файловом варианте), либо сервер «1С:Предприятия» (в клиент-серверном варианте).

В этом случае ключ может располагаться как на компьютере, где установлен модуль расширения веб-сервера (или сервер «1С:Предприятия»), так и в сети. Подсчетом лицензий занимается непосредственно модуль расширения веб-сервера (или сервер «1С:Предприятия»). В этом случае лицензии расходуются из расчета «один сеанс – одна лицензия». Таким образом, если на одном компьютере открыто два окна веб-браузера с доступом к одной информационной базе, то на это будет израсходовано две лицензии.

ВНИМАНИЕ! Если подсчетом клиентских лицензий занимается сервер «1С:Предприятия» или расширение веб-сервера, то клиентские лицензии, у которых в программе Aladdin Monitor (скачать можно по адресу <http://www3.safenet-inc.com/support/hasp/hasp4/vendor.aspx#testtools>) значение колонки **Timeout** равно **0**, будут считаться занятыми. В связи с этим **не рекомендуется** использовать одни и те же многопользовательские ключи HASP для **одновременного** получения клиентских лицензий с помощью HASP License Manager и сервера (или расширения веб-сервера) «1С:Предприятия».

Необходимо учитывать следующую особенность: если в сети будет обнаружено несколько многопользовательских клиентских ключей серии ORGL8, то сервером будет выбран **один** произвольный ключ. После исчерпания лицензий этого ключа возможно использование **одного** многопользовательского ключа ORG8A, и затем возможно использование **одного** многопользовательского ключа ORG8B. При выборе клиентским приложением того же сетевого клиентского ключа, который выбран сервером, клиентское приложение также может прекратить поиск лицензии в других доступных по сети ключах той же серии.

9.1.3.4. Тонкий клиент, работающий через веб-сервер

Тонкий клиент может использовать для получения лицензий:

- локальный ключ;
- многопользовательский ключ, доступный тонкому клиенту по сети через HASP License Manager;
- модуль расширения веб-сервера или сервер «1С:Предприятия».

Если лицензия получается непосредственно тонким клиентом, то на одном компьютере возможен запуск произвольного количества экземпляров системы в режиме 1С:Предприятие.

Лицензия также может выдаваться модулем расширения веб-сервера (в случае файлового варианта) или сервером «1С:Предприятия» (в случае клиент-серверного варианта). При этом подсчетом лицензий занимается непосредственно модуль расширения веб-сервера или сервер «1С:Предприятия». В этом случае лицензии расходуются из расчета «один сеанс – одна лицензия». Таким образом, если на одном компьютере запущено два экземпляра «1С:Предприятия», то на это будет израсходовано две лицензии. В этом случае ключ может располагаться как на компьютере, где установлен модуль расширения веб-сервера (или сервер «1С:Предприятия»), так и в сети.

ВНИМАНИЕ! Если подсчетом клиентских лицензий занимается сервер «1С:Предприятия» или расширение веб-сервера, то клиентские лицензии, у которых в программе Aladdin Monitor (скачать можно по адресу <http://www3.safenet-inc.com/support/hasp/hasp4/vendor.aspx#testtools>) значение колонки **Timeout** равно **0**, будут считаться занятыми. В связи с этим **не рекомендуется** использовать одни и те же многопользовательские ключи HASP для **одновременного** получения клиентских лицензий с помощью HASP License Manager и сервера (или расширения веб-сервера) «1С:Предприятия».

Необходимо учитывать следующую особенность подсчета лицензий с помощью модуля расширения веб-сервера или сервера «1С:Предприятия»: если в сети будет обнаружено несколько многопользовательских клиентских ключей серии ORGL8, то сервером будет выбран **один** произвольный ключ. После исчерпания лицензий этого ключа возможно использование **одного** многопользовательского ключа ORG8A, и затем возможно использование **одного** многопользовательского ключа ORG8B. При выборе клиентским приложением того же сетевого клиентского ключа, который выбран сервером, клиентское приложение также может прекратить поиск лицензии в других доступных по сети ключах той же серии.

9.1.3.5. Локальный ключ и веб-клиент

В случае, если на компьютере с сервером «1С:Предприятия» или веб-сервером (в случае файлового

варианта информационной базы) установлен локальный ключ, то имеется возможность запустить:

- произвольное количество конфигураторов на компьютере с ключом;
- произвольное количество клиентских приложений (кроме веб-клиента) на других компьютерах, если им доступны клиентские лицензии;
- в случае файлового варианта информационной базы:
 - одно произвольное клиентское приложение (включая веб-клиента) на любом компьютере, если ему недоступен клиентский ключ;
 - произвольное количество клиентских приложений (кроме веб-клиента) на компьютере с ключом.
- в случае клиент-серверного варианта информационной базы:
 - одно произвольное клиентское приложение (включая веб-клиента) на любом компьютере, если ему недоступен клиентский ключ;
 - произвольное количество клиентских приложений (кроме веб-клиента) на компьютере с ключом.

Другими словами, имеется возможность вести разработку и отладку с использованием веб-клиента, используя только локальный ключ.

ПРИМЕЧАНИЕ. При использовании локального ключа может быть запущен только один веб-клиент.

9.1.3.6. COM-соединение

При использовании 32-разрядного COM-соединения поиск доступной лицензии осуществляется в следующем порядке:

- локальные клиентские лицензии;
- локальные серверные лицензии (как 32-разрядные, так и 64-разрядные);
- сетевые клиентские лицензии;
- клиентские лицензии на сервере «1С:Предприятия» (в клиент-серверном варианте) или веб-сервере (в случае файлового варианта, подключенного через веб-сервер).

При использовании 64-разрядного COM-соединения поиск доступной лицензии осуществляется в следующем порядке:

- локальные клиентские лицензии;
- локальная серверная лицензия (**только** 64-разрядные);
- сетевые клиентские лицензии;
- клиентские лицензии на сервере «1С:Предприятия» (в клиент-серверном варианте) или веб-сервере (в случае файлового варианта, подключенного через веб-сервер).

9.1.3.7. Web-сервис

Для работы Web-сервисов не требуется клиентских лицензий. Однако если информационная база, которая предоставляет Web-сервисы, работает в клиент-серверном варианте, необходимо наличие серверной лицензии для работы сервера «1С:Предприятия».

9.1.3.8. Терминальный сервер

В данном случае возможны следующие варианты получения лицензий.

9.1.3.8.1. Локальный ключ

Позволяет работать только одному пользователю, который подключился к терминальной сессии с идентификатором 0. Обеспечивает запуск произвольного количества экземпляров системы в режиме 1С:Предприятие или Конфигуратор. Для терминальных сессий с отличным от 0 идентификатором сессии локальные ключи недоступны.

9.1.3.8.2. Многопользовательский клиентский ключ

Лицензии с многопользовательского клиентского ключа, установленного в терминальный сервер, доступны для использования только в том случае, если на терминальный сервер установлен и настроен HASP License Manager. Клиентские лицензии учитываются аналогично многопользовательскому клиентскому ключу, доступному по сети через HASP License Manager. При этом одним рабочим местом считается одна терминальная сессия.

9.1.4. Особенности получения серверной лицензии

Ключ аппаратной защиты должен быть установлен на компьютере, на котором выполняется (один или несколько) рабочий процесс (**rphost**) кластера серверов. Рабочие процессы могут принадлежать разным кластерам серверов. Проверка наличия серверной лицензии осуществляется в тот момент, когда клиентское приложение выполняет подключение к рабочему процессу.

Ключ аппаратной защиты сервера является локальным и недоступен по сети.

9.1.5. Установка драйвера защиты

9.1.5.1. Для ОС Windows

Программа установки HASP Device Driver (**haspdinst.exe**) входит в комплект поставки и устанавливается на компьютер при установке кластера серверов «1С:Предприятия».

Для установки HASP Device Driver нужно выбрать пункт меню **Пуск – Программы – 1С Предприятие 8 – Дополнительно – Установка HASP Device Driver**.

Можно также установить HASP Device Driver «вручную». Для этого из командной строки следует запустить программу **haspdinst.exe**, размещенную в каталоге **\Program Files\1cv8\common**, с ключом **-i**. Таким образом, командная строка для установки HASP Device Driver имеет следующий вид:

[Копировать в буфер обмена](#)

```
haspdinst -i
```

СОВЕТ. Рекомендуется сначала произвести установку HASP Device Driver, а затем присоединить ключ к USB-порту.

ВНИМАНИЕ! Отсоединение аппаратного ключа защиты от USB-порта во время работы не допускается!

В случае ненадобности HASP Device Driver может быть удален из системы. Для удаления HASP Device Driver нужно выбрать пункт меню **Пуск – Программы – 1С Предприятие 8 – Дополнительно – Удаление HASP Device Driver**.

Для удаления HASP Device Driver можно также воспользоваться командной строкой следующего вида:

[Копировать в буфер обмена](#)

```
haspdinst -r
```

9.1.5.2. Для ОС Linux

Драйвер защиты можно загрузить с сайта компании Aladdin (<http://www3.safenet-inc.com/support/hasp/enduser.aspx>).

Для установки HASP License Manager необходимо выполнить следующие действия (действия должны выполняться от имени администратора):

- Распаковать архив, воспользовавшись следующей командой:

[Копировать в буфер обмена](#)

```
tar xzf HASP_SRM_LINUX_3.50_Run-time_Installer_script.tar.gz
```

- Перейти в каталог с распакованным драйвером:

[Копировать в буфер обмена](#)

```
cd HASP_SRM_LINUX_3.50_Run-time_Installer_script
```

- Установить драйвер (точка в конце приведенной командной строки является обязательной):

[Копировать в буфер обмена](#)

```
./dinst .
```

СОВЕТ. Рекомендуется сначала произвести установку HASP Device Driver, а затем присоединить ключ к USB-порту.

ВНИМАНИЕ! Отсоединение аппаратного ключа защиты от USB-порта во время работы не допускается!

Для удаления драйвера ключа следует перейти в каталог с распакованным драйвером и выполнить там команду:

[Копировать в буфер обмена](#)

```
./dunst
```

9.1.6. Установка HASP License Manager

9.1.6.1. Для ОС Windows

В комплект поставки системы «1С:Предприятие» входит утилита [lmsetup.exe](#), используемая для установки HASP License Manager. Утилита находится на установочном диске системы «1С:Предприятие» и может быть запущена как непосредственно из командной строки, так и через меню программы установки системы «1С:Предприятие».

HASP License Manager может быть установлен на любом компьютере в локальной сети, работающем под управлением операционных систем Microsoft Windows. При этом в любой из этих систем HASP License Manager может быть установлен как обычное приложение, а в среде операционных систем Windows 2000/XP/Vista/7/Server 2003/Server 2008 еще и как служба Windows.

ВНИМАНИЕ! Возможна нестабильная работа менеджера лицензий в случае его установки на компьютер, используемый в качестве терминального сервера.

Установка менеджера лицензий на компьютер, который используется в качестве терминального сервера не рекомендуется.

Для установки HASP License Manager следует запустить программу установки [lmsetup.exe](#) (ниже рассматривается пример установки HASP License Manager версии 8.32).

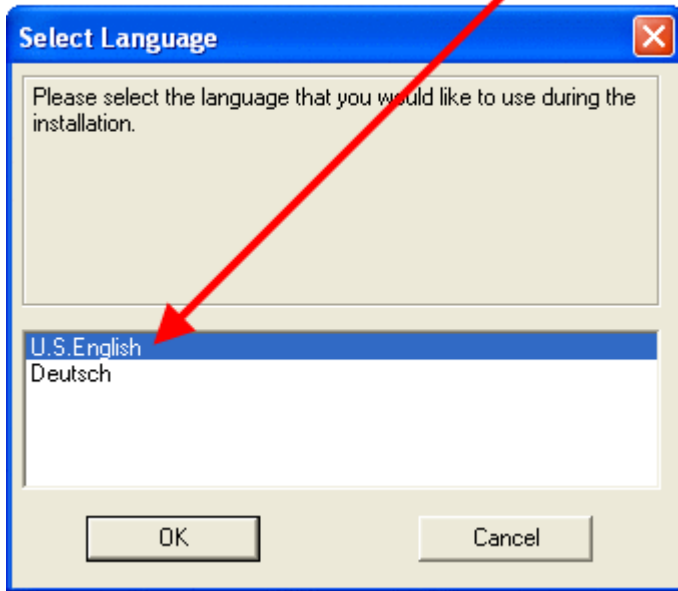


Рис. 77. Выбор языка

Затем выбрать английский язык для программы установки (см. [рис. 77](#)).

Далее нужно подтвердить, что вы согласны с предлагаемой лицензией.

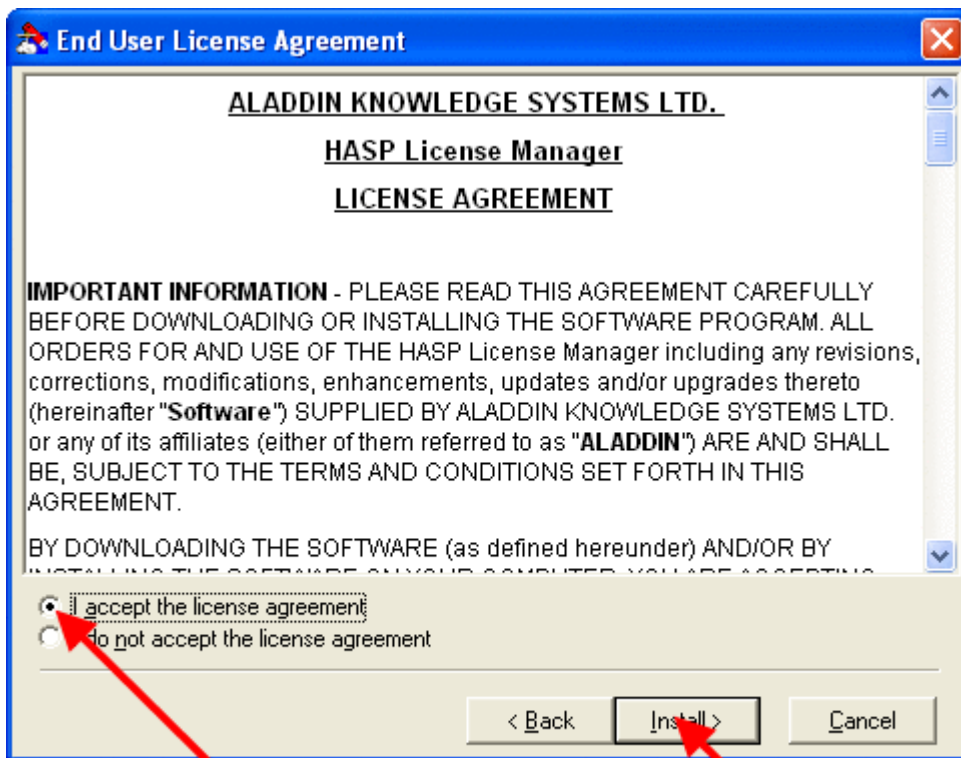


Рис. 78. Согласие с лицензией

В случае установки HASP License Manager на компьютере, работающем под управлением операционных систем Windows 2000/XP/Vista/7/Server 2003/Server 2008, будет предложено два варианта установки HASP License Manager – как приложение ([Application](#)) или как служба ([Service](#)). В случае установки HASP License Manager на компьютере, работающем под управлением операционных систем Windows 98/Me, этот диалог будет пропущен, так как в этих операционных системах может быть установлено только приложение.

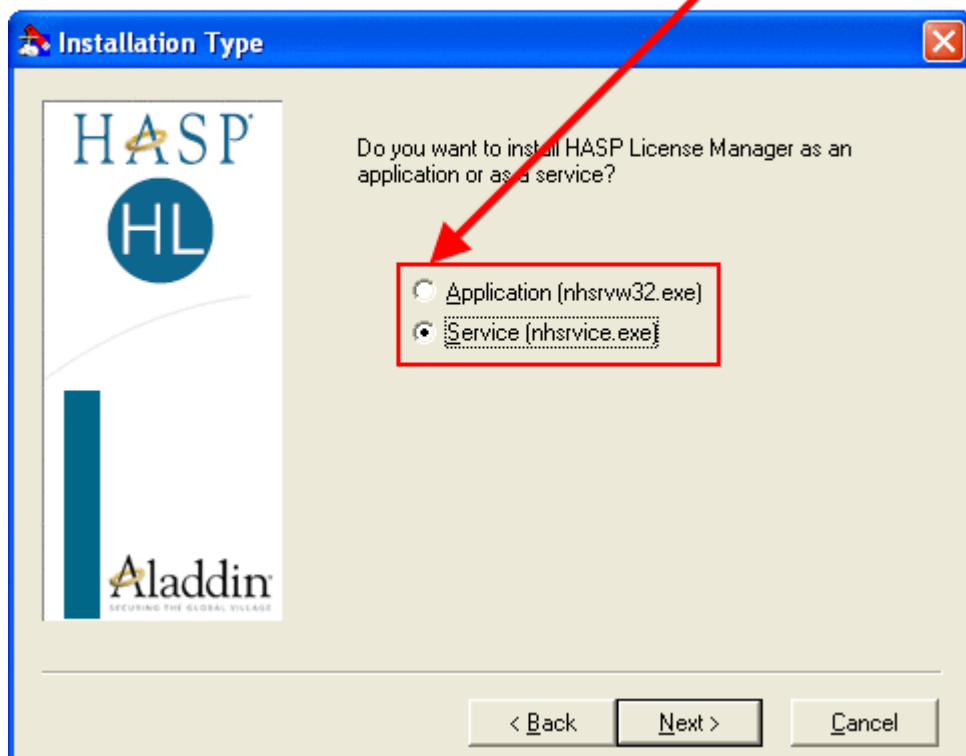


Рис. 79. Выбор режима установки

Далее будет предложено выбрать каталог, где будут размещены исполняемые файлы HASP License Manager и файл помощи. В случае установки HASP License Manager как службы Windows исполняемые файлы будут размещены в системном каталоге Windows, а в выбранном каталоге будет установлен только файл помощи.

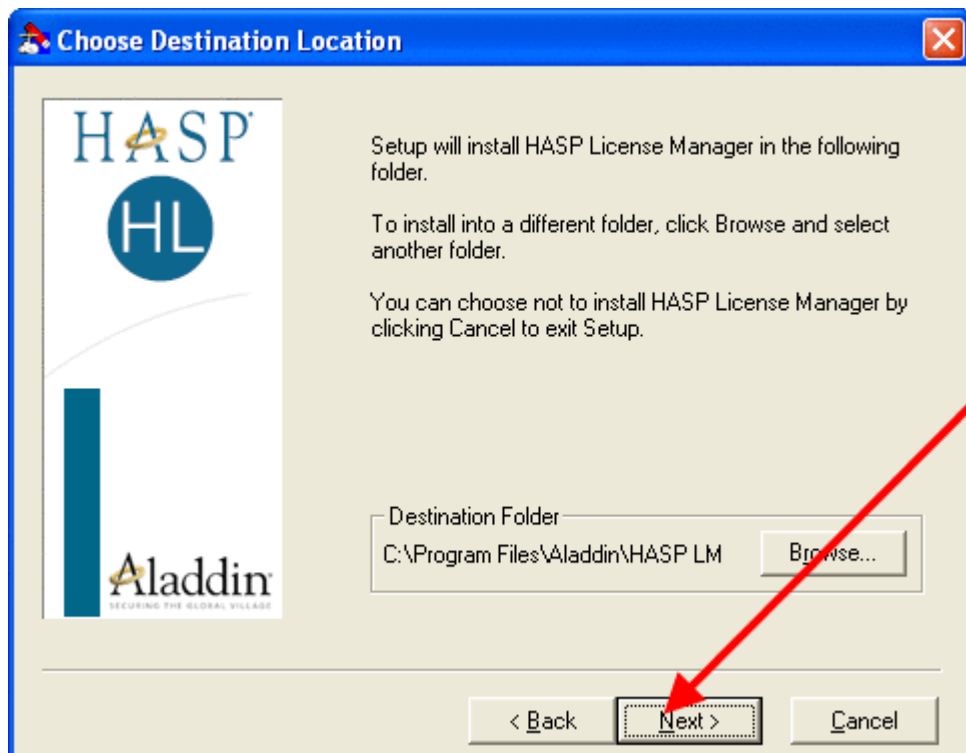


Рис. 80. Выбор пути установки HASP License Manager

На следующем этапе установки предлагается выбрать группу, в которой будут размещены ярлыки запуска HASP License Manager и файла помощи. По умолчанию создается новая группа с названием HASP License Manager, но можно выбрать уже существующую группу или изменить название создаваемой.

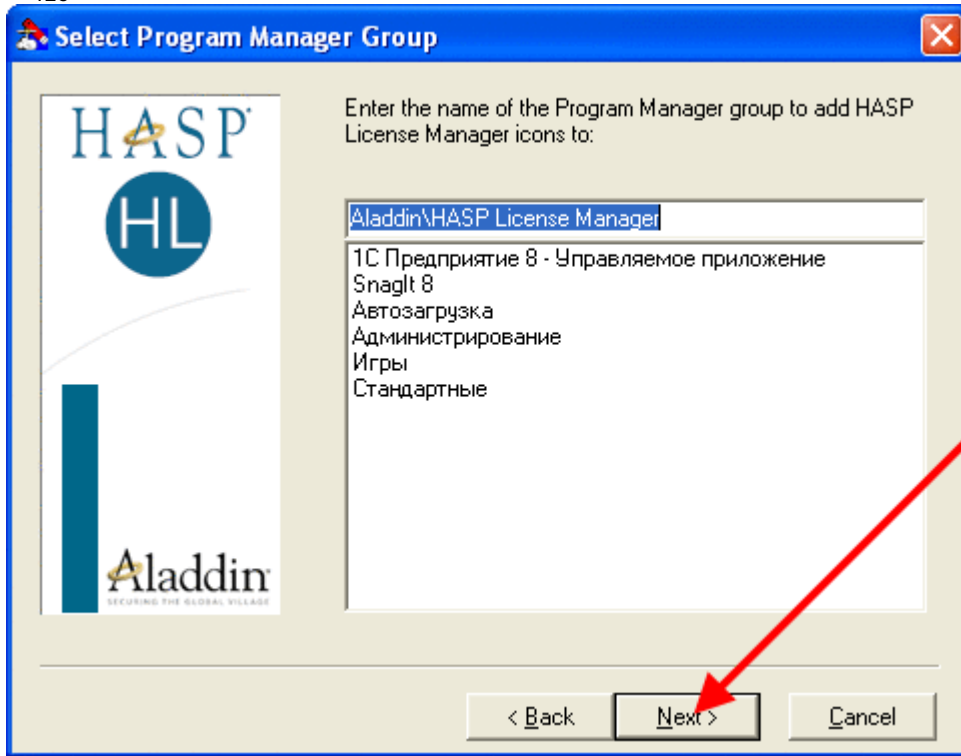


Рис. 81. Указание имени группы

При установке HASP License Manager в качестве приложения Windows будет предложено поместить ярлык HASP License Manager в каталог **Автозагрузка (Startup)**. В этом случае будет осуществляться автоматический запуск HASP License Manager при загрузке операционной системы. При выборе альтернативного варианта придется запускать HASP License Manager вручную.

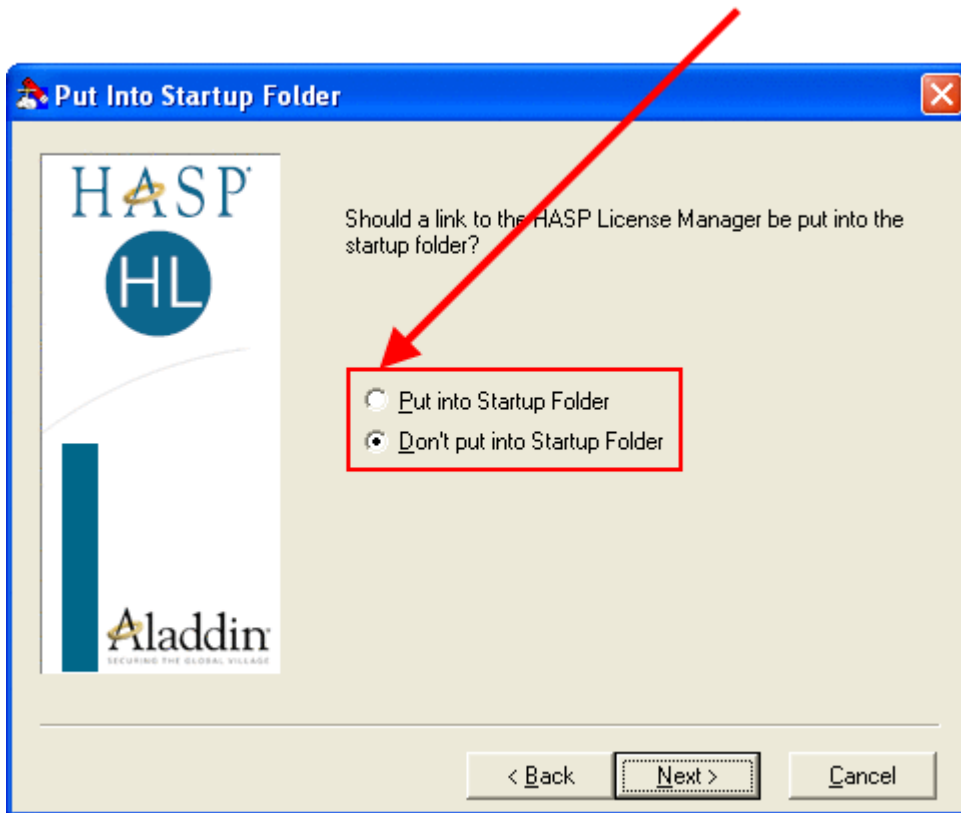


Рис. 82. Выбор режима запуска

На следующем этапе предлагается установить HASP Device Driver, необходимый для нормальной работы HASP License Manager. С помощью этого драйвера осуществляется взаимодействие HASP License Manager с аппаратным ключом HASP4 Net. Если HASP Device Driver уже был установлен на компьютере, то повторная установка HASP License Manager не требуется.

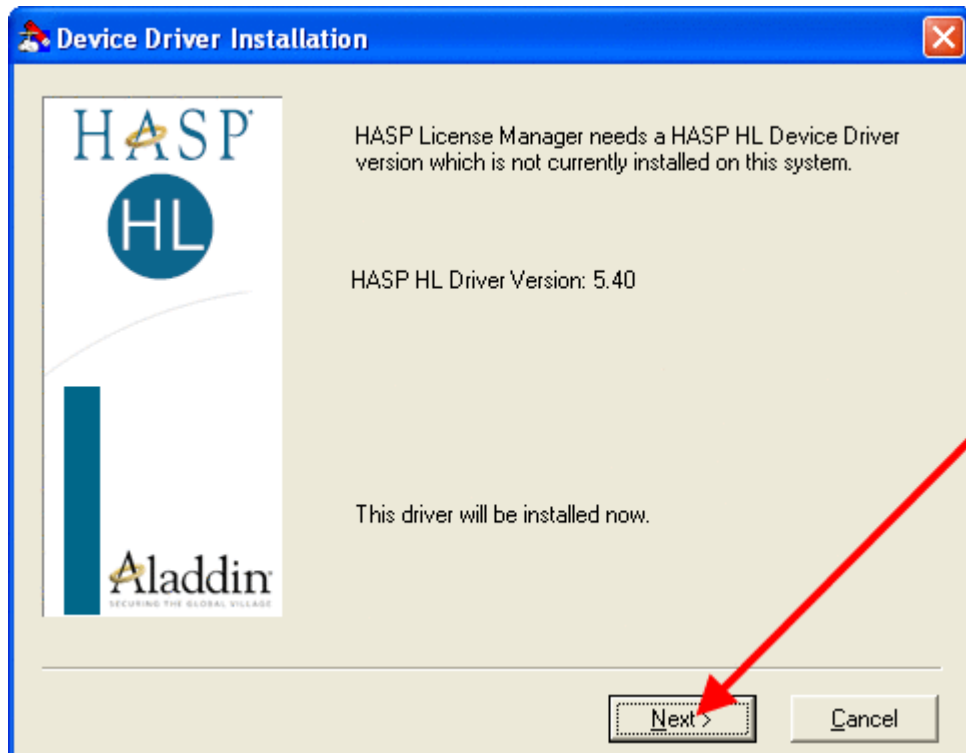


Рис. 83. Установка драйвера защиты

После завершения процесса установки будет предложено запустить HASP License Manager. При отказе можно будет запустить его позже вручную. Процедуры запуска HASP License Manager для различных вариантов установки описаны ниже.

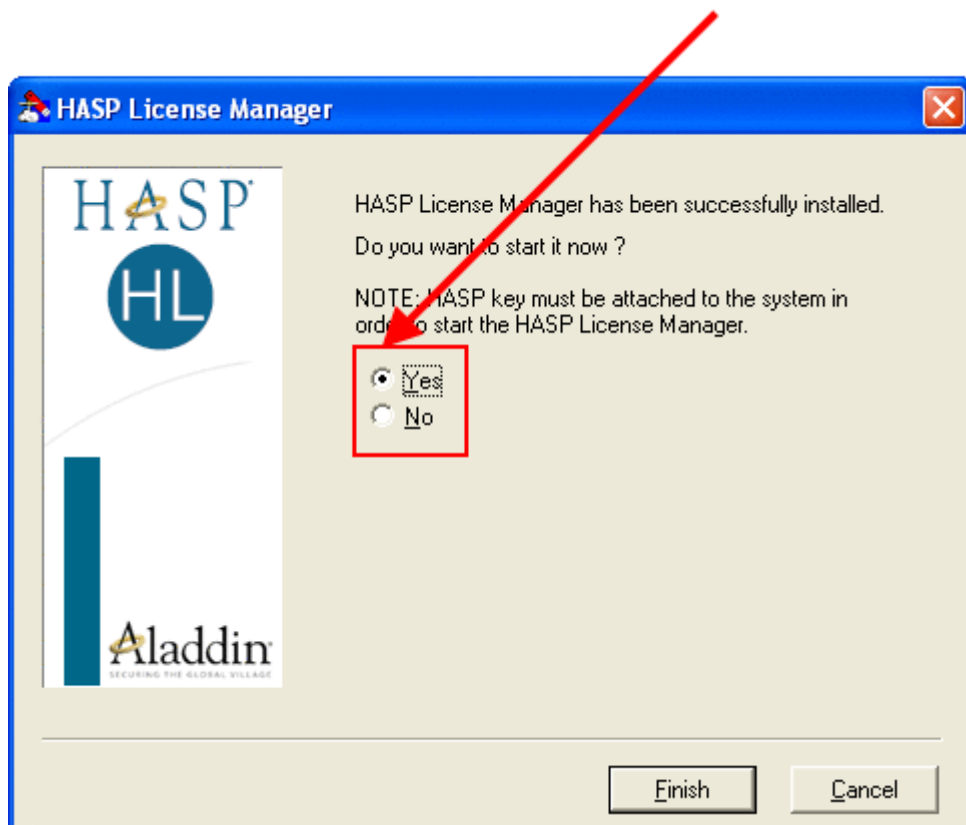


Рис. 84. Необходимость запуска HASP License Manager

9.1.6.1.1. Запуск HASP License Manager как приложения Microsoft Windows

Если HASP License Manager был установлен как приложение Microsoft Windows, то он запускается с помощью программы [nhsrvw32.exe](#), которая размещается на жестком диске компьютера программой установки HASP License Manager.

При запуске из командной строки программе [nhsrvw32.exe](#) могут быть заданы параметры, с помощью которых HASP License Manager может быть более точно «проинструктирован» об использовании того или иного сетевого протокола для взаимодействия с защищенными программами.

Следует заметить, что настройку сетевых протоколов имеет смысл производить только в тех случаях, когда режим использования сетевых протоколов по умолчанию приводит к неустойчивой работе или наблюдаются серьезные задержки при запуске защищенных программ.

Перед каждым параметром должен быть указан символ «-» или «/». Например:

```
nsvw32 -tcpip
```

[Копировать в буфер обмена](#)

Или:

```
nsvw32 /tcpip
```

[Копировать в буфер обмена](#)

При запуске программы [nhsrvw32.exe](#) могут быть использованы следующие параметры.

```
-addrpath=<path>
```

Определяет место сохранения файла [haspaddr.dat](#). По умолчанию файл сохраняется в том каталоге, откуда был загружен HASP License Manager.

```
-ipx
```

Инструктирует систему HASP4 Net использовать протокол IPX с SAP.

```
-ipxnosap
```

Инструктирует систему HASP4 Net использовать протокол IPX без SAP. При использовании HASP License Manager для Win32 другие протоколы можно загрузить с помощью ключей [-tcpip](#) или [-netbios](#). В этом случае HASP License Manager создает файл [newaddr.dat](#), в котором содержится адрес станции, на которой запущен HASP License Manager. При загрузке HASP License Manager с одним из этих ключей обмениваться данными с ним смогут только те защищаемые приложения, которые имеют доступ к файлу [newaddr.dat](#).

```
-ipxsocket num=<number>
```

Данный ключ следует использовать в тех случаях, когда необходимо изменить сокет, который используется для обмена данными HASP License Manager. Сокет по умолчанию – **7483** (шестнадцатеричное значение).

```
-localnet
```

Данный ключ следует использовать только в том случае, если вы хотите, чтобы HASP License Manager обслуживал станции исключительно в локальной сети. Если HASP License Manager получает запросы от станций, которые не входят в локальную сеть, им возвращается код ошибки 140.

```
-nbname=<name>
```

Присваивает HASP License Manager имя NetBIOS. Действие ключа идентично [-nethaspnb name](#).

```
-netbios
```

Данный ключ позволяет использовать системе HASP4 Net исключительно протокол NetBIOS. При использовании HASP License Manager для Win 32 другие протоколы можно загрузить с помощью ключей [-tcpip](#) или [-ipxnosap](#).

```
-portnum=<number>
```

Если используется протокол TCP/IP, данный ключ позволяет задать сетевой порт, который будет

использовать HASP License Manager. Порт по умолчанию – 475.

`-srvname=<name> [,name]`

Присваивает HASP License Manager одно или несколько имен IPX, TCP/IP или NetBIOS. Может быть присвоено не более шести имен.

`-tcpip`

Данный ключ позволяет использовать системе HASP4 Net исключительно протокол TCP/IP. При использовании HASP License Manager для Win32 другие протоколы можно загрузить с помощью ключей `-ipx` или `-netbios`.

`-use lananum=<x> [,x]`

Инструктирует HASP License Manager работать с определенными номерами коммуникационного канала.

`-userlist`

Ограничивает число пользователей, обслуживаемых HASP License Manager. Значение по умолчанию – 250.

9.1.6.1.2. Запуск HASP License Manager как службы Microsoft Windows

HASP License Manager может быть запущен как служба Microsoft Windows только в том случае, если был установлен для работы в качестве службы. А это, как было отмечено выше, возможно только в среде операционных систем Microsoft Windows 2000/XP/Server 2003/Vista/Windows 7/Server 2008.

При установке HASP License Manager как службы Microsoft Windows он устанавливается как запускаемый автоматически, то есть служба HASP License Manager будет стартовать при каждом запуске Microsoft Windows.

При необходимости можно изменить настройки запуска службы и осуществлять ее запуск и остановку «вручную».

Для запуска, остановки и настройки службы HASP License Manager «вручную» следует обратиться к системному меню **Пуск – Настройка – Панель управления – Администрирование – Службы (Start – Control Panel – Administrative Tools – Services)**. В появившемся списке служб нужно найти службу **HASP Loader** и щелкнуть на ней правой кнопкой мыши. Через появившееся контекстное меню можно осуществить все необходимые действия со службой.

9.1.6.2. Для ОС Linux

HASP License Manager для ОС Linux можно загрузить с сайта компании Aladdin (<http://www3.safenet-inc.com/support/hasp/hasp4/enduser.aspx#lm>).

ПРИМЕЧАНИЕ. Для успешной работы HASP License Manager необходимо также установить драйвер ключа защиты, который можно загрузить с сайта компании Aladdin (<http://www3.safenet-inc.com/support/hasp/enduser.aspx>) в зависимости от версии используемой ОС Linux.

Для установки HASP License Manager необходимо выполнить следующие действия (действия должны выполняться от имени администратора):

- Скопировать загруженный файл в каталог, где будет располагаться HASP License Manager (например, `/opt/hasplm`).
- Распаковать архив, воспользовавшись следующей командой:

```
tar xzf hasplm_linux_8.30.tgz
```

Копировать в буфер обмена

- Добавить в файл `/etc/rc.local` команду запуска HASP License Manager (до команды `exit 0`) из

```
/opt/hasplm/hasplm
```

Добавление команды в файл [rc.local](#) приведет к тому, что HASP License Manager автоматически запустится при перезапуске системы.

- Запустить HASP License Manager:

```
hasplm
```

Если требуется выполнить настройку HASP License Manager с помощью конфигурационного файла [nhsrv.ini](#) (см. [здесь](#)), то путь к конфигурационному файлу следует указать в командной строке HASP License Manager:

```
/opt/hasplm/hasplm -c /etc/nhsrv.ini
```

9.1.6.3. Настройка HASP License Manager с помощью файла конфигурации

Некоторые настройки HASP License Manager могут задаваться при помощи файла конфигурации [nhsrv.ini](#) (см. [здесь](#)).

Если используются ключи с большим количеством пользовательских лицензий (на 300 и 500 пользователей), то при настройке HASP License Manager следует обратить внимание на параметр [NHS_USERLIST](#).

9.1.7. Настройка системы «1С:Предприятие» для работы с HASP License Manager

Система «1С:Предприятие» способна использовать для связи с HASP License Manager сетевые протоколы IPX, TCP/IP или NetBIOS. По умолчанию определение сетевого протокола производится автоматически. Этот режим рекомендуется к использованию всегда, кроме тех случаев, когда автоматический режим определения сетевого протокола и установки связи работает нестабильно или вызывает существенные задержки.

ПРИМЕЧАНИЕ. Обращение к HASP License Manager **всегда** выполняется по протоколу UDP. Указание в файле [nethasp.ini](#) протокола TCP/IP игнорируется.

Для настройки параметров взаимодействия системы «1С:Предприятие» с HASP License Manager используется конфигурационный файл [nethasp.ini](#) (см. [здесь](#)).

Пример файла [nethasp.ini](#):

```
[NH_COMMON]
NH_TCPIP=Enabled
[NH_TCPIP]
NH_SERVER_ADDR=192.168.0.12
NH_PORT_NUMBER=475
NH_TCPIP_METHOD=UDP
NH_USE_BROADCAST=Disabled
```

В данном примере сервер защиты находится в сети по адресу **192.168.0.12**, используется сетевой порт **475**, для работы **используются UDP-пакеты** и **запрещен широковещательный механизм TCP/IP**.

При установке системы «1С:Предприятие» образец файла [nethasp.ini](#) копируется в каталог конфигурационных файлов системы «1С:Предприятие». Данный файл практически полностью состоит из закомментированных строк и никак не переопределяет значений параметров по умолчанию, но при этом он содержит наиболее полный список параметров, которые могут использоваться для настройки взаимодействия системы «1С:Предприятие» с HASP License Manager.

9.2. Система программного лицензирования

Система программного лицензирования обеспечивает совместную работу пользователей без использования каких-либо дополнительных физических устройств. Для работы необходимо использовать специальный файл – **программная лицензия платформы**. В этом файле, в зашифрованном виде, находится информация, требуемая для функционирования системы – параметры самой лицензии и характеристики компьютера, для которого активирована лицензия. Для активации программной лицензии необходимо особое число – **пинкод**, а также серийный номер комплекта. В комплекте поставки поставляются несколько пинкодов. Количество пинкодов в комплекте поставки и количество одновременно активных пинкодов определяются вариантом лицензии.

Приложение «1С:Предприятия» (клиентское или серверное), при необходимости получения лицензии, выполняет поиск файлов лицензий по всем доступным путям (см. [здесь](#)). Далее из файлов получают параметры самой лицензии и характеристики компьютера, для которого лицензия получена. Если параметры текущего компьютера совпадают с полученными – выполняются проверки, связанные с количеством пользователей и типом лицензии (клиентская или серверная), в противном случае лицензия отвергается. Доступ к файлу определяется правами доступа используемой операционной системы. Если пользователь, от имени которого работает приложение, не имеет доступа к файлу лицензии (или каталогу, в котором этот файл расположен), значит лицензия не будет получена.

Программные лицензии бывают:

- Однопользовательские клиентские. Позволяют запускать произвольное количество клиентских приложений на одном компьютере.
- Многопользовательские клиентские. Позволяют запускать не более определенного количества клиентских приложений с произвольных компьютеров. Количество одновременно запущенных клиентских приложений определяется номиналом лицензии.
- Комбинированные клиентские. Являются комбинацией группы однопользовательских и одной многопользовательской лицензии. Если первой активируется любая однопользовательская лицензия, – многопользовательскую лицензию активировать не получится, и допустимо использование только однопользовательских лицензий. Если первой активируется многопользовательская лицензия, – однопользовательские лицензии активировать не получится.
- Серверная лицензия на 32-разрядный сервер. Позволяет использование произвольного количества 32-разрядных рабочих процессов (**rphost**) на одном компьютере.
- Серверная лицензия на 64-разрядный сервер. Позволяет использование произвольного количества 32-разрядных или 64-разрядных рабочих процессов (**rphost**) на одном компьютере.
- Серверная лицензия с ограниченным количеством одновременно обслуживаемых сеансов. Позволяет использование одного рабочего сервера любой разрядности (32 или 64) на одном компьютере.

Многопользовательские лицензии могут располагаться на компьютере сервера «1С:Предприятие», модуля расширения веб-сервера или терминальном сервере. На клиентском компьютере могут располагаться только однопользовательские лицензии. Программные лицензии, расположенные на сервере («1С:Предприятия» или терминальном), складываются без ограничений.

При активации программной лицензии происходит сбор информации о компьютере (ключевые параметры), для которого получается лицензия. Если в процессе работы будет изменен хотя бы один из ключевых параметров, – будет необходимо повторно активировать программную лицензию (с использованием нового пинкода). Параметры компьютера опрашиваются не чаще одного раза в сутки. Далее приводится список ключевых параметров:

- наименование операционной системы;
- версия операционной системы, при этом для ОС Windows анализируются только первые две цифры номера версии;
- серийный номер операционной системы (только в случае ОС Windows);

- дата установки операционной системы (только в случае ОС Windows);
- сетевое имя компьютера;
- модель материнской платы;
- объем оперативной памяти;
- тип и версия BIOS;
- список процессоров и их параметры;
- список сетевых адаптеров и их MAC-адреса, однако, из процедуры сравнения ключевых параметров исключаются:
 - сетевые адаптеры Bluetooth;
 - сетевые адаптеры, подключенные по IEEE 1394 или USB;
 - программные адаптеры WAN и RAS;
 - адаптеры, не имеющие MAC-адреса и данных VEN_ и DEV_ из PNP-идентификатора.
- список жестких дисков и их параметры, однако, из процедуры сравнения ключевых параметров исключаются:
 - внешние накопители, подключаемые по IEEE 1394 и USB.

ПРИМЕЧАНИЕ. При проверке соответствия параметров программной лицензии параметрам текущего компьютера, не анализируется наименование и версия операционной системы, если проверка выполняется в ОС Linux.

При использовании «1С:Предприятия» на виртуальных компьютерах, необходимо активирование программной лицензии на каждый виртуальный компьютер. При использовании виртуальных машин программная лицензия привязывается к параметрам виртуальной машины (параметры виртуальной машины эквивалентны параметрам реального компьютера и перечислены выше). Изменение этих параметров потребует повторной активации лицензии на новый пинкод.

При планировании использования программных лицензий следует помнить о следующих особенностях:

- из одного сеанса операционной системы к одному файлу программной лицензии могут получить одновременный доступ не более 256 процессов;
- на одном компьютере доступ к одному файлу программной лицензии можно получить не более чем из 256 сеансов операционной системы.

При изменении ключевых параметров компьютера следует помнить о следующих особенностях:

- При проверке информации о компьютере анализируется только удаление, а не добавление устройств. Например, при активации программной лицензии на компьютере был установлен один сетевой адаптер. Можно добавить еще один сетевой адаптер, без необходимости повторной активации программной лицензии, но нельзя заменить один сетевой адаптер на другой.
- Оперативную память на компьютере можно увеличивать, но нельзя уменьшать. Например, активация лицензии выполнялась с оперативной памятью равной 2 Гбайт. Без необходимости повторной активации программной лицензии имеется возможность увеличить память до 6 Гбайт, а потом уменьшить ее объем до 4 Гбайт. Однако уменьшение объема оперативной памяти ниже 2 Гбайт приведет к необходимости повторной активации программной лицензии.
- Изменения анализируются по текущему состоянию компьютера относительно того состояния, когда выполнялась активация лицензии.

Для получения программной лицензии на электронном носителе предназначена страница <http://users.v8.1c.ru/lc>. Она позволяет передать подготовленный файл запроса на лицензию в [Центр лицензирования](#) и получить из [Центра лицензирования](#) файл данных лицензии. Для автоматического получения программной лицензии используется веб-сервис <http://users.v8.1c.ru/LicenseCenter/ws/lm.1cws>.

9.2.1. Варианты лицензий

Варианты пользовательских программных лицензий:

Тип	Лицензий	Набор пинкодов		
		Пользователей	Активных	Поставляется
Однопользовательская, 1 пользователь	1	1	1	3
Комбинированная, 5 пользователей	5	1	5	8
		5	1	3
Комбинированная, 10 пользователей	10	1	10	14
		10	1	3
Комбинированная, 20 пользователей	20	1	20	25
		20	1	3
Многопользовательская, 50 пользователей	50	50	1	3
Многопользовательская, 100 пользователей	100	100	1	3
Многопользовательская, 300 пользователей	300	300	1	3
Многопользовательская, 500 пользователей	500	500	1	3
Многопользовательская, 1000 пользователей	1000	2*500	2*1	2*3

Многопользовательская лицензия на 1000 пользователей поставляется как комплект из двух многопользовательских лицензий на 500 пользователей.

Для комбинированных лицензий можно определить, какой вид лицензий наиболее подходит для работы. Если при работе с комбинированной лицензией первой активируется однопользовательская лицензия, то считается, что выбран набор однопользовательских лицензий и дальнейшая активация многопользовательской лицензией становится невозможной. Если первой активируется многопользовательская лицензия, то считается, что для работы выбрана многопользовательская лицензия и дальнейшая активация однопользовательских лицензий становится невозможной.

ПРИМЕЧАНИЕ. Пинкоды, дополнительно входящие в состав поставки, можно использовать в том случае, если изменены ключевые параметры лицензии.

Варианты серверных программных лицензий:

Тип	Описание	Набор пинкодов	
		Активных	Поставляется
Серверная, 32-разрядная	Возможность работы произвольного количества 32-разрядных рабочих процессов на одном физическом компьютере	1	3
Серверная,	Возможность работы произвольного	1	3

64-разрядная	количества 32-разрядных и 64-разрядных рабочих процессов на одном физическом компьютере		
Серверная, с ограниченным количеством одновременно обслуживаемых сеансов	Возможность работы одного рабочего сервера любой разрядности на одном физическом компьютере	1	3

Серверная лицензия с ограниченным количеством одновременно обслуживаемых сеансов позволяет запустить на одном компьютере один рабочий сервер произвольной разрядности (32– или 64-разрядный) и при этом обслуживать одновременно не более 5 клиентских сеансов и 1 сеанс конфигуратора. Под клиентскими сеансами понимаются следующие сеансы: толстый клиент, тонкий клиент, веб-клиент, внешнее соединение. Для работы клиентских приложений используются клиентские лицензии, получаемые обычным способом.

9.2.2. Защита от несанкционированного использования

Для предотвращения возможности незаконного использования система «1С:Предприятие» предоставляется пользователям в защищенном виде.

Возможность использования программного продукта на одном или нескольких рабочих местах, а также возможность использования сервера «1С:Предприятия» определяются имеющимися лицензионными соглашениями.

9.2.3. Особенности учета клиентских лицензий

В зависимости от вида клиента и места расположения файлов с программными лицензиями возможны несколько вариантов учета лицензий. Рассмотрим их более подробно.

9.2.3.1. Файловый вариант

В данном случае возможно использование только однопользовательских лицензий, что обеспечивает запуск на компьютере, к которому привязана лицензия, произвольного количества экземпляров системы в режиме 1С:Предприятие или Конфигуратор.

Исключением является терминальный режим использования «1С:Предприятия». В этом случае возможно использование с файловым вариантом информационной базы многопользовательских лицензий. Подробнее см. [здесь](#).

9.2.3.2. Клиент-серверный вариант

В данном случае возможны следующие варианты получения лицензий.

9.2.3.2.1. Однопользовательская программная лицензия

Обеспечивает запуск на компьютере, к которому привязана лицензия, произвольного количества экземпляров системы в режиме 1С:Предприятие или Конфигуратор.

9.2.3.2.2. Многопользовательская программная лицензия

Подсчетом лицензий занимается сервер «1С:Предприятия».

В этом случае файлы программных лицензии располагаются на компьютере, где установлен сервер «1С:Предприятия». Подсчетом лицензий занимается непосредственно сервер. В этом случае лицензии расходуются из расчета «один сеанс – одна лицензия». Таким образом, если на одном компьютере запущено два экземпляра «1С:Предприятия» (в любом режиме запуска и с любым видом клиента), то на это будет израсходовано две лицензии.

9.2.3.3. Веб-клиент

В зависимости от варианта информационной базы (файловый или клиент-серверный) подсчетом лицензий занимается либо модуль расширения веб-сервера (в файловом варианте) либо сервер

«1С:Предприятия» (в клиент-серверном варианте).

В этом случае файл программной лицензии может располагаться на компьютере, где установлен модуль расширения веб-сервера, либо на компьютере, где установлен сервер «1С:Предприятия». Подсчет лицензий занимается непосредственно модуль расширения веб-сервера (или сервер). В этом случае лицензии расходуются из расчета «один сеанс – одна лицензия». Таким образом, если на одном компьютере запущено два экземпляра «1С:Предприятия» (в любом режиме запуска и с любым видом клиента), то на это будет израсходовано две лицензии.

9.2.3.4. Тонкий клиент, работающий через веб-сервер

Тонкий клиент может использовать для получения лицензий:

- однопользовательскую программную лицензию;
- модуль расширения веб-сервера или сервер «1С:Предприятия».

В случае однопользовательской программной лицензии обеспечивается запуск на компьютере, к которому привязана лицензия, произвольного количества экземпляров системы в режиме 1С:Предприятие.

Если для получения лицензии используется модуль расширения веб-сервера или сервер «1С:Предприятия», то в случае файлового варианта подсчетом лицензий занимается модуль расширения веб-сервера, а в клиент-серверном варианте – сервер «1С:Предприятия». В этом случае лицензии расходуются из расчета «один сеанс – одна лицензия». Таким образом, если на одном компьютере запущено два экземпляра «1С:Предприятия» (в любом режиме запуска и с любым видом клиента), то на это будет израсходовано две лицензии.

При этом файл программной лицензии может располагаться на компьютере, где установлен модуль расширения веб-сервера либо на компьютере, где установлен сервер «1С:Предприятия».

9.2.3.5. Однопользовательская программная лицензия и веб-клиент

В случае, если на компьютере с сервером «1С:Предприятия» или веб-сервером (в случае файлового варианта информационной базы) установлена однопользовательская программная лицензия, имеется возможность запустить:

- Произвольное количество конфигураторов на компьютере с однопользовательской программной лицензией.
- Произвольное количество клиентских приложений (кроме веб-клиента) на других компьютерах, если им доступны клиентские лицензии.
- Также доступен один из следующих вариантов запуска:
 - Одно произвольное клиентское приложение (включая веб-клиента) на любом компьютере, если ему недоступна клиентская лицензия.
 - Произвольное количество клиентских приложений (кроме веб-клиента) на компьютере с программной лицензией.

Другими словами, имеется возможность вести разработку и отладку с использованием веб-клиента, используя только однопользовательскую программную лицензию. При этом при использовании веб-клиента на локальном компьютере остается возможность запуска только конфигулятора, другие виды клиентов не могут быть запущены.

9.2.3.6. COM-соединение

При использовании 32-разрядного COM-соединения поиск доступной лицензии осуществляется в следующем порядке:

- однопользовательские программные лицензии;

- многопользовательские программные лицензии;
- клиентские лицензии на сервере «1С:Предприятия» (в клиент-серверном варианте) или веб-сервере (в случае файлового варианта, подключенного через веб-сервер).

При использовании 64-разрядного COM-соединения поиск доступной лицензии осуществляется в следующем порядке:

- однопользовательские программные лицензии;
- многопользовательские программные лицензии;
- клиентские лицензии на сервере «1С:Предприятия» (в клиент-серверном варианте) или веб-сервере (в случае файлового варианта, подключенного через веб-сервер).

Если COM-соединение запускается из кода, исполняемого на сервере «1С:Предприятия», как внутрипроцессный COM-сервер и сервер использует серверную программную лицензию, то COM-соединение использует серверную программную лицензию. В противном случае COM-соединение выполняет поиск клиентской программной лицензии так, как описано выше в этом разделе.

9.2.3.7. Web-сервис

Для работы Web-сервисов не требуется клиентских лицензий. Однако если информационная база, которая предоставляет Web-сервисы, работает в клиент-серверном варианте, необходимо наличие серверной лицензии для работы сервера «1С:Предприятия».

9.2.3.8. Терминальный сервер

При использовании Windows-систем следует учитывать следующую особенность: с точки зрения системы программного лицензирования, рабочее место определяется номером идентификатора сессии. Все запросы на получения лицензии, выполненные с одного компьютера и с одним идентификатором сессии, считаются полученными с одного рабочего места. Например, если существует компьютер, на котором установлена однопользовательская программная лицензия, то этой лицензией смогут пользоваться произвольное количество клиентских приложений, запущенных интерактивно. Однако, если на этом компьютере клиентское приложение (в любом виде) будет запущено из какой-либо службы ОС Windows – это будет считаться аналогом терминального сервера, и будет требоваться дополнительная лицензия. Эта особенность касается любых программных лицензий (не обязательно однопользовательских).

Также можно рассмотреть следующие особенности учета клиентских лицензий.

9.2.3.8.1. Однопользовательская программная лицензия

Обеспечивает запуск произвольного количества экземпляров системы в режиме 1С:Предприятие или Конфигуратор от лица одной терминальной сессии.

Программные лицензии (как однопользовательские, так и многопользовательские), активированные для терминального сервера, складываются в том случае, если файлы с лицензиями доступны всем пользователям терминального сервера.

9.2.3.8.2. Многопользовательская программная лицензия

Многопользовательская программная лицензия может быть активирована для терминального сервера и использоваться как для файлового, так и для клиент-серверного варианта системы. В этом случае обеспечивается запуск произвольного количества экземпляров системы в режиме 1С:Предприятие или Конфигуратор для такого количества одновременных подключений к терминальному серверу (терминальных сессий), на сколько пользователей активирована многопользовательская программная лицензия.

Программные лицензии (как однопользовательские, так и многопользовательские), активированные для терминального сервера, складываются в том случае, если файлы с лицензиями доступны всем пользователям терминального сервера.

9.2.4. Особенности активации и получения серверной лицензии

Программная лицензия должна быть активирована для компьютера, на котором выполняется (один или несколько) рабочий процесс ([rphost](#)) кластера серверов или работает менеджер кластера ([rmngr](#)), на который назначен сервис лицензирования.

Проверка наличия серверной лицензии осуществляется в тот момент, когда клиентское приложение выполняет подключение к рабочему процессу ([rphost](#)).

9.2.5. Активация программной лицензии

9.2.5.1. Общие правила

Одной из составных частей используемой системы защиты является программная лицензия на использование «1С:Предприятия». Для активации программной лицензии необходимо воспользоваться специальным помощником. Для запуска помощника следует использовать команду [Сервис – Получение лицензии...](#) Конфигуратора.

При активации лицензии необходимо:

- Выбрать компьютер для активации программной лицензии: на текущем компьютере или на серверном компьютере.

Флажок [Установка на сервер](#) имеет смысл выбирать в случае активации любых программных лицензий, если использование активированной лицензии предполагается на компьютере, где расположен сервер «1С:Предприятия» и это не тот компьютер, на котором выполняется активация программной лицензии. В этом случае файл с активированной программной лицензией всегда будет размещаться в каталоге профиля того пользователя, от имени которого работает сервер «1С:Предприятия» (по умолчанию – [usr1cv83](#)) на том компьютере, на котором работает сервер «1С:Предприятия».

- Выбрать операцию с лицензией:
 - первичное получение лицензии,
 - повторное получение лицензии,
 - обновление лицензии.
- Лицензия активируется для всех пользователей компьютера. Если пользователь, от имени которого выполняется активация лицензии, не имеет возможности выполнять запись в каталог, доступный всем пользователям, то система сообщит об этом и предложит активировать лицензию только для текущего пользователя.
- Выбрать способ получения лицензии:
 - автоматически,
 - на электронном носителе,
 - вручную.
- Для получения лицензии на электронном носителе выбрать этап получения лицензии:
 - создать файл запроса лицензии,
 - получить лицензию из Центра лицензирования.
- Для повторного получения лицензии следует ввести активный пинкод, использование которого должно быть прекращено.
- Ввести регистрационный номер комплекта, пинкод и данные о пользователе.
- При получении лицензии вручную:

- продиктовать оператору Центра лицензирования набор цифр, которые отображаются в помощнике (48 цифр);
- ввести в специальное поле данные программной лицензии под диктовку оператора Центра лицензирования (120 цифр).
- При получении лицензии на электронном носителе:
 - Ввести имя файла запроса на лицензию. Этот файл должен быть передан в [Центр лицензирования](#).
 - Ввести имя файла ответа [Центра лицензирования](#). Этот файл должен быть предварительно получен в [Центре лицензирования](#).

ПРИМЕЧАНИЕ. Следует помнить, что при загрузке файла, полученного из [Центра лицензирования](#), параметры (информация о программном продукте и владельце лицензии), указанные в диалоге активации лицензии, должны быть в точности такими же, как и в тот момент, когда формировался файл запроса лицензии.

При активации лицензий нужно помнить о следующих особенностях:

- Если первичная активация программной лицензии выполнялась по Интернету или на электронном носителе, то повторная активация и обновление лицензии возможно только по Интернету или на электронном носителе.
- Если первичная активация программной лицензии выполнялась вручную по телефону, то повторная активация и обновление лицензии возможно только вручную по телефону.
- Если требуется активировать дополнительную клиентскую программную лицензию на компьютере, на котором уже активирована программная лицензия, то делать это необходимо в точности так, как и при первой активации программной лицензии на выбранном компьютере.
- Если выполняется активация серверной и многопользовательской программной лицензии и существует возможность запустить Конфигуратор на компьютере с установленным сервером «1С:Предприятия», то рекомендуется выполнять активацию обеих лицензий с компьютера, на котором расположен сервер «1С:Предприятия».
- В процессе проверки привязки файла лицензии к данному компьютеру сетевые устройства и внешние накопители данных, подключаемые через интерфейсы USB и IEEE 1394, не учитываются.

Однако в процессе получения лицензии на носителе требуется, чтобы в момент формирования запроса в центр лицензирования на получение лицензии и в момент построения файла лицензии с использованием ответа центра лицензирования, параметры компьютера совпадали с учетом устройств, подключенных по USB и IEEE 1394. Если это требование не будет выполнено, то при выполнении загрузки файла-ответа из центра лицензирования в диалоге получения лицензии будет выдано сообщение об ошибке: [Ответ Центра лицензирования не соответствует введенным данным лицензии или владельца. Проверьте регистрационный номер комплекта, пинкод и данные о владельце лицензии](#). Чтобы завершить активацию лицензии в этом случае необходимо вернуть конфигурацию компьютера к состоянию, которое было в момент формирования файла запроса на лицензию, например, вставить в USB-порт тот же самый внешний накопитель, и повторить загрузку того же самого файла ответа центра лицензирования. После этого накопитель можно извлечь.

Если восстановить параметры компьютера невозможно, то потребуются повторное получение лицензии с использованием дополнительного пинкода.

ВНИМАНИЕ! Не рекомендуется размещать один файл программной лицензии одновременно в нескольких различных каталогах, доступных приложениям «1С:Предприятия». Это может привести к тому, что файл с лицензией станет непригодным для использования, т. к. является одним из вариантов нарушения лицензионного соглашения.

При автоматическом получении файла с программными лицензиями этот файл располагается:

- Для компьютера с сервером «1С:Предприятия»:

- Для ОС Windows: в каталог `%ALLUSERSPROFILE%\Application Data\1C\licenses` (`%ALLUSERSPROFILE%\1C\licenses` для ОС Windows Vista и старше) пользователя, от имени которого работает сервер «1С:Предприятие».
- Для ОС Linux: каталог `/var/1C/licenses`.
- Для текущего компьютера – будет задан вопрос, кому может быть доступна получаемая лицензия:
 - В случае выбора текущего пользователя файл будет помещен в каталог:
 - Для ОС Windows: `%USERPROFILE%\Local Settings\Application Data\1C\1Cv8\Conf` (`%LOCALAPPDATA%\1C\1Cv8\Conf` для ОС Windows Vista и старше) пользователя, от имени которого выполняется получение лицензии.
 - Для ОС Linux: `~/1cv8/conf` (~ – домашний каталог пользователя, от имени которого конфигурируется).
 - В случае выбора всех пользователей файл будет помещен в каталог:
 - Для ОС Windows: `%ALLUSERSPROFILE%\Application Data\1C\licenses` (`%ALLUSERSPROFILE%\1C\licenses` для ОС Windows Vista и старше) данных для всех пользователей компьютера.
 - Для ОС Linux: такой вариант не поддерживается.

Каталоги `%ALLUSERSPROFILE%\Application Data\1C\licenses` (`%ALLUSERSPROFILE%\1C\licenses` для ОС Windows Vista и старше) и `/var/1C/licenses` создаются при установке системы на компьютер (в соответствующей операционной системе). При этом следует учитывать следующие особенности, связанные с этими каталогами:

- Для ОС Windows: права записи и чтения в создаваемый каталог получает пользователь, от имени которого работает сервер «1С:Предприятия» (выбираемый в окне установки системы, см. [здесь](#)). Если при установке флажок [Установить сервер 1С:Предприятия как сервис Windows](#) не установлен, то права на создаваемый каталог никому не назначаются и остаются в варианте по умолчанию.
- Для ОС Linux: в процессе установки создается группа `grp1cv8`, в которую должны входить все учетные записи пользователей, от имени которых работают сервера «1С:Предприятия» в режиме демона. При этом на создаваемый каталог назначаются права следующим образом:
 - Владелец каталога: `root`. Права: чтение и запись (`rwX`).
 - Группа владельца: `grp1cv8`. Права: чтение и запись (`rwX`).
 - Права для остальных пользователей: только чтение (`r-x`).

9.2.5.2. Рекомендации по активации

Не имеет смысла активировать многопользовательскую программную лицензию на клиентский компьютер (в любом варианте информационно-базы).

ПРИМЕЧАНИЕ. Клиентское приложение, работающее на ОС Linux, позволяет активировать лицензию, доступную всем пользователям, только в том случае, если активация выполняется от имени пользователя `root`.

9.2.6. Расположение файлов с программными лицензиями

Программная лицензия представляет собой файл с расширением `.lic`, который может располагаться в различных местах файловой системы.

СОВЕТ. Не рекомендуется размещать один файл программной лицензии одновременно в нескольких различных каталогах, доступных приложениям «1С:Предприятия». Это может привести к тому, что файл с лицензией станет непригодным для использования, т. к. является одним из вариантов нарушения лицензионного соглашения.

Получение программной лицензии выполняется следующим образом:

- Формируется список файлов программных лицензий по всем каталогам, в которых могут располагаться программные лицензии.
- Для каждого файла из списка выполняется получение требуемой лицензии (клиентской или серверной) до момента успешного получения лицензии или до окончания списка файлов с программными лицензиями.

9.2.6.1. Для ОС Windows

В ОС Windows файлы программных лицензий могут располагаться в следующих каталогах (каталоги перечислены в порядке поиска):

- Каталог конфигурационных файлов конкретной версии платформы. По умолчанию – [C:\Program Files\1Cv8\8.3.XX.YYY\bin\conf](#).
- Каталог [%USERPROFILE%\Local Settings\1C\1Cv8\Conf](#) ([%LOCALAPPDATA%\1C\1Cv8\Conf](#) для ОС Windows Vista и старше) пользователя, от имени которого работает система.
- Каталог, который указан в файле [conf.cfg](#), расположенном в каталоге [bin\conf](#) конкретной версии. Подробнее о формате файла [conf.cfg](#) см. [здесь](#).
- Каталог [%ALLUSERSPROFILE%\Application Data\1C\1Cv8\Conf](#) ([%ALLUSERSPROFILE%\1C\1Cv8\Conf](#) для ОС Windows Vista и старше) данных для всех пользователей компьютера.
- Каталог [%ALLUSERSPROFILE%\Application Data\1C\licenses](#) ([%ALLUSERSPROFILE%\1C\licenses](#) для ОС Windows Vista и старше) данных для всех пользователей компьютера.
- Каталог [%ALLUSERSPROFILE%\1C\licenses](#) ([%ProgramData%\1C\licenses](#) для ОС Windows Vista и старше) данных для всех пользователей компьютера.

Если во всех этих каталогах лицензия не была обнаружена, то для поиска используется каталог [%APPDATA%\1C\1cv8\](#). Если в этом каталоге имеется файл [location.cfg](#), то для поиска будет использоваться каталог, указанный в параметре [location](#) (см. [здесь](#)).

9.2.6.2. Для ОС Linux

В ОС Linux файлы программных лицензий могут располагаться в следующих каталогах (каталоги перечислены в порядке поиска):

- Каталог [conf](#) установленной версии. Для 32-разрядной версии «1С:Предприятия» путь к этому каталогу будет выглядеть следующим образом: [/opt/1C/v8.3/i386/conf](#), а для 64-разрядной версии: [/opt/1C/v8.3/x86_64/conf](#).
- Каталог [~/1cv8/conf](#) (~ – домашний каталог пользователя, от имени которого работает сервер «1С:Предприятия»).
- Каталог, который указан в файле [conf.cfg](#), расположенном в каталоге [conf](#) установленной версии. Подробнее о формате файла [conf.cfg](#) см. [здесь](#).
- Каталог [/var/1C/licenses](#).

Если во всех этих каталогах лицензия не была обнаружена, то для поиска используется каталог [~/1cv8/1C/1cv8/](#). Если в этом каталоге имеется файл [location.cfg](#), то для поиска будет использоваться каталог, указанный в параметре [location](#) (см. [здесь](#)).

9.3. Определение возможности запуска

9.3.1. При запуске клиентского приложения

При запуске «1С:Предприятие» выполняет проверку возможности запуска по следующему алгоритму

(если на каком-либо шаге обнаруживается необходимая лицензия, дальнейший поиск прекращается):

1. На компьютере с клиентским приложением:

- Выполняется попытка получения лицензии из того файла программной лицензии или ключа защиты **HASP** того же типа (серия, сетевой или локальный), откуда лицензия была получена при последнем успешном подключении (см. [здесь](#));
- Выполняется поиск программных лицензий на локальном компьютере;
- Выполняется поиск локального ключа **HASP**;
- Выполняется поиск многопользовательского ключа **HASP**, доступного с помощью программы **HASP License Manager**.
- Если конфигурация базовая, клиентское приложение выполняет поиск лицензии базовой версии на локальном компьютере.

2. На компьютере менеджера кластера, на который назначен сервис сеансовых данных:

- Выполняется попытка получения лицензии из того файла программной лицензии или многопользовательского ключа защиты **HASP** того же типа (серия, сетевой или локальный), откуда лицензия была получена при последнем успешном подключении (см. [здесь](#));
- Выполняется поиск программных лицензий на компьютере менеджера кластера сервера «1С:Предприятия»;
- Выполняется поиск многопользовательских ключей **HASP**, установленных на компьютере менеджера кластера сервера «1С:Предприятия»;
- Выполняется поиск многопользовательского ключа **HASP**, доступного с помощью программы **HASP License Manager**.

3. На компьютере менеджера кластера, на который назначен сервис лицензирования:

- Выполняется попытка получения лицензии из того файла программной лицензии, откуда лицензия была получена при последнем успешном подключении (см. [здесь](#));
- Выполняется поиск программных лицензий на компьютере менеджера кластера сервера «1С:Предприятия».

Если поиск ключа защиты **HASP** выключен (параметр `UseHwLicenses=0`) с помощью конфигурационного файла `1cestart.cfg` (см. [здесь](#)), то при определении возможности запуска не выполняется поиск свободных лицензий в ключах защиты **HASP**, доступных на клиентском компьютере (как локальных, так и сетевых), а также не выполняется попытка получения лицензии из запомненного ключа.

9.3.2. При запуске сервера

При установке соединения клиентского приложения с сервером «1С:Предприятия», выполняется проверка наличия серверной лицензии (если на каком-либо шаге обнаруживается необходимая лицензия, дальнейший поиск прекращается):

- Выполняется поиск лицензии на компьютере рабочего процесса, обслуживающего соединение с информационной базой:
 - Выполняется попытка получения лицензии из того файла программной лицензии или ключа защиты **HASP**, откуда лицензия была получена при последнем успешном подключении (см. [здесь](#));
 - Выполняется поиск программной лицензии 32-разрядного сервера (только для 32-разрядного сервера «1С:Предприятия»);
 - Выполняется поиск программной лицензии 64-разрядного сервера;

- Выполняется поиск локального ключа 32-разрядного сервера (см. [здесь](#)) (только для 32-разрядного сервера «1С:Предприятия»);
- Выполняется поиск локального ключа 64-разрядного сервера (см. [здесь](#)).
- Выполняется поиск лицензии на менеджере кластера, на который назначен сервис лицензирования:
 - Выполняется попытка получения лицензии из того файла программной лицензии, откуда лицензия была получена при последнем успешном подключении (см. [здесь](#));
 - Выполняется поиск программной лицензии 32-разрядного сервера (только для 32-разрядного сервера «1С:Предприятия»);
 - Выполняется поиск программной лицензии 64-разрядного сервера.

9.3.3. Действия при неполучении лицензии

Если в процессе поиска (описано выше) не обнаружено ни одной лицензии, то выполняются следующие действия:

- Конфигуратор, тонкий и толстый клиенты запускают помощника получения программной лицензии.
- Веб-клиент формирует сообщение об ошибке:
 - **Не найдена лицензия. Не обнаружен ключ защиты программы или полученная электронная лицензия!**
 - **На сервере 1С:Предприятия не найдена лицензия. Не обнаружен ключ защиты программы или полученная электронная лицензия!**
- При отказе от получения программной лицензии конфигуратор, тонкий и толстый клиенты также формируют вышеописанные сообщения об ошибке.

Файл программной лицензии считается доступным для использования, если:

- Не содержится в черном списке.
- Имеет корректный формат.
- Привязан к текущему компьютеру.
- Содержит свободную лицензию.
- В сети не используются другие файлы лицензий, полученных для того же пинкода и серийного номера программы. Если обнаруживается такая ситуация, то файл лицензии делается непригодным для использования и помещается в черный список.

ПРИМЕЧАНИЕ. Лицензии, предоставляемые с помощью ключей HASP и с помощью файлов программных лицензий, могут складываться и использоваться одновременно. В случае совместного использования вначале будут использоваться программные лицензии, а затем – лицензии из ключей HASP.

Глава 10. Обновление системы

10.1. Обновление

В случае если будет запущена программа [setup.exe](#) из каталога дистрибутива версии, которая уже установлена на компьютере пользователя, произойдет автоматическое обновление установленной версии в соответствии с настройками, заданными параметром `InstallComponents` конфигурационных файлов (см. [здесь](#)).

Запуск программы [setup.exe](#) из каталога дистрибутива версии, не установленной на компьютере, приведет к установке этой версии, а не к обновлению какой-либо из установленных предыдущих версий.

10.2. Обновление системы «1С:Предприятие» пользователями Microsoft Windows без прав администратора

Чтобы разрешить пользователям Microsoft Windows, не обладающим административными правами в операционной системе, выполнять установку «1С:Предприятия» из общего каталога, необходимо установить политику [AlwaysInstallElevated](#) для компьютера и пользователя. Установить политику можно как локально, в панели управления групповых политик (запустив [gpedit.msc](#)), так и через управление политиками [Active Directory](#).

Вышеуказанные действия можно проделать не для конкретных пользователей, а для группы [Authenticated users](#).

Глава 11. Удаление системы

11.1. Удаление информационной базы

Программа удаления системы «1С:Предприятие» не производит автоматического удаления каталогов на жестком диске, содержащих информационные базы. Эти каталоги следует удалить самостоятельно.

Если на каталоги с информационными базами, которые необходимо удалить, существуют ссылки в списке информационных баз, то следует удалить как строки из списка информационных баз (см. [здесь](#)), так и сами каталоги.

11.2. Удаление технологической платформы

Удаление «1С:Предприятия» выполняется специальной программой, которая удаляет компоненты системы с жесткого диска компьютера, производит изменения в меню **Пуск** и системной информации Microsoft Windows.

Перед удалением необходимо закончить работу с системой «1С:Предприятие» (включая завершение работы сервера «1С:Предприятия»).

Для удаления системы «1С:Предприятие» необходимо выполнить следующие действия:

- Запустить панель управления операционной системы Microsoft Windows и щелкнуть мышью пиктограмму **Установка и удаление программ** (**Программы и компоненты** для Windows Vista и выше).
- Если необходимо, в выданном на экран диалоге щелкнуть мышью на пиктограмме **Замена или удаление**.
- В списке установленных программ выбрать строку **1С Предприятие 8 (8.3.3.100)** и нажать кнопку **Удалить**.

Будет задан вопрос о необходимости выполнения удаления. В случае утвердительного ответа будет запущено удаление выбранной версии с компьютера и выполнение необходимых изменений в системной информации.

Глава 12. Администрирование мобильной платформы

12.1. Запуск и работа со списком приложений для ОС iOS

12.1.1. Запуск мобильного приложения

Для запуска мобильного приложения необходимо найти нужное приложение в списке и нажать на его картинку. При этом в большинстве случаев будет открыто основное окно мобильного приложения. Однако если с данной программой связано несколько приложений, то будет открыт список этих приложений. В этом случае, для запуска приложения следует нажать на нужное приложение в списке.

12.1.2. Работа со списком приложений

Для выполнения большинства нижеперечисленных действий необходимо попасть в список приложений. Это можно сделать из запущенного приложения с помощью главного меню, выбрав там пункт [Список приложений](#). Также список приложений доступен сразу при запуске мобильного приложения, если зарегистрировано более одного приложения.

12.1.2.1. Создание приложения

Для того чтобы создать приложение для мобильной платформы, следует:

- Выбрать команду добавления приложения;
- Затем следует указать имя приложения и нажать кнопку [Готово](#);
- После закрытия окна будет создано приложение.

12.1.2.2. Запуск приложения

При запуске мобильного приложения, в большинстве случаев, будет открыто основное окно приложения. Однако если с данной программой связано несколько приложений, то будет открыт их список. В этом случае, для запуска приложения следует нажать на нужное приложение в списке.

12.1.2.3. Изменение свойств приложения

Свойства приложения изменяются в специальном окне. Для открытия окна свойств приложения следует в правой части строки с именем нужного приложения нажать кнопку [>](#).

В открывшемся окне можно изменить наименование приложения, запустить его (кнопка [Открыть](#)) или удалить (кнопка [Удалить](#)).

ВНИМАНИЕ! После удаления приложения, данные информационной базы восстановить невозможно.

12.1.2.4. Удаление приложения

Для удаления приложения следует нажать кнопку [Изменить](#), затем нажать картинку в левой части строки с именем удаляемого приложения. Затем в правой части этой же строки нажмите кнопку [Удалить](#).

ВНИМАНИЕ! После удаления приложения, данные информационной базы восстановить невозможно.

12.1.2.5. Обновление приложения

Мобильное приложение обновляется с помощью используемого магазина приложений. После его успешного обновления мобильного приложения, необходимо обновить ранее созданные приложения. Для этого достаточно запустить приложение.

Если в процессе выполнения обновления будет обнаружена необходимость реструктуризации базы

данных, то будет запрошено подтверждение выполнения этой операции. В случае отказа обновление будет отложено до следующего запуска приложения.

Также имеет смысл отказаться от обновления, чтобы сделать резервную копию базы данных.

12.2. Запуск и работа со списком приложений для ОС Android

12.2.1. Запуск мобильного приложения

Для запуска мобильного приложения необходимо найти нужное приложение в списке и нажать на его картинку. При этом в большинстве случаев будет открыто основное окно мобильного приложения. Однако если с данной программой связано несколько приложений, то будет открыт список этих приложений. В этом случае, для запуска приложения следует нажать на нужное приложение в списке.

12.2.2. Работа со списком приложений

Для выполнения большинства нижеперечисленных действий необходимо попасть в список приложений. Это можно сделать из запущенного приложения с помощью главного меню, выбрав там пункт [Список приложений](#). Также список приложений доступен сразу при запуске мобильного приложения, если зарегистрировано более одного приложения.

12.2.2.1. Создание приложения

Для того чтобы создать приложение для мобильной платформы, следует:

- Выбрать команду добавления приложения;
- Затем следует указать имя приложения и нажать кнопку [Готово](#);
- После закрытия окна будет создано приложение.

12.2.2.2. Запуск приложения

При запуске мобильного приложения, в большинстве случаев, будет открыто основное окно приложения. Однако если с данной программой связано несколько приложений, то будет открыт их список. В этом случае, для запуска приложения следует нажать на нужное приложение в списке.

12.2.2.3. Изменение свойств приложения

Свойства приложения изменяются в специальном окне. Для открытия окна свойств приложения следует выполнить длинное нажатие на нужном приложении. В открывшемся контекстном меню выбрать команду [Изменить](#).

В открывшемся окне можно изменить наименование приложения, запустить его (кнопка [Открыть](#)) или удалить (кнопка [Удалить](#)).

ВНИМАНИЕ! После удаления приложения, данные информационной базы восстановить невозможно.

12.2.2.4. Удаление приложения

Для удаления приложения следует выполнить следующие действия:

- Выполнить длинное нажатие на удаляемом приложении.
- В открывшемся контекстном меню выбрать команду [Удалить](#).
- Чтобы подтвердить удаление, нажать [Да](#).

ВНИМАНИЕ! После удаления приложения, данные информационной базы восстановить невозможно.

12.2.2.5. Обновление приложения

Мобильное приложение обновляется с помощью используемого магазина приложений. После его успешного обновления мобильного приложения, необходимо обновить ранее созданные приложения. Для этого достаточно запустить приложение.

Если в процессе выполнения обновления будет обнаружена необходимость реструктуризации базы данных, то будет запрошено подтверждение выполнения этой операции. В случае отказа обновление будет отложено до следующего запуска приложения.

Также имеет смысл отказаться от обновления, чтобы сделать резервную копию базы данных.

12.3. Подготовка инфраструктуры для выполнения обмена

Если мобильное приложение является частью распределенной информационной системы, то, как правило, взаимодействие в рамках этой инфраструктуры выполняется с помощью Web-сервисов. Конкретная система может предъявлять свои требования к организации взаимодействия (это определяется на этапе разработки системы). В любом случае, рекомендуется ознакомиться с документацией к системе для получения информации о том, какие механизмы используются для организации взаимодействия.

Соответственно, в задачи администратора входит организация каналов связи и создание программно-аппаратной среды, удовлетворяющей требованиям, выставляемой прикладной системой.

12.4. Резервное копирование

Резервное копирование данных выполняется в зависимости от используемого мобильного приложения.

Если используется мобильное приложение, не связанное с удаленной системой, то для резервного копирования можно воспользоваться штатными средствами используемой операционной системы мобильного устройства. При этом следует помнить о следующей особенности поведения системы под управлением ОС iOS: сразу после создания информационной базы из шаблона, она исключена из процесса резервного копирования. После того, как в информационной базе будет выполнена запись данных с помощью расширений форм объектов и регистров, информационная база включается в процесс резервного копирования.

Если используемое приложение выполняет синхронизацию данных с удаленной системой, то рекомендуется выполнить сеанс синхронизации данных. Необходимость использования штатных средств резервного копирования, после выполнения синхронизации, зависит от наличия в мобильном приложении данных, которые не синхронизируются с удаленной системой. Если таких данных нет, то в случае возникновения проблем, приложение можно пересоздать и провести начальную инициализацию данных из удаленной системы.

Глава 13. Утилита контроля целостности

13.1. Общая информация

Утилита контроля целостности (**ci**) предназначена для контроля состояния объектов файловой системы и базы данных, используемых при работе «1С:Предприятие», и обнаружения ситуации изменения этих объектов. Для определения факта неизменности объекта используется сравнение хеш-сумм контролируемых объектов (**объектов контроля**), которые вычислены по алгоритму SHA-1 (алгоритм криптографического хеширования). Процесс проверки состоит из формирования эталонных значений хеш-сумм и последующей регулярной проверки.

Утилита работает со следующими объектами контроля:

- Файлы, расположенные в файловой системе;
- Некоторые таблицы базы данных системы «1С:Предприятие».

Списки объектов контроля для работы утилиты задаются с помощью **шаблона источника информации**. Шаблон источника информации описывается **универсальным описателем источника информации** (см. [здесь](#)). Такие шаблоны могут указываться в командной строке запуска утилиты непосредственно или как содержимое файла, указанного в качестве значения параметра **in** (см. [здесь](#)). Такой файл будет называться **списком шаблонов** (см. [здесь](#)).

Перечень объектов контроля и соответствующих хеш-сумм (такая пара называется **эталоном**) хранится в специальной **базе эталонов**, который формируется утилитой при запуске в режиме формирования базы эталонов. При запуске утилиты в режиме проверки выполняется вычисление хеш-сумм и сверка вычисленных хеш-сумм с ранее сформированной базой эталонов. В результате формируется отчет о работе в виде файла.

Утилита при запуске (или в каких-либо других режимах) автоматически не контролирует собственную целостность. Ограничение доступа к утилите, базе эталонов и результатам проверок следует осуществлять средствами операционной системы.

13.2. Список шаблонов источников информации

При необходимости указания для контроля одновременно нескольких объектов контроля (или набора объектов) можно использовать специальный файл – список шаблонов источников информации. Каждая строка в этом файле содержит шаблон источника информации. Количество строк в файле не ограничено. Подробное описание формата шаблона источника информации см. [здесь](#). Комментарий начинается с символа "#". Строка комментария, а также пустые строки и строки, состоящие из одних пробелов, игнорируется при формировании списка объектов контроля.

Файл списка шаблонов должен быть только в кодировке UTF-8.

Примеры универсальных описателей источников информации:

```
# объект контроля – файл /tmp/vokas/spru.cvs
file:///tmp/vokas/spru.cvs
# объект контроля – все файлы в каталоге c:\Program Files\lcv8\8.3.4.408\bin
rdir://c:\Program Files\lcv8\8.3.4.408\bin
# объект контроля – все файлы .rc и .cfg в каталоге /home/user/.kde и подчиненных каталогах
rdir:///home/user/.kde?*.*rc,*.*cfg
# объект контроля – все файлы, начинающиеся с "V8" файлы в каталоге /tmp, без подкаталогов
ndir:///tmp?V8*.txt
# объект контроля – все поддерживаемые таблицы в базе данных, расположенной на MS SQL Server
mssql://user:password@server/instance/dbname
# объект контроля – таблицы users и config на сервер PostgreSQL
postgres://user:password@server:123/dbname?users,config
# объект контроля – таблицы users в файловой базе данных
dbe://c:\DB\checked_db?users
```

[Копировать в буфер обмена](#)

13.3. Универсальный описатель источника информации

Для описания объекта контроля используется специальный формат – универсальный описатель источника информации. В общем виде универсальный описатель источника информации выглядит следующим образом:

```
proto://[user:[password]@][server[:port]][/path/resource][?mask]
```

Подробнее рассмотрим каждую составную часть описателя:

- **proto** – описание типа объекта контроля. В качестве способа работы могут выступать:
 - **file** – отдельный файл в файловой системе;
 - **ndir** – файлы, расположенные в каталоге (без обхода подкаталогов);
 - **rdir** – файлы, расположенные в каталоге и всех вложенных каталогах;
 - **mssql** – база данных, расположенная в СУБД Microsoft SQL Server;
 - **ora** – база данных, расположенная в СУБД Oracle Database;
 - **postgres** – база данных, расположенная в СУБД PostgreSQL;
 - **db2** – база данных, расположенная в СУБД IBM DB2;
 - **dbe** – файловый вариант базы данных «1С:Предприятия».
- **user:password@** – описывает имя пользователя (**user**) и пароль (**password**), которые требуются для доступа к объекту контроля. Если выполняется контроль таблиц базы данных, то в качестве пользователя указывается пользователь СУБД, а не информационной базы «1С:Предприятия». Рекомендуется использовать параметры того пользователя, от имени которого создавалась база данных «1С:Предприятия». Символ «@» является обязательным в том случае, если указывается имя пользователя и пароль.
- **server:port** – имя компьютера (и порт доступа), на котором запущена СУБД, обслуживающая базу данных «1С:Предприятия»;
- **/path/resource** – полный путь к файлу или каталогу в нотации используемой операционной системы. В ОС Windows должен начинаться с имени диска, в ОС Linux – с обозначения корневой файловой системы (/). Если используется подключение к клиент-серверному варианту базы данных, то в качестве пути выступает имя базы данных в терминах используемой СУБД, если СУБД не поддерживает организацию экземпляров сервера, и комбинация из имени экземпляра и имени базы данных для СУБД, которые такую организацию поддерживают. Если в качестве пути к ресурсу указывается имя базы данных с указанием экземпляра СУБД, то доступ к СУБД осуществляется с использованием порта доступа по умолчанию и указание номера порта в описании

имени сервера, на котором функционирует СУБД, не поддерживается.

• **?mask** – перечень параметров описателя источника информации, разделенных запятыми. Зависит от протокола. В том случае, если в качестве объекта контроля выступают файлы, описанные по протоколам **ndir** или **rdir**, то в качестве параметров могут выступать маски файлов. Если объектом контроля выступает таблица базы данных «1С:Предприятия», то в качестве параметров могут выступать имена таблиц:

- **config** – конфигурация базы данных;
- **users** – таблица пользователей конфигурации;

Обе этих таблицы являются виртуальными и содержат в себе информацию из нескольких таблиц базы данных «1С:Предприятия». Если в универсальном описателе информации присутствует ссылка на одну из приведенных выше таблиц, то хеш-сумма считается для всех данных, входящих в состав таблицы (физические таблицы, полностью или нет). Возможность контролировать фрагмент таблицы не предоставляется.

ПРИМЕЧАНИЕ. При формировании универсального описателя источника информации следует избегать использования следующих символов: @ (кроме случая разделения имени пользователя/пароля и имени компьютера, на котором функционирует СУБД), /, \, :, ? (кроме случая указания в масках файлов, но не ранее), ~.

13.4. Файл базы эталонов

Хеш-суммы объектов контроля хранятся в специальном файле – базе эталонов. Файл содержит информацию в формате UTF-8. Местоположение и имя файла указывается при запуске утилиты. Данный файл имеет следующий формат:

[Нормализованный вид универсального описателя источника информации]
<объект контроля> = <хеш>

[Копировать в буфер обмена](#)

В этом формате **Нормализованный вид универсального описателя источника информации** содержит строку, аналогично тому, как она указана в шаблоне источников информации. Однако универсальный описатель источника информации приводится к нормализованному виду. При нормализации выполняются следующие действия:

- Для всех протоколов в конце описания объекта контроля (перед «?») добавляется прямой слеш («/»);
- Для протоколов, описывающих каталоги, в случае отсутствия маски, добавляется маска «*»;
- Для протоколов, описывающих базы данных, при отсутствии маски добавляется явное указание параметров **config,users**;
- Для протоколов, описывающих базы данных, маска «*» заменяется на явное указание **config,users**;
- Маски лексикографически сортируются;
- Обратные слешы («\») заменяются на прямые («/»);
- Несколько идущих друг за другом слешей (любых) заменяются одним прямым (например, конструкция «//\//» заменяется на «/»).

Строка <объект> = <хеш> – содержит имя объекта контроля и хеш-сумму этого объекта. Таких строк может быть более одной. Если универсальный описатель источника информации содержит указание на конкретный файл, то выражение <объект> отсутствует, и строка сразу начинается с символа «=».

Пример фрагмента базы эталонов:

```
[dbe://C:/1C DB/DB folder/?config]
config = FEC3FC5E46AE98299217D6885B3BE28C4F4D6FB9
[dbe://C:/1C DB/Another DB folder/?config,users]
config = FEC3FC5E46AE98299217D6885B3BE28C4F4D6FB9
users = A244BE3830C2B7075C0BB684896B97A0324984A0
[file://C:/Program Files (x86)/lcv82/8.2.9.356/docs/ru/V8UpdateFrom82Beta.htm]
= 392BC75149AE02565C7E31592EDCD60F00BFA03C
[ndir://C:/Program Files (x86)/lcv8/8.3.5.625/bin/?*]
lcv8.exe = 232F2BCCE6ABB95FD56E3CB3FADE528D851A5D69
lcv8_root.hbk = 025B4C8465D3CD851363B2102478B1CAF2EC444E
lcv8_root.res = 8F48869F82BB222EE25E18502605AC117BB5736A
lcv8_ru.hbk = 81CB0D5573D8517913053EA88AAADE9785AB443
...
```

[Копировать в буфер обмена](#)

13.5. Файл отчета

Результат работы утилиты фиксируется в файле отчета. Файл содержит информацию в формате UTF-8. Местоположение и имя файла указывается при запуске утилиты. Однако указанное имя будет преобразовано следующим образом: в имя файла будет добавлено дата и время формирования отчета, например, если указано, что отчета надо поместить в файл **c:\temp\report.rpt**, то фактическое имя файла будет иметь вид **c:\temp\report-14.02.26-14.07.58.019446.rpt**.

Сам файл имеет следующий формат:

```
[.params]
Key=value
[Нормализованный вид универсального описателя источника информации]
<Объект контроля> = <Обнаруженное изменение>
```

[Копировать в буфер обмена](#)

Секция **.params** присутствует всегда и содержит описание параметров запуска утилиты. В качестве значения **Key** могут выступать следующие значения:

- **datetime** – дата и время запуска утилиты;
- **workdir** – рабочий каталог при запуске утилиты;
- **exepath** – путь к исполняемому файлу с утилитой;
- **mode** – режим запуска утилиты:
 - **create** – режим формирования базы эталонов;
 - **check** – режим проверки эталонов.

- **etalon** – путь к базе эталонов;
- **report** – путь к файлу отчета (как он указан в командной строке запуска утилиты);
- **debug** – путь к файлу с отладочной информацией или пустая строка, если не задан вывод отладочной информации;
- **in** – путь к файлу с параметрами. Если в командной строке запуска указано более одного параметра **-in**, то возможно наличие нескольких ключей **in** в секции **.params**;
- **param** – все распознанные параметры командной строки запуска утилиты, разделенные запятыми.

Если в каком-либо источнике (из базы эталонов) обнаружены различия между базой эталонов и фактически состоянием файлов на диске, то в файл отчета помещается секция с описанием изменившегося источника и ниже выводятся обнаруженные изменения для объектов контроля. Если объект контроля не изменился – по такому объекту информации не выводится.

Обрабатываются следующие возможные изменения:

- **A** – в источнике информации обнаружен новый объект контроля;
- **D** – в источнике информации удален объект контроля, который существовал на момент формирования базы эталонов;
- **M** – объект контроля модифицирован.

Рассмотрим фрагмент файла отчета:

[Копировать в буфер обмена](#)

```
[dbe://C:/1C DB/DB folder/?config,users]
users = M
[ndir://C:/Program Files (x86)/1cv81/bin/?*]
1CmailV8.dll = D
1CmailV8.dll.32 = A
```

В данном файле отчета зафиксировано, что:

- Для файловой информационной базы, расположенной в каталоге **C:/1C DB/DB folder** изменено содержимое таблицы пользователей.
- В каталоге **C:/Program Files (x86)/1cv81/bin** удален файл **1CmailV8.dll** и обнаружен новый файл **1CmailV8.dll.32**.

13.6. Использование утилиты

Запуск утилиты осуществляется с помощью командной строки, имеющей следующие параметры:

[Копировать в буфер обмена](#)

```
ci <mode> --in <ШаблонИсточника> --etalon <ФайлБазыЭталонов> --report <ФайлОтчета> --debug <ФайлОтладочнаяИнформация> [описание источника]
```

Параметры имеют следующее значение:

mode

Режим работы утилиты:

- **--version, -v** – показать номер версии утилиты контроля целостности.
- **--help, -h** – показать справочную информацию по режимам **make** и **check**. В этом случае режим следует указать через пробел после команды получения справочной информации.
- **make** – сформировать файл базы эталонов или выполнить обновление данных по каким-либо источникам информации (если база эталонов существует, и в ней присутствуют данные по источнику, который указан при запуске утилиты).
- **check** – проверить целостность переданного списка источников по базе эталонов.

Следующие параметры необходимо указывать только при использовании утилиты в режимах **make** и **check**.

--in, -i

Параметр предназначен для указания пути к шаблону источников информации (см. [здесь](#)). Таких параметров может быть более одного.

Для режимов **make** и **check** следует указать хотя бы одно описание источника. Это может быть параметр **--in (-i)** или простое указание описания источника (см. ниже параметр [описание источника](#)).

--etalon, -e

Параметр предназначен для указания полного пути к файлу с базой эталонов (см. [здесь](#)).

Для режимов **make** и **check** является обязательным параметром.

--report, -r

Параметр предназначен для указания полного пути к файлу отчетом (см. [здесь](#)).

Для режимов **make** и **check** является обязательным параметром.

--debug, -d

Параметр предназначен для указания полного пути к файлу с отладочной информацией. Файл необходим для технического персонала фирмы «1С» в случае расследования некорректного поведения утилиты.

описание источника необязательный

Позволяет указать источник информации без указания файла шаблона (в том числе и несколько описаний). Для этого источник следует указывать точно в таком же формате, как и каждая строка списка шаблонов источников информации (см. [здесь](#)). В своей работе утилита использует все описания источников, которые обнаружила при своем запуске (как с помощью параметра **in**, так и с помощью простого указания).

13.7. Рекомендации по установке и использованию

Утилита контроля целостности не имеет функций самоконтроля. В связи с этим обеспечение неизменности файлов самой утилиты возложено на системного администратора. В качестве средства обеспечения можно использовать права доступа файловой системы на каталог с утилитой, файлы баз эталонов и файлы отчетов.

В качестве возможного сценария использования можно привести следующий вариант:

1. Утилита устанавливается в каталог, отличающийся от каталога установки системы «1С:Предприятие».
2. Для работы с утилитой создаётся привилегированный пользователь с ограниченными правами, однако, достаточными для работы со всеми объектами контроля в режиме чтения.
3. Права доступа к исполняемым и конфигурационным файлам утилиты ограничиваются созданным пользователем (т.е. любой доступ к файлам любых других пользователей должен быть запрещён).
4. Работать с утилитой следует от имени созданного пользователя.

Вышеперечисленные правила обеспечат высокий уровень защищённости утилиты от несанкционированного доступа и дискредитации.

В этом случае общая схема использования выглядит следующей:

1. Для запуска утилиты необходимо войти в систему от имени привилегированного пользователя. Если утилита запускается с помощью системного планировщика, необходимо обеспечить, чтобы запуск утилиты планировщиком также выполнялся от имени привилегированного пользователя.
2. Выполняется запуск утилиты в одном из необходимых режимов: создания базы эталонов или ее проверки целостности источников.
3. Выполняется анализ отчета о работе утилиты.

Таким образом, применение утилиты (и использование результатов работы утилиты) возможно только для ограниченного круга доверенных лиц, которые знают имя и пароль привилегированного пользователя, созданного для использования утилиты контроля целостности.

Приложение 1. Структура каталога установки и назначение каталогов и файлов

После установки на диске будет организована определенная структура каталогов, в которых расположены исполняемые и конфигурационные файлы системы. В данном разделе описана структура каталогов, а также назначение некоторых исполняемых и конфигурационных файлов.

1.1. Для ОС Windows

По умолчанию система будет установлена в каталог `C:\Program Files\1Cv8` (далее будем называть этот каталог **корневым каталогом установки**). Если клиентское приложение или 32-разрядный сервер «1С:Предприятия» устанавливаются на 64-разрядную ОС, то корневой каталог будет иметь следующее имя: `C:\Program Files (x86)\1Cv8`. В этом каталоге создаются остальные каталоги и конфигурационные файлы:

- **common** – данный каталог содержит общие файлы «1С:Предприятия». К ним относится программа запуска `1cestart`, программа установки драйвера ключа защиты, оснастка консоли управления для администрирования кластера серверов «1С:Предприятия» (`1CV8 Servers.msc`), библиотека пиктограммы файлов «1С:Предприятия» для нужд операционной системы.
- **conf** – данный каталог содержит конфигурационные файлы, необходимые для работы «1С:Предприятия».
- **srvinfo** – рабочий каталог центрального сервера. Содержит данные кластера серверов в том случае, если сервер «1С:Предприятия» установлен как сервис ОС Windows.
- **8.3.X.YYY** – данный каталог содержит файлы конкретной версии «1С:Предприятия». В данном имени **X** и **YYY** – номера конкретной установленной версии. Следует отметить, что программа установки позволяет устанавливать одновременно несколько версий системы программ «1С:Предприятие». В этом случае в корневом каталоге установки будет расположено несколько каталогов версий. Так, в случае установки версий 8.3.3.100 и 8.3.3.150 (номера версий – условные) будут присутствовать два каталога с именами `8.3.3.100` и `8.3.3.150`. Каждый каталог версии содержит в себе все файлы (кроме файла `1cestart`), присущие этой версии, а именно: собственно исполняемые файлы, сопроводительные файлы, лицензии к используемым программным продуктам и т. д. Структура каталога версии следующая:
 - **bin** – содержит исполняемые файлы версии (каталог исполняемых файлов).
 - **bin\conf** – содержит конфигурационные файлы конкретной версии или файл `conf.cfg`, который содержит путь к общему каталогу конфигурационных файлов (по умолчанию каталог `conf` корневого каталога установки). Описание файла `conf.cfg` см. [здесь](#).
 - **docs** – данный каталог содержит сопроводительные файлы на русском и английском языках. Состав файлов может меняться от версии к версии.
 - **licenses** – содержит лицензионное соглашение на «1С:Предприятие» на русском (файл `1CEnterprise_ru.htm`) и английском языках (файл `1CEnterprise_en.htm`), а также лицензионные соглашения на используемые программные компоненты сторонних производителей (этот список может меняться от версии к версии).
 - **readme** – данный каталог содержит файлы `readme` на языках локализации платформы.

Некоторые каталоги при установке всегда находятся в фиксированном месте файловой системы, вне зависимости от того, какой каталог выбран при установке системы:

- При установке клиентского приложения на 32-разрядную ОС каталоги `common` и `conf` располагаются в каталоге `%ProgramFiles%\1cv8`.
- При установке клиентского приложения на 64-разрядную ОС каталоги `common` и `conf` располагаются в каталоге `%ProgramFiles(x86)\1cv8`.

- При установке 32-разрядного сервера «1С:Предприятия» каталог [srvinfo](#) (дополнительно к каталогам [common](#) и [conf](#)) располагается:
 - на 32-разрядной ОС в каталоге [%ProgramFiles%\1cv8](#);
 - на 64-разрядной ОС в каталоге [%ProgramFiles\(x86\)%\1cv8](#).
- При установке 64-разрядного сервера «1С:Предприятия» каталог [srvinfo](#) (дополнительно к каталогам [common](#) и [conf](#)) располагается в каталоге [%ProgramFiles%\1cv8](#).

1.2. Для ОС Linux

Система будет установлена в каталог [/opt/1C/v8.3/i386](#) в случае 32-разрядной версии или в каталог [/opt/1C/v8.3/x86_64](#) в случае 64-разрядной версии (далее будем называть этот каталог **корневым каталогом установки**). В этом каталоге создаются остальные каталоги и конфигурационные файлы:

- [conf](#) – данный каталог содержит конфигурационные файлы, необходимые для работы «1С:Предприятия».
- [docs](#) – данный каталог содержит сопроводительные файлы на русском и английском языках. Состав файлов может меняться от версии к версии.
- [licenses](#) – содержит лицензионное соглашение на «1С:Предприятие» на русском (файл [1CEnterprise_ru.htm](#)) и английском языках (файл [1CEnterprise_en.htm](#)), а также лицензионные соглашения на используемые программные компоненты сторонних производителей (этот список может меняться от версии к версии).
- [readme](#) – данный каталог содержит файлы [readme](#) на языках локализации платформы.
- [utils](#) – содержит дополнительные утилиты, предназначенные для настройки сервера «1С:Предприятия».

1.3. Назначение каталогов и файлов

Данный раздел содержит описание некоторых каталогов и файлов, входящих в состав системы «1С:Предприятие».

1cestart

Программа запуска системы «1С:Предприятие».

С помощью программы запуска можно запустить все виды клиентов (толстый клиент, тонкий клиент, веб-клиент), конфигуратор.

СОВЕТ. Если программа запуска располагается в сетевом каталоге (см. [здесь](#)), то рекомендуется использовать эту программу из самой новой версии, которая планируется к установке из этого сетевого каталога.

Расположение файла:

- Для ОС Windows: в каталоге [common](#) корневого каталога установки.
- Для ОС Linux: в корневом каталоге установки.

1cv8s

Интерактивная программа запуска системы «1С:Предприятие» конкретной версии.

Может запустить все виды клиентов (толстый клиент, тонкий клиент, веб-клиент), конфигуратор.

Расположение файла:

- Для ОС Windows: в каталоге исполняемых файлов конкретной версии.
- Для ОС Linux: в корневом каталоге установки.

1cv8

Исполняемый файл толстого клиента или конфигуратора.

Не может запустить тонкий клиент и веб-клиент.

Расположение файла:

- Для ОС Windows: в каталоге исполняемых файлов конкретной версии.
- Для ОС Linux: в корневом каталоге установки.

1cv8c

Исполняемый файл тонкого клиента.

Расположение файла:

- Для ОС Windows: в каталоге исполняемых файлов конкретной версии.
- Для ОС Linux: в корневом каталоге установки.

ragent, rmngr, rphost

Исполняемые файлы сервера «1С:Предприятия» (подробнее см. книгу «1С:Предприятие 8.3. Клиент-серверный вариант. Руководство администратора»).

Расположение файла:

- Для ОС Windows: в каталоге исполняемых файлов конкретной версии.
- Для ОС Linux: в корневом каталоге установки.

webinst

Утилита настройки публикации веб-клиента на веб-сервере (подробнее см. [здесь](#)).

Расположение файла:

- Для ОС Windows: в каталоге исполняемых файлов конкретной версии.
- Для ОС Linux: в корневом каталоге установки.

<Номер версии>\bin\conf Только ОС Windows

Конфигурационные файлы конкретной версии системы «1С:Предприятие». Подробную информацию о поиске конфигурационных файлов см. [здесь](#).

\conf Только ОС Linux

Конфигурационные файлы системы «1С:Предприятие». Подробную информацию о поиске конфигурационных файлов см. [здесь](#).

chdbf1

Утилита тестирования базы данных файлового варианта (подробнее см. [здесь](#)).

Расположение файла:

- Для ОС Windows: в каталоге исполняемых файлов конкретной версии.
- Для ОС Linux: в корневом каталоге установки.

v7cnv.exe Только ОС Windows

Конвертер информационных баз из версии «1С:Предприятия 7.7» в текущую версию (подробнее см. книгу «1С:Предприятие 8.3. Руководство разработчика»).

Расположение файла:

- Для ОС Windows: в каталоге исполняемых файлов конкретной версии.
- Для ОС Linux: отсутствует.

RegMSC.cmd Только ОС Windows

Командный файл для регистрации утилиты администрирования кластера серверов «1С:Предприятия» конкретной версии (расположен в каталоге исполняемых файлов конкретной версии).

Расположение файла:

- Для ОС Windows: в каталоге исполняемых файлов конкретной версии.
- Для ОС Linux: отсутствует.

1ceunt.dll Только ОС Windows

Библиотека пиктограмм, которыми операционная система отображает различные типы файлов системы «1С:Предприятие». Данная библиотека является общей для всех версий системы. Регистрация этой библиотеки (и привязка пиктограмм к типам файлов) выполняется при первой установке на компьютере «1С:Предприятия». Отмена регистрации библиотеки (и удаление привязки пиктограмм к типам файлов) выполняется при удалении с компьютера последней версии «1С:Предприятия».

Расположение файла:

- Для ОС Windows: в каталоге **common** корневого каталога установки.
- Для ОС Linux: отсутствует.

1.4. Конфигурационные файлы: расположение и поиск

Конфигурационные файлы, которые используются при работе системы «1С:Предприятие» (**logcfg.xml**, **nethasp.ini** и т. д.), могут располагаться в различных местах файловой системы.

1.4.1. Для ОС Windows

В ОС Windows файлы могут располагаться в следующих местах (в порядке поиска):

- Каталог **bin\conf** конкретной версии, например, для версии 8.3.3.100, путь будет выглядеть так: **C:\Program Files\1Cv8\8.3.3.100\bin\conf**.
- Каталог **%USERPROFILE%\Local Settings\Application Data\1C\1Cv8\Conf** (**%LOCALAPPDATA%\1C\1Cv8\Conf** для ОС Windows Vista и старше) пользователя, под которым работает система.
- Каталог, который указан в файле **conf.cfg**, расположенном в каталоге **bin\conf** конкретной версии. Подробнее о формате файла **conf.cfg** см. [здесь](#).

- Каталог `%ALLUSERSPROFILE%\Application Data\1C\1cv8\Conf` (`%ALLUSERSPROFILE%\1C\1cv8\Conf` для ОС Windows Vista и старше) данных для всех пользователей компьютера.

ПРИМЕЧАНИЕ. При установке системы конфигурационные файлы записываются в каталог `C:\Program Files\1cv8\conf`, и данный путь записывается в файл `bin\conf\conf.cfg` устанавливаемой версии.

Такой порядок поиска конфигурационных файлов позволяет:

- Формировать единые конфигурационные файлы для всех версий и компонент, установленных на компьютере. Для этого конфигурационные файлы должны располагаться только в каталоге `C:\Program Files\1cv8\conf`.
- Формировать конфигурационные файлы отдельно для каждой версии, установленной на компьютере. Для этого конфигурационные файлы должны располагаться только в каталоге `bin\conf` конкретной версии.
- Формировать различные конфигурационные файлы для разных компонентов (для клиентского приложения и сервера «1С:Предприятия», функционирующего под другим пользователем системы) любой версии, работающих на компьютере. Для этого конфигурационные файлы должны располагаться только в каталогах `%USERPROFILE%\Local Settings\Application Data\1C\1cv8\Conf` (`%LOCALAPPDATA%\1C\1cv8\Conf` для ОС Windows Vista и старше) соответствующих пользователей.
- Использовать комбинации перечисленных способов для различных конфигурационных файлов.

1.4.2. Для ОС Linux

В ОС Linux файлы могут располагаться в следующих местах (в порядке поиска):

- Каталог `conf` установленной версии, например, для 32-разрядной версии «1С:Предприятия» путь к этому каталогу будет выглядеть следующим образом: `/opt/1C/v8.3/i386/conf`, а для 64-разрядной версии: `/opt/1C/v8.3/x86_64/conf`.
- Каталог `~/1cv8/1C/1cv8/conf` (~ – домашний каталог пользователя, от имени которого работает сервер «1С:Предприятия»).
- Каталог, который указан в файле `conf.cfg`, расположенном в каталоге `conf` установленной версии. Подробнее о формате файла `conf.cfg` см. [здесь](#).

Приложение 2. Описание элементов журнала регистрации

В данном приложении описывается структура журнала регистрации в случае выгрузки его в формате XML-файла.

Формат XML-документа выгрузки журнала регистрации:

- пространство имен: <http://v8.1c.ru/eventLog>;
- префикс пространства имен (по умолчанию): **v8e**.

EventLog

Корневой элемент документа. Содержит собственно события журнала регистрации (элементы **Event**).

Event

Содержит элементы, описывающие событие журнала регистрации.

Level

Тип: Перечисление уровня события. Значение уровня события.

Date

Тип: **ДатаВремя**. Значение даты и времени события.

Application

Тип: **Строка**. Имя приложения, в котором произошло событие.

ApplicationPresentation

Тип: **Строка**. Представление приложения, в котором событие произошло.

EventName

Тип: **Строка**. Имя произошедшего события.

EventPresentation

Тип: **Строка**. Представление произошедшего события.

UserID

Тип: **UUID**. Идентификатор пользователя, инициировавшего событие.

UserName

Тип: **Строка**. Имя пользователя, инициировавшего событие.

MetadataName

Тип: **Строка**. Составное имя в англоязычном варианте термов и с использованием имен метаданных. Для событий, предусматривающих наличие списка метаданных (события доступа к данным), содержит список элементов **Item**, в каждом из которых содержится имя объекта метаданных.

MetadataPresentation

Тип: **Строка**. Представление объекта метаданных на языке пользователя (синонимы). Для событий, предусматривающих наличие списка метаданных (события доступа к данным), содержит список элементов **Item**, в каждом из которых содержится представление объекта метаданных.

Comment

Тип: **Строка**. Комментарий к событию.

Data

Тип: **Произвольный**. Данные события. Если тип данных не может быть представлен в виде XML, то производится запись значения Неопределено (пустой элемент с атрибутом **xsi:nil = "true"**).

Может содержать структуры, массивы и таблицы (для событий доступа к данным, аутентификации и работы с пользователями информационной базы).

DataPresentation

Тип: **Строка**. Представление данных события.

TransactionStatus

Тип: Статус транзакции для события. Статус транзакции может принимать следующие значения:

- **InProgress** – транзакция не завершена;
- **Committed** – транзакция зафиксирована;
- **RolledBack** – транзакция отменена;
- **NotApplicable** – запись выполняется вне транзакции.

TransactionID

Тип: **Строка**. Идентификатор транзакции.

Connection

Тип: **Число**. Номер соединения.

Session

Тип: **Число**. Номер сеанса.

ServerName

Тип: **Строка**. Имя рабочего сервера.

Port

Тип: **Число**. Основной IP порт.

SyncPort

Тип: **Число**. Вспомогательный IP порт.

SessionDataSeparation

Содержит коллекцию элементов, описывающих разделители, которые использовались в сеансе во время регистрации события. Имя элемента соответствует имени разделителя, значение элемента содержит значение разделителя.

Содержит коллекцию элементов `Item`, со значениями типа `Строка`, содержащими представления разделителей в том же порядке, что и в элементе `SessionDataSeparation`.

Приложение 3. Описание и расположение служебных файлов

Данное приложение содержит описание и расположение различных служебных файлов, которые могут использоваться при работе «1С:Предприятия».

3.1. *.1ccr

Конфигурационный файл Web-сервиса для работы с удаленным хранилищем может иметь произвольное имя (расширение **1ccr** обязательно), формат XML и содержит единственный узел с произвольным именем и атрибутом `connectString` – в этом атрибуте указывается адрес сервера хранилища в схеме tcp.

Например, такой конфигурационный файл может иметь имя `repository.1ccr` и следующее содержание:

```
<?xml version="1.0" encoding="UTF-8"?>
<repository connectString="tcp://RepServ"/>
```

[Копировать в буфер обмена](#)

В данном случае в качестве произвольного имени узла выбрано имя `repository`, а адрес сервера хранилища конфигурации – `tcp://RepServ`.

3.2. *.mft

Файл с расширением **mft** является файлом-манифестом – специальным файлом, описывающим шаблон конфигурации. Файл может иметь произвольное имя.

Файл располагается в каталоге установленного шаблона конфигурации.

Файл-манифест имеет произвольное имя и расширение **mft**. Внутренний формат файла-манифеста близок к формату ini-файла. Для поддержки нескольких языков файл-манифест использует кодировку UTF-8. В начале файла-манифеста указываются следующие параметры.

Vendor

Поставщик решения. Совпадает с указанным в конфигурации.

Name

Наименование решения. Совпадает с указанным в конфигурации.

Version

Версия решения. Совпадает с указанной в конфигурации.

AppVersion

Версия «1С:Предприятия», с помощью которой создавался комплект поставки.

Следующие параметры относятся к частям решения и разделяются наименованиями секций. Наименования секций выбираются произвольно и заключаются в квадратные скобки.

Source

Относительный путь к файлу конфигурации (**cf**), файлу обновления (**cfu**) или выгрузки базы данных (**dt**).

Catalog_<суффикс языка>

Наименование решения в каталоге решений. Параметров `Catalog_<суффикс языка>` в файле-манифесте может быть несколько. Суффикс определяет язык пользовательского интерфейса **1С:Предприятия 8** (например, **ru** для указания русского языка). Если суффикс языка не указан (имя параметра задано как `Catalog`), то значение этого параметра используется для всех пользовательских интерфейсов, кроме тех, для которых в данной секции указан параметр `Catalog` с нужным языковым суффиксом.

Destination

Рекомендуемый каталог создания информационной базы. Этот параметр используется при создании информационной базы из шаблона. Каталог представляет частичный путь. В качестве одной из частей каталог должен включать в себя каталог поставщика (во избежание совпадения имен каталогов у различных решений).

В каталоге шаблона конфигурации допускается наличие нескольких **cfu**-файлов.

[Копировать в буфер обмена](#)

```
Vendor=Фирма "1С"
Name=Управление Торговлей
Version=8.10.0.2
[Main]
Source=Main\lcv8.cf
Catalog_ru=Управление торговлей/Управление торговлей
Destination=1C\Trade
[Demo]
Source=Demo\lcv8.dt
Catalog_ru=Управление торговлей/Управление торговлей (Демо)
Destination=1C\DemoTrd
```

3.3. *.v8i

В данном приложении приводится описание формата файла описаний зарегистрированных информационных баз. Этот список используют все клиенты. По умолчанию файл имеет имя **ibases.v8i**.

Расположение файла:

- Для ОС Windows: в каталоге `%APPDATA%\1C\1CStart\` локального компьютера.
- Для ОС Linux: в каталоге `~\1C\1cstart`.

Файл представляет собой текстовый документ в кодировке UTF-8 и состоит из секций. Каждая секция описывает одну информационную базу.

Файл состоит из секций описаний информационных баз.

Секция описания информационной базы:

```
[Наименование секции]
Connect=
ID=
OrderInList=
Folder=
OrderInTree=
External=
ClientConnectionSpeed=
App=
DefaultApp=
WA=
```

[Копировать в буфер обмена](#)


```

WSA=
Version=
DefaultVersion=
AdditionalParameters=
WebCommonInfoBaseURL=
HttpsCA=
HttpsCert=
HttpsCAFile=
HttpsCertFile=
HttpsCertSelect=
HttpsNSSPath=
HttpsNSSCertName=

```

Секция состоит из наименования секции и параметров.

Наименование и каждый параметр секции записываются в отдельной строке файла описаний.

Наименование секции

Наименование секции совпадает с наименованием информационной базы и является обязательным параметром. Наименование заключается в квадратные скобки.

Параметр может редактироваться в окне свойств информационной базы.

Пример:

```
[Демонстрационнаяверсия8.2]
```

[Копировать в буфер обмена](#)

ID

Внутренний идентификатор информационной базы. Формируется автоматически. Должен быть уникальным в рамках одного файла **v8i**.

Формирование идентификатора вручную не рекомендуется.

Пример:

```
ID=cf9f0d4b-b4a3-11d8-861e-0050baaa2f3f
```

[Копировать в буфер обмена](#)

Connect необязательный

Строка соединения с информационной базой. Может существовать несколько описаний информационных баз, которые имеют одинаковую строку запуска (но разное наименование). Это может потребоваться, когда необходимо запускать одну базу в нескольких режимах запуска (например, тонкий и толстый клиенты), не меняя свойства информационной базы.

Пример:

Файловый вариант указывается в виде:

```
Connect=File=<Путь>;
```

[Копировать в буфер обмена](#)

Клиент-серверный вариант указывается в виде:

```
Connect=Srvr=<ИмяСервера1СПредприятия>;Ref=<ИмяИнформационнойБазыНаСервере>;
```

[Копировать в буфер обмена](#)

Folder необязательный

Наименование папки в дереве информационных баз.

ПРИМЕЧАНИЕ. Если наименование папки не указано или параметр опущен, то данная информационная база располагается в корне списка информационных баз.

Пример:

```
Folder=/Торговыебазы
```

[Копировать в буфер обмена](#)

OrderInList

Порядок в списке при представлении списком. Задается числом, которое является значением порядка информационной базы в списке баз (сортировка по наименованию не установлена).

Пример:

```
OrderInList=5
```

[Копировать в буфер обмена](#)

OrderInTree

Порядок в ветви при представлении деревом.

Пример:

```
OrderInTree=16358
```

[Копировать в буфер обмена](#)

UseProxy необязательный

Указывает на вариант использования прокси-сервера для варианта ws-соединения.

- **0** – прокси-сервер не используется;
- **1** – автоматическое определение настроек прокси-сервера;
- **2** – явное указание настроек прокси-сервера.

Если **UseProxy** не указан, то используется автоматическое определение настроек прокси-сервера. Для файлового и клиент-серверного варианта не имеет смысла.

Пример:

```
UseProxy=1
```

[Копировать в буфер обмена](#)

PSrv

Строка, содержащая адрес прокси-сервера (обязателен только в том случае, когда параметр [UseProxy](#) имеет значение 2).

Пример:

```
Proxy=192.168.0.1
```

[Копировать в буфер обмена](#)

Port

Номер порта прокси-сервера (обязателен только в том случае, когда параметр [UseProxy](#) имеет значение 2).

Пример:

```
Port=123
```

[Копировать в буфер обмена](#)

User *необязательный*

Имя пользователя прокси-сервера.

Пример:

```
User=username
```

[Копировать в буфер обмена](#)

Password *необязательный*

Зашифрованный пароль для прокси-сервера.

Пример:

```
Password=XNKxbVbqnxUCwklUrovbo7bZFrG/ZpF6cQ10qVtzpk=
```

[Копировать в буфер обмена](#)

ClientConnectionSpeed

Скорость клиентского соединения (имеет смысл только для тонкого и веб-клиентов). Принимает значения:

- **Normal** – обычная скорость соединения;
- **Low** – низкая скорость соединения.

Если параметр не указан, то скорость клиентского соединения будет определяться значением флажка [Низкая скорость соединения](#) окна запуска (что эквивалентно значению [Выбирать при запуске](#) параметра [Скорость соединения](#) окна с параметрами запуска информационной базы, см. [здесь](#)).

Пример:

```
ClientConnectionSpeed=Low
```

[Копировать в буфер обмена](#)

WA

Определяет вариант аутентификации пользователя. Принимает значения:

- **1** – попытка выполнить аутентификацию средствами ОС. Если выполнено неудачно, запрашивается логин/пароль.
- **0** – всегда использовать аутентификацию с помощью логина/пароля.

Пример:

```
WA=1
```

[Копировать в буфер обмена](#)

WSA

Определяет вариант аутентификации пользователя на веб-сервере, если веб-сервер используется в качестве промежуточного звена (тонкий клиент, подключенный через веб-сервер и веб-клиент). Принимает значения:

- **1** – попытка выполнить аутентификацию на веб-сервере средствами ОС. Если выполнено неудачно, запрашивается логин/пароль.
- **0** – всегда запрашивать логин/пароль.

Пример:

```
WSA=1
```

[Копировать в буфер обмена](#)

App

Определяет тип клиентского приложения:

- **Auto** – автоматический выбор типа клиентского приложения;
- **ThinClient** – тонкий клиент;
- **ThickClient** – толстый клиент;
- **WebClient** – веб-клиент.

Параметр может редактироваться в окне свойств информационной базы.

Пример:

```
App=Auto
```

[Копировать в буфер обмена](#)

DefaultApp

Тип клиента, который определяется и помещается в данный файл программой запуска при автоматическом определении типа клиентского приложения (ключ [/AppAutoCheckMode](#)):

- **ThinClient** – тонкий клиент;
- **ThickClient** – толстый клиент.

Если значение параметра `App` равно `Auto` и параметр `DefaultApp` не указан, выполняется запуск тонкого клиента с параметром командной строки `/AppAutoCheckMode`.

Если параметр `DefaultApp` задан, то выполняется запуск указанного в нем клиента с указанием параметра `/AppAutoCheckMode`.

Пример:

```
DefaultApp=ThinClient
```

[Копировать в буфер обмена](#)

`Version`

Версия «1С:Предприятия», которая должна использоваться для запуска информационной базы.

Параметр может редактироваться в окне свойств информационной базы.

Пример:

```
Version=8.3.3
```

[Копировать в буфер обмена](#)

`DefaultVersion`

Версия системы «1С:Предприятие», которая реально использовалась при последнем запуске данной информационной базы. Автоматически определяется и помещается в данный файл программой запуска, если выполняется запуск с параметром `/AppAutoCheckVersion`.

При последующих запусках будет использоваться именно эта версия, а не та, которая указана в параметре `Version`.

Пример:

```
DefaultVersion=8.3.3.100
```

[Копировать в буфер обмена](#)

`AdditionalParameters`

Содержит дополнительные параметры запуска, которые могут быть введены в окне свойств информационной базы, в элементе [Дополнительные параметры запуска](#).

Пример:

```
AdditionalParameters=/DisplayAllFunctions /LogUI
```

[Копировать в буфер обмена](#)

`WebCommonInfoBaseURL`

Если информационная база добавлена в текущий список с помощью Интернет-сервиса получения списка общих информационных баз (см. [здесь](#)), то в данном параметре будет размещен адрес сервиса, предоставившего информацию об информационной базе.

Если при интерактивном запуске интерактивной программы запуска (`1cv8s`) обнаруживается, что список общих информационных баз, полученных с помощью Интернет-сервиса, не требует обновления, то описания всех информационных баз (вызов Web-сервиса `WebCommonInfoBases.CheckInfoBases()` вернул параметр `InfoBaseChanged` равный значению `Ложь`), которые получены из этого источника, остаются в списке до следующего запуска.

Если из файла `1cestart.cfg` удалены параметры `InternetService` или `WebCommonInfoBases`, то из списка информационных баз будет удалена информация об информационных базах, полученных из удаленных источников.

Пример:

```
WebCommonInfoBaseURL=http://info-server/listservice
```

[Копировать в буфер обмена](#)

`HttpsCA` *необязательный*

Тип источника сертификатов удостоверяющих центров, используемых для проверки серверного сертификата. Может принимать следующие значения:

- `None` – сертификаты удостоверяющих центров не используются, проверка серверного сертификата не осуществляется;
- `File` – сертификаты удостоверяющих центров находятся в файле;
- `Windows` – сертификаты удостоверяющих центров находятся в системном хранилище сертификатов OC Windows;
- `NSS` – сертификаты удостоверяющих центров находятся в хранилище сертификатов NSS.

`HttpsCert` *необязательный*

Тип источника клиентского сертификата и его закрытого ключа. Может принимать следующие значения:

- `None` – клиентский сертификат не используется;
- `File` – клиентский сертификат находится в файле;
- `Windows` – клиентский сертификат находится в системном хранилище сертификатов OC Windows;
- `NSS` – клиентский сертификат находится в хранилище сертификатов NSS.

Если в качестве значения параметра `HttpsCA` установлено `NSS`, то параметр `HttpsCert` может принимать только значение `NSS`. Другие значения будут означать ошибку.

Если значение данного параметра равно `NSS`, то:

- параметр `HttpsCA` может принимать только значения `None` и `NSS`;
- если значение параметра `HttpsCA` равно `File` или `Windows`, то значение будет заменено на значение `None`, и проверка серверного сертификата не осуществляется.

`HttpsCAFile` *необязательный*

Путь к файлу, который содержит сертификаты удостоверяющих центров. Если параметр `HttpsCA` установлен в значение `File`, а данный параметр отсутствует или равен пустой строке, то считается, что параметр `HttpsCA` установлен в значение `None`.

`HttpsCertFile` *необязательный*

Путь к файлу, который содержит клиентский сертификат и его закрытый ключ. Если параметр `HttpsCert` установлен в значение `File`, а данный параметр отсутствует или равен пустой строке, то считается, что параметр `HttpsCert` установлен в значение `None`.

`HttpsCertSelect` *необязательный*

Способ выбора клиентского сертификата OC Windows, в случае если установлено более одного подходящего для данного соединения сертификата. Может принимать

следующие значения:

- **Recent** – использовать выбранный, если есть запомненный – используется он, если нет – открывается диалог выбора и выбранный сертификат запоминается для использования в дальнейшем;
- **Choose** – всегда выбирать сертификат. Выбранный сертификат запоминается, и в дальнейшем может использоваться, если данный параметр будет установлен в значение **Recent**;
- **Auto** – автоматически подбирать подходящий сертификат для данного соединения. Диалог выбора не открывается.

HttpsNSSPath **необязательный**

Путь каталога хранилища сертификатов NSS. Если параметр **HttpsCert** установлен в значение **NSS**, а данный параметр отсутствует или равен пустой строке, то считается, что параметры **HttpsCA** и **HttpsCert** установлен в значение **None**.

HttpsNSSCertName **необязательный**

Имя конкретного используемого клиентского сертификата в хранилище сертификатов NSS.

3.4. 1cescmn.cfg

Файл **1cescmn.cfg** содержит общие настройки программ запуска (**1cestart** и **1cv8s**).

Файл располагается в каталоге, откуда выполняется установка системы в случае сетевого каталога расположения дистрибутивов (см. [здесь](#)).

ПРИМЕЧАНИЕ. Применяется только для системы «1С:Предприятие» под ОС Windows.

Файл представляет собой текстовый документ в кодировке UTF-8 или UTF-16LE и содержит строки формата **<Параметр>=<Значение>**.

Описание файла эквивалентно описанию файла **1cestart.cfg** (см. [здесь](#)) с тем исключением, что общий конфигурационный файл не может содержать строку с параметром **CommonCfgLocation**.

Пример:

```
CommonInfoBases=ibcommon.v8i
DistributiveLocation=\\server\v8dst
```

[Копировать в буфер обмена](#)

В данном примере задается имя файла со списком общих информационных баз (**ibcommon.v8i**), который должен располагаться в том же каталоге, что и файл с интерактивной программой запуска (**1cestart**). Также задается каталог расположения дистрибутивов версий системы: **\\server\v8dst**.

3.5. 1cestart.cfg

Файл **1cestart.cfg** содержит настройки, которые используют программы запуска (**1cestart** и **1cv8s**), клиентские приложения (**1cv8** и **1cv8c**) и внешнее соединение.

Расположение файла:

- Для ОС Windows:
 - **%APPDATA%\1C\1CEstart** – для конкретного пользователя. Файл изменяется при выполнении настроек окна запуска (см. [здесь](#)).
 - **%ALLUSERSPROFILE%\Application Data\1C\1CEstart** (**%ALLUSERSPROFILE%\1C\1CEstart** для ОС Windows Vista и выше) – для всех пользователей компьютера. Файл изменяется только во время установки системы «1С:Предприятие».
- Для ОС Linux: **~/1C/1cestart**.

Файл представляет собой текстовый документ в кодировке UTF-16LE и содержит строки формата **<Параметр>=<Значение>**. Далее приведены описания параметров, которые могут содержаться в этом файле.

Пример:

```
DefaultVersion=8.2-8.2.16
DefaultVersion=8.3-8.3.3
CommonCfgLocation=\\Server\v8\1cescmn.cfg
CommonInfoBases=\\Server\common\common_dblast.v8i
InstalledLocation=C:\Program Files\1cv8
DistributiveLocation=\\server\dst1C\v8
ConfigurationTemplatesLocation=\\server\tplmts
ConfigurationTemplatesLocation=C:\Documents and Settings\User\Application Data\1C\1Cv8\tplmts
InstallComponents=DESIGNERALLCLIENTS=1 THINCLIENT=1 WEBSEVEREXT=1 SERVER=1 CNFRPEOSSERVER=0 CONVERTER77=1 SERVERCLIENT=1 LANGUAGES=RU
UseHwLicenses=0
AppAutoInstallLastVersion=0
```

[Копировать в буфер обмена](#)

DefaultVersion

Данный параметр определяет версию, используемую по умолчанию. Допускается наличие нескольких строк с таким параметром.

Используются значения из всех конфигурационных файлов.

Пример 1:

```
DefaultVersion=8.3-8.3.3.100
```

[Копировать в буфер обмена](#)

Данная строка означает, что при попытке запуска информационной базы с указанием версии **8.3** будет использоваться версия **8.3.3.100**.

Пример 2:

```
DefaultVersion=8.2.15-8.2.15.315
```

[Копировать в буфер обмена](#)

Данная строка означает, что при попытке запуска информационной базы с версией **8.2.15** будет использоваться версия **8.2.15.315**.

CommonInfoBases

Параметр указывает путь и имя файла со списком общих информационных баз.

Используются значения из всех конфигурационных файлов.

Описание формата файла со списком информационных баз см. [здесь](#).

InstalledLocation

Параметр содержит указание на каталог, в который выполнена установка «1С:Предприятие». По умолчанию это значение `C:\Program Files\1Cv8`.

Используются значения из всех конфигурационных файлов в следующем порядке:

- из общего конфигурационного файла;
- из локального конфигурационного файла для всех пользователей;
- из локального конфигурационного файла.

ПРИМЕЧАНИЕ. Не рекомендуется использовать данный параметр в общем конфигурационном файле (`1cescmn.cfg`).

DistributiveLocation

Параметр содержит указание на каталог, в котором будет производиться поиск новой версии для автоматической установки.

Используются значения из всех конфигурационных файлов.

Поиск каталога с дистрибутивами новых версий будет также выполняться в каталоге, где расположен общий конфигурационный файл (`1cescmn.cfg`).

CommonCfgLocation

Параметр указывает путь и имя общего конфигурационного файла. Допускается наличие нескольких строк с таким параметром.

Используются значения из всех конфигурационных файлов.

ПРИМЕЧАНИЕ. Не рекомендуется использовать данный параметр в общем конфигурационном файле (`1cescmn.cfg`).

InstallComponents

В локальном конфигурационном файле и локальном конфигурационном файле для всех пользователей (`1cestart.cfg`) содержится перечень установленных компонент.

В общем конфигурационном файле (`1cescmn.cfg`) содержится перечень компонент, которые необходимо установить (формируется администратором системы).

Используется значение параметра из одного конфигурационного файла в соответствии со следующим приоритетом:

- локальный конфигурационный файл для всех пользователей;
- локальный конфигурационный файл;
- общий конфигурационный файл.

Параметр содержит строку компонентов с признаком необходимости установки, разделенных пробелом:

- `0` – не устанавливать;
- `1` – устанавливать.

Возможны следующие компоненты:

- **DESIGNERALLCLIENTS** – все клиенты и конфигуратор.
- **THINCLIENT** – тонкий клиент для клиент-серверного варианта работы.
- **THINCLIENTFILE** – тонкий клиент с возможностью работы с файловыми информационными базами.
- **SERVER** – сервер «1С:Предприятия». Если программа установки запускается из программы запуска, то сервер будет установлен как приложение.
- **WEBSERVEREXT** – компоненты расширения для веб-сервера.
- **CONFREPOSSERVER** – сервер хранилища конфигураций «1С:Предприятия».
- **SERVERCLIENT** – компоненты для администрирования кластера серверов «1С:Предприятия».
- **CONVERTER77** – конвертер информационных баз из версии «1С:Предприятия 7.7».
- **LANGUAGES** – список языков интерфейса для установки. Если указано несколько языков, они перечисляются через «,». Перечень кодов языков локализации см. [здесь](#).

ПРИМЕЧАНИЕ. Язык с кодом **EN** будет установлен даже в том случае, если он не указан в параметре **LANGUAGES** или не указан параметр **LANGUAGES**.

Пример:

```
LANGUAGES=RU,UK,BG
```

[Копировать в буфер обмена](#)

Пример параметра:

```
InstallComponents=DESIGNERALLCLIENTS=0 THINCLIENT=1 WEBSERVEREXT=0 SERVER=0 CONFREPOSSERVER=0 CONVERTER77=0 SERVERCLIENT=1 LANGUAGES=RU,EN
```

[Копировать в буфер обмена](#)

ConfigurationTemplatesLocation

Параметр указывает путь к каталогу шаблонов конфигураций. Может быть более одной записи.

Используются значения из всех конфигурационных файлов.

UseHwLicenses

Параметр управляет поиском ключа защиты при запуске «1С:Предприятия»:

- `1` – поиск ключа защиты выполняется (значение по умолчанию);
- `0` – поиск ключа защиты не выполняется.

Используется значение параметра из одного конфигурационного файла в соответствии со следующим приоритетом:

- локальный конфигурационный файл;
- локальный конфигурационный файл для всех пользователей;
- общий конфигурационный файл.

Данный параметр позволяет отключить поиск ключа защиты в тех случаях, когда получение клиентских лицензий реализовано с помощью расширения веб-сервера, сервера «1С:Предприятия» или в случае базовой версии.

Значение параметра может изменяться системой в следующих случаях:

- Если поиск ключа защиты включен, то при старте клиентского приложения выполняется анализ времени поиска ключа защиты. Если ключ защиты не был найден, запуск выполнен успешно и время поиска превысило 3 секунды, пользователю предлагается отключить поиск ключа защиты для ускорения последующих запусков. В случае согласия пользователя в файл `1cstart.cfg` этого пользователя выполняется запись параметра `UseHwLicenses=0`.
- Если поиск ключа защиты отключен и при запуске обнаруживается, что лицензия не получена от сервера «1С:Предприятия» или расширения веб-сервера, то пользователю предлагается включить поиск ключа защиты. В случае согласия пользователя в файл `1cstart.cfg` этого пользователя выполняется запись параметра `UseHwLicenses=1` и выполняется перезапуск клиентского приложения.

Если осуществляется запуск внешнего соединения, то выполняется попытка анализа параметра из файла `1cstart.cfg`, расположенного в профиле пользователя, от имени которого запускается внешнее соединение. Если у пользователя отсутствует профиль (например, пользователь `LocalSystem` в ОС Windows), то поиск ключа выполняется всегда.

InternetService

URL интернет-сервиса, который предоставляет список общих информационных баз и дистрибутив клиентского приложения.

Вначале выполняется попытка получения необходимого файла (со списком общих информационных баз или дистрибутивом клиентского приложения) с помощью HTTP-запроса, если эта попытка оказалась неудачной – выполняется попытка получения файла с помощью Web-сервиса.

При этом для HTTP-запроса полный URL сервиса формируется следующим образом: `<Адрес из параметра InternetService>/<Имя сервиса>/<Имя метода>/?<Параметры метода>`.

Для запроса с помощью Web-сервиса адрес описания (в формате WSDL) формируется следующим образом: `<Адрес из параметра InternetService>/<Имя сервиса>/?wsdl`.

WebCommonInfoBases

URL интернет-сервиса, который предоставляет список общих информационных баз.

Вначале выполняется попытка получения списка общих информационных баз с помощью HTTP-запроса, если эта попытка оказалась неудачной – выполняется попытка получения файла с помощью Web-сервиса.

При этом для HTTP-запроса полный URL сервиса формируется следующим образом: `<Адрес из параметра WebCommonInfoBases>/<Имя сервиса>/<Имя метода>/?<Параметры метода>`.

Для запроса с помощью Web-сервиса адрес описания (в формате WSDL) формируется следующим образом: `<Адрес из параметра WebCommonInfoBases>/<Имя сервиса>/?wsdl`.

Если указан и параметр `InternetService` и параметр `WebCommonInfoBases`, то вначале используется адрес, указанный в параметре `WebCommonInfoBases`, и в случае неудачи – адрес, указанный в параметре `InternetService`.

WebDistributiveLocation

URL интернет-сервиса, который предоставляет дистрибутив клиентского приложения.

Вначале выполняется попытка получения дистрибутива клиентского приложения с помощью HTTP-запроса, если эта попытка оказалась неудачной – выполняется попытка получения файла с помощью Web-сервиса.

При этом для HTTP-запроса полный URL сервиса формируется следующим образом: `<Адрес из параметра WebDistributiveLocation>/<Имя сервиса>/<Имя метода>/?<Параметры метода>`.

Для запроса с помощью Web-сервиса адрес описания (в формате WSDL) формируется следующим образом: `<Адрес из параметра WebDistributiveLocation>/<Имя сервиса>/?wsdl`.

Если указан и параметр `InternetService` и параметр `WebDistributiveLocation`, то вначале используется адрес, указанный в параметре `WebDistributiveLocation`, и в случае неудачи – адрес, указанный в параметре `InternetService`.

AppAutoInstallLastVersion

Определяет необходимость автоматической установки новой версии «1С:Предприятия»:

- 0 – выключить автоматическую установку новой версии;
- 1 – включить автоматическую установку новой версии (значение по умолчанию).

Используется значение параметра из одного конфигурационного файла в соответствии со следующим приоритетом:

- локальный конфигурационный файл;
- локальный конфигурационный файл для всех пользователей;
- общий конфигурационный файл.

Если на локальном компьютере не установлена версия, которая требуется сервером в клиент-серверном варианте использования или явно указана для информационной базы, то значение ключа (из конфигурационных файлов или командной строки) `AppAutoInstallLastVersion` игнорируется и будет выполнена попытка установить новую версию.

3.6. 1cv8wsrv.lst

Файл хранится на компьютере рабочего сервера, отмеченного как центральный, в каталоге служебных файлов кластера и содержит список кластеров, зарегистрированных на данном компьютере сервера «1С:Предприятия». Содержащиеся в нем данные необходимы для нормальной работы приложений, использующих данный сервер «1С:Предприятия».

Пример:

```
C:\Program Files\1cv8\srvinfo\1cv8wsrv.lst
```

[Копировать в буфер обмена](#)

3.7. 1CV8C1st.lst

Файл расположен в каталоге данных каждого рабочего сервера, отмеченного как центральный.

Файл содержит реестр кластера и хранит следующую информацию:

- список информационных баз, зарегистрированных в данном кластере;
- список рабочих серверов, входящих в кластер;
- список рабочих процессов, входящих в кластер;

- список менеджеров кластера;
- список сервисов кластера;
- список администраторов кластера.

Пример:

```
C:\Program Files\lcv8\srvinfo\reg_1541\lcv8Clst.lst
```

[Копировать в буфер обмена](#)

3.8. admininstall.cfg

Файл [admininstall.cfg](#) указывает на то, что установка системы программ «1С:Предприятие» выполнялась с использованием средств администрирования ОС Windows.

ПРИМЕЧАНИЕ. Применяется только для системы «1С:Предприятие» под ОС Windows.

Файл располагается в каталоге конфигурационных файлов системы «1С:Предприятие» и представляет собой текстовый документ в кодировке UTF-8.

В файле может располагаться единственная строка, определяющая вариант установки:

```
AdmInstall=<Mode>
```

[Копировать в буфер обмена](#)

```
<Mode>
```

Описывает режим установки:

- **Logon** – установка выполнена с помощью logon-скрипта во время входа пользователя в домен.
- **Restart** – установка выполнена с помощью групповых политик.

Ниже приведен пример скрипта установки, который может использоваться для установки системы «1С:Предприятие» с помощью административных средств ОС Windows (см. [здесь](#)).

[Копировать в буфер обмена](#)

```
Option Explicit
' изменять интерфейс пользователя
Const msiUILevelNoChange = 0
'Использовать интерфейс пользователя, заданный по умолчанию
Const msiUILevelDefault = 1
'Не отображать интерфейс пользователя (молчаливая установка)
Const msiUILevelNone = 2
'Только индикатор прогресса и отображение ошибок
Const msiUILevelBasic = 3
'Интерфейс пользователя без диалоговых сообщений
Const msiUILevelReduced = 4
'Полный интерфейс пользователя
Const msiUILevelFull = 5
'Если используется с msiUILevelBasic, то отображается индикатор
'прогресса без кнопки Cancel
Const msiUILevelHideCancel = 32
'Если используется с msiUILevelBasic, то отображается индикатор
'прогресса без отображения любых диалогов, в т. ч. и ошибок.
Const msiUILevelProgressOnly = 64
'Если используется с любым из перечисленных значений, инсталлятор
'выводит сообщение в конце установки о конечном результате.
Const msiUILevelEndDialog = 128
'***** Необходимо изменить на реальный каталог установки
Const DistrFolder="\\Server\lCDistr\"
Const shortcutName = "Запуск 1С Предприятия"
Dim shortcutTarget : shortcutTarget = DistrFolder & "lcestart.exe"
'Константы для определения действия
' требуется установка
Const requiredInstall = 1
' требуется удаление
Const requiredUninstall = 0
'Значение параметра ProductCode из файла setup.ini ...
'... для удаляемой версии
Const unInstallUID="{9173B91C-FF56-4F25-82D1-7F68244044CD}"
'... для устанавливаемой версии
Const InstallUID="{0BC98727-04AD-470F-9E8E-0162C543833F}"
'процедура установки или удаления указанной версии продукта
Sub installOrUninstall (ByVal productCode, ByVal msiPackage, ByVal mstTransform, ByVal requiredAction)
'productCode - Информация о коде продукта. Находится в файле
' setup.ini, ключ ProductCode
'msiPackage - Установочный пакет lCEnterprise
'mstTransform - файл языкового преобразования для инсталлятора
'requiredAction - требуемое действие requiredInstall или
' requiredUninstall
'Переменная для формирования дополнительных
' параметров инсталлятору
Dim cmdLine
On Error Resume Next
Dim installer, session
Set installer = Nothing
Set session = Nothing
Set installer = Wscript.CreateObject("WindowsInstaller.Installer") : processError
installer.UILevel = msiUILevelBasic 'msiUILevelNone 'или укажите другой вариант интерфейса пользователя
'проверка установки продукта
Set session = installer.OpenProduct(productCode)
If session Is Nothing AND requiredAction = requiredInstall Then
'продукт не установлен, и требуется его установка
cmdLine = "TRANSFORMS=admininstallrelogon.mst;"
If Not mstTransform Is Empty Then
'добавим указание инсталлятору разговаривать на указанном языке
cmdLine = cmdLine & mstTransform
'можно дополнительно указать, какие компоненты нужно установить
'cmdLine = cmdLine & " DESIGNERALLCLIENTS=1 THINCLIENT=1 WEBSEVEREXT=0 SERVER=0 CONFREPOSSERVER=0 CONVERTER77=0 SERVERCLIENT=1 LANGUAGES=RU"
End If
'установим платформу
Set session = installer.InstallProduct(msiPackage, cmdLine) : processError
'создадим ярлык на рабочем столе
createShortcut()
ElseIf Not session Is Nothing AND requiredAction = requiredUninstall Then
'платформа уже установлена, и требуется ее удаление
'объект session может быть только один!
Set session = Nothing
'укажем, что нужно удалить с компьютера пользователя эту версию
cmdLine = "REMOVE=ALL"
'удалим
```

```

    Set session = installer.InstallProduct(msiPackage, cmdLine) : processError
End If
Set session = Nothing
Set installer = Nothing
End Sub

'обработка ошибок
Sub processError
    Dim msg
    If Err = 0 Then Exit Sub
    msg = Err.Source & " " & Hex(Err) & ": " & Err.Description
    Wscript.Echo msg
    Wscript.Quit 2
End Sub

'создание ярлыка
Sub createShurtcut
    Dim WshShell, oShellLink
    Set WshShell = WScript.CreateObject("WScript.Shell")
    Dim strDesktop : strDesktop = WshShell.SpecialFolders("Desktop")
    Set oShellLink = WshShell.CreateShortcut(strDesktop & "\" & shortcutName & ".lnk")
    oShellLink.TargetPath = shortcutTarget
    oShellLink.WindowStyle = 1
    oShellLink.Description = shortcutName
    oShellLink.Save
    Set oShellLink = Nothing
    Set WshShell = Nothing
End Sub

'удалим версию 260
installOrUninstall unInstallUID, DistrFolder + "8.2.9.260\setup\1CEnterprise 8.2.msi", "1049.mst", requiredUninstall
'установим версию 356
installOrUninstall InstallUID, DistrFolder + "8.2.9.356\setup\1CEnterprise 8.2.msi", "1049.mst", requiredInstall

```

3.9. appsrvrs.lst

Содержит список серверов «1С:Предприятия», зарегистрированных в утилите администрирования информационных баз в варианте клиент-сервер.

ПРИМЕЧАНИЕ. Применяется только для системы «1С:Предприятие» под ОС Windows.

Расположен в каталоге `%APPDATA%\1C\1Cv8`.

Пример:

```

C:/Documents and Settings/User/Local Settings/Application Data/1C/
1Cv8/appsrvrs.lst
C:/Users/User/AppData/Roaming/1C/1Cv8/appsrvrs.lst

```

[Копировать в буфер обмена](#)

3.10. cfgrepo.conf

Файл `cfgrepo.conf` служит для настройки места положения и размера кеша версий при работе с хранилищем конфигураций.

Расположение файла настройки локального кеша:

- Для ОС Windows: `%APPDATA%\1C\1Cv8\<Уникальный идентификатор информационной базы>\cfgrepo`.
- Для ОС Linux: `~/1c/1Cv8/1Cv8/<Уникальный идентификатор информационной базы>/cfgrepo`.

Расположение файла настройки глобального кеша: в каталоге хранилища конфигурации.

В файле могут быть следующие параметры:

`cfgrepo.cache.path`

Локальный кеш: указывает путь к каталогу, в котором располагается кеш версий.

Глобальный кеш: указывает путь к каталогу, в котором располагается кеш версий, в терминах файловой системы того компьютера, на котором расположен каталог с хранилищем или установлен сервер хранилища. Другими словами, это локальный путь к глобальному кешу версий.

`cfgrepo.cache.network.path`

Глобальный кеш: указывает UNC-путь к каталогу, описывающему расположение глобального кеша версий. Путь, указанный в этом параметре, должен приводить в тот же каталог, что и путь, указанный в параметре `cfgrepo.cache.path`.

`cfgrepo.cache.limit`

Данный параметр описывает максимальный размер кеша версий конфигурации.

3.11. comcntrcfg.xml

Файл `comcntrcfg.xml` служит для указания внешнему соединению необходимости запуска в отладочном режиме.

ПРИМЕЧАНИЕ. Применяется только для системы «1С:Предприятие» под ОС Windows.

Файл располагается в каталоге конфигурационных файлов системы «1С:Предприятие», и его наличие не является обязательным.

Если файл не найден, внешнее соединение открывается в обычном режиме.

Пример:

```

<config xmlns="http://v8.1c.ru/v8/comcntrcfg">
  <debugconfig debug="true" debuggerURL="tcp://localhost:1560"/>
</config>

```

[Копировать в буфер обмена](#)

Элемент `debugconfig` имеет атрибуты, которые описаны ниже.

Атрибут `debug`

Тип: **Булево**. Указывает необходимость запуска в отладочном режиме:

- `debug="true"` – отладка включена;
- `debug="false"` – отладка выключена.

[Копировать в буфер обмена](#)

`debug="true"`

Тип: *Строка*. Указывает адрес отладчика, к которому нужно автоматически подключиться для отладки, где `localhost` указывает на поиск на локальном компьютере, `1560` – номер сетевого порта. Если порт не указан, то будут проверяться все порты в диапазоне `1560-1591`. Указание `tcp://` эквивалентно `tcp://localhost`. Если адрес отладчика не указан, в процессе исполнения кода на встроенном языке отладка производится не будет.

[Копировать в буфер обмена](#)

```
debuggerURL="tcp://localhost:1560"
```

3.12. `conf.cfg`

Файл `conf.cfg` определяет расположение каталога общих конфигурационных файлов и язык интерфейса системы по умолчанию.

Расположение файла:

- Для ОС Windows:
 - В каталоге `bin\conf` каталога конкретной версии «1С:Предприятия»;
 - В каталоге `C:\Program Files\1Cv8\conf`.
- Для ОС Linux:
 - Каталог `conf` установленной версии, например, для 32-разрядной версии «1С:Предприятия» путь к этому каталогу будет выглядеть следующим образом: `/opt/1C/v8.3/i386/conf`, а для 64-разрядной версии: `/opt/1C/v8.3/x86_64/conf`.
 - Каталог `~/.1cv8/1C/1cv8/conf` (`~` – домашний каталог пользователя, от имени которого работает сервер «1С:Предприятия»).

Файл представляет собой текстовый документ в кодировке UTF-8.

В файле могут быть указаны следующие параметры:

`ConfLocation`

Параметр определяет каталог, в котором система будет выполняться поиск конфигурационных файлов (`logcfg.xml`, `nethasp.ini` и т. д.), в том случае, если они не найдены по стандартным путям поиска. Данный параметр имеет смысл, если файл расположен в каталоге `conf` конкретной версии.

По умолчанию значение параметра равно:

- Для ОС Windows: `C:\Program Files\1Cv8\conf`.
- Для ОС Linux:
 - Для 32-разрядной версии: `/opt/1C/v8.3/i386/conf`.
 - Для 64-разрядной версии: `/opt/1C/v8.3/x86_64/conf`.

Пример:

```
ConfLocation=C:\MySettings\v8\conf
```

[Копировать в буфер обмена](#)

`SystemLanguage`

Параметр определяет язык интерфейса системы. В качестве значения параметра могут быть указаны коды языков интерфейса (см. [здесь](#)) или значение `System`. Если указано значение языка, то будет использоваться именно этот язык. Если указано значение `System`, то язык интерфейса будет определяться локализацией операционной системы.

Если указан несуществующий язык локализации, то будет выполнена попытка использовать язык локализации в соответствии с региональными установками операционной системы. Если пользовательский интерфейс на указанном языке не установлен, то будет использован английский интерфейс.

При использовании клиентского приложения, работающего под ОС Windows, следует учитывать следующую особенность: если файл `conf.cfg` с указанием языка интерфейса расположен в каталоге `conf` конкретной версии, то указанный язык интерфейса будет использован для конкретной версии, а если этот файл расположен в каталоге `C:\Program Files\1Cv8\conf`, то указанный язык интерфейса будет использован для всех установленных версий на данном компьютере.

Если в конфигурационном файле параметр `SystemLanguage` не указан, то будет использован механизм определения языка интерфейса с помощью файла `*.res`. Если файл с расширением `*.res` отсутствует, при запуске будет выбираться интерфейс, соответствующий региональным установкам операционной системы. Указание неизвестного или несуществующего кода языка интерфейса эквивалентно отсутствию такого файла.

Пример:

```
SystemLanguage=System
```

[Копировать в буфер обмена](#)

Использовать язык интерфейса в соответствии с региональными установками операционной системы.

```
SystemLanguage=RU
```

[Копировать в буфер обмена](#)

Использовать русский (`RU`) язык интерфейса.

`PublishDistributiveLocation`

Параметр определяет расположение дистрибутива клиентского приложения. Данный параметр по поведению и содержанию аналогичен атрибуту `pubdst` элемента `point` файла `default.vrd` (см. [здесь](#)).

ПРИМЕЧАНИЕ. Применяется только для системы «1С:Предприятие» под ОС Windows.

`AllowSSLv3`

При работе в ОС Windows данный параметр управляет возможностью использования протокола SSL v3.0 в следующих объектах системы: `HTTPСоединение`, `FTPСоединение`, `ИнтернетПочта` (с отключенным режимом совместимости в «1С:Предприятие» версии 8.3.6 и старше), `WSПрокси`, `WSCсылка`, `WSОпределение`. Для использования протокола SSL v3.0 следует установить данный параметр в значение `true`:

```
AllowSSLv3=true
```

[Копировать в буфер обмена](#)

Значение параметра по умолчанию: `false`.

При работе в ОС Linux возможность использования протокола SSL v3 определяется версией используемого пакета `OpenSSL`.

3.13. `debugcfg.xml`

Файл `debugcfg.xml` предназначен для настройки дополнительного диапазона портов, используемого при отладке конфигураций.

Файл располагается в каталоге конфигурационных файлов системы «1С:Предприятие», и его наличие не является обязательным.

Если файл не найден, то для отладки используются порты из стандартного диапазона (1560–1591). Предметы отладки на сервере используют те же порты, что и процессы сервера: **rmng** и **rphost**. Указания дополнительных диапазонов портов для предметов отладки на сервере не требуется.

Пример:

```
<config xmlns="http://v8.1c.ru/v8/debugcfg">
  <debugports range="1540:1550"/>
</config>
```

[Копировать в буфер обмена](#)

Элемент **debugports** имеет атрибуты, которые описаны ниже.

Атрибут **range**

Тип: **Строка**. Содержит дополнительный диапазон портов, используемых при отладке.

3.14. def.usr

Файл содержит имя пользователя, который последним открывал данную информационную базу.

Расположение файла:

- Для ОС Windows: %APPDATA%\1C\1Cv8\<Уникальный идентификатор информационной базы>.
- Для ОС Linux: ~/.1cv8/1C/1cv8/<Уникальный идентификатор информационной базы>.

Пример:

```
C:\Documents and Settings\User\Application Data\1C\1Cv8\4129dbdb-b495-41cb-99ea-ef315060a03e\def.usr
~/.1cv8/1C/1cv8/4129dbdb-b495-41cb-99ea-ef315060a03e/def.usr
```

[Копировать в буфер обмена](#)

3.15. default.vrd

Данный файл служит для настройки веб-клиента и использования Web-сервисов и находится в каталоге виртуального приложения.

ПРИМЕЧАНИЕ. Формат ссылок, используемых в данном файле, должен соответствовать [RFC 1738](http://tools.ietf.org/html/rfc1738.html) (<http://tools.ietf.org/html/rfc1738.html>), [RFC 2396](http://tools.ietf.org/html/rfc2396.html) (<http://tools.ietf.org/html/rfc2396.html>).

3.15.1. Корневой элемент <point>

Корневым элементом конфигурационного файла является элемент **<point>**, который определяет настройки виртуального ресурса. Он может содержать по одному элементу **<zones>**, **<ws>**, **<pool>**, **<debug>** и **<openid>**. При этом в элементе **<ws>** допустимо несколько вложенных элементов **<point>**, а для элемента **<zones>** допустимо несколько вложенных элементов **<zone>**:

```
<point...>
  <ws...>
    <point>...</point>
    <zones>
      <zone>...</zone>
      <zone>...</zone>
    </zones>
  </ws>
  <point>...</point>
</ws>
<httpServices>
  <service>...</service>
</httpServices>
<pool.../>
<debug.../>
<openid>
  <rely... />
  <provider>
    <lifetime>...</lifetime>
  </provider>
</openid>
</point>
```

[Копировать в буфер обмена](#)

Пример:

```
<?xml version="1.0" encoding="UTF-8"?>
<point xmlns="http://v8.1c.ru/8.2/virtual-resource-system"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  base="/demo"
  ib="Srvr=&quot;tcp://Server&quot;;Ref=&quot;demo&quot;;"
  enable="false"
  allowexecutescheduledjobs="force"
  enableStandardOData="true">
  <ws>
    <point name="OperationalData" alias="OperData"/>
    <point name="AnalyticalData" alias="AnalytData"/>
  </ws>
  <httpServices>
    <service name="ПримерРаботы" enable="true"/>
  </httpServices>
  <pool size="50" maxAge="10" attempts="2"/>
  <debug enable="true" url="tcp://localhost"/>
  <zones>
    <zone value="8214" safe="true"/>
    <zone value="last" specify="true" />
  </zones>
</point>
```

[Копировать в буфер обмена](#)

Корневой элемент файла **default.vrd** может содержать следующие атрибуты.

Атрибут **base**

Элемент **base** указывает на относительный путь (относительно корневого каталога сайта) к каталогу виртуального приложения.

СОВЕТ. Рекомендуется задавать имя каталога виртуального приложения с использованием только символов US ASCII.

Пример:

[Копировать в буфер обмена](#)

base="/demoMA"

Атрибут *ib*

Содержит строку подключения к информационной базе «1С:Предприятия». Следует помнить о том, что для файлового варианта и варианта клиент-сервер строки подключения выглядят по-разному.

ПРИМЕЧАНИЕ. Если в строке подключения встречаются символы, недопустимые с точки зрения стандарта XML (<http://www.w3.org/TR/xml11/>), то эти символы должны быть соответствующим образом заменены.

Пример файловой базы:

```
ib="File=c:/bases/demo;"
```

[Копировать в буфер обмена](#)

Пример клиент-серверной базы:

```
ib="Srvr=&quot;;tcp://myServer&quot;;Ref=&quot;;mybase&quot;;;"
```

[Копировать в буфер обмена](#)

В строке подключения можно указать логин и пароль пользователя. В этом случае подключение к информационной базе будет выполняться от имени указанного пользователя. В следующем примере подключение будет осуществляться от имени пользователя **Продавец**:

```
ib="Srvr=&quot;;tcp://myServer&quot;;Ref=&quot;;mybase&quot;;;Usr=Продавец;Pwd=123;"
```

[Копировать в буфер обмена](#)

Однако если в командной строке запуска клиентского приложения указан логин и пароль, то подключение будет выполняться с параметрами, указанными в командной строке.

Атрибут *enable*

Отвечает за возможность работы с опубликованной информационной базой тонким и веб-клиентами. Если атрибут имеет значение **true**, работа с опубликованной информационной базой с помощью тонкого и веб-клиентов возможна, и в этом случае строка соединения будет выглядеть следующим образом (для примера в начале раздела):

```
http://host/demo
```

[Копировать в буфер обмена](#)

В противном случае (атрибут имеет значение **false**) работа с помощью тонкого клиента и веб-клиента невозможна.

Значение по умолчанию: **true** (работа с помощью тонкого клиента и веб-клиента разрешена).

Атрибут *temp*

Позволяет указать каталог временных файлов для работы расширения веб-сервера (**wsisapi.dll**, **wsap22.dll**, **wsapch2.dll**) или файлового варианта информационной базы. Если атрибут не задан, то:

- для файловой информационной базы используется подкаталог **1Cv8Tmp** каталога, в котором расположен файл информационной базы.
- в остальных случаях используется каталог временных файлов пользователя, от имени которого выполняется запрос.

В том случае, если атрибут содержит указание на каталог, который используется в качестве каталога временных файлов для работы расширения веб-сервера, пользователь, от имени которого исполняется расширение веб-сервера, должен иметь полный доступ к этому каталогу и его содержимому.

Атрибут *pubdst*

Указывает полный URL файла с дистрибутивом клиентского приложения, который необходимо скачать и установить в случае несовпадения версий клиентского приложения и сервера. С помощью данного URL дистрибутив должен быть доступен извне компьютера, на котором этот дистрибутив расположен.

Если получение дистрибутива клиентского приложения будет выполняться с помощью **https**-соединения, то компьютер, получающий дистрибутив, будет выполнять проверку сертификата сервера, с которого выполняется получение дистрибутива, с помощью сертификатов корневых удостоверяющих центров, которые получаются из файла **cacert.pem** (расположенного в каталоге исполняемых файлов платформы) и из хранилища сертификатов ОС Windows.

ПРИМЕЧАНИЕ. Применяется только для системы «1С:Предприятие» под ОС Windows.

Вид архива: **zip**. Структура файлов в архиве: без иерархии и каталогов, только файлы дистрибутива клиентского приложения.

Пример:

```
pubdst="http://www.myhost.ru/files/client.zip"
```

[Копировать в буфер обмена](#)

В случае смены версии сервера, достаточно заменить файл с архивом клиентского приложения.

Атрибут *allowexecutescheduledjobs*

Атрибут управляет возможностью исполнения регламентных заданий расширением веб-сервера для файлового варианта информационной базы.

Атрибут может принимать следующие значения:

- **off** – в этом случае расширение веб-сервера не будет выполнять регламентные задания. В этом случае выполнять регламентные задания будет клиентское приложение (если таковое существует), которое подключается к информационной базе напрямую, без использования веб-сервера.
- **force** – в этом случае расширение веб-сервера будет выполнять регламентные задания.

Значение по умолчанию: не задано. В этом случае регламентные задания будет выполнять то приложение, с помощью которого будет осуществлено первое подключение к информационной базе.

Атрибут *enableStandardOData*

Атрибут управляет доступностью стандартного интерфейса OData через указанную публикацию. Атрибут может принимать следующие значения:

- **true** – через текущую публикацию возможна работа с данными с помощью REST-сервисов;
- **false** – через текущую публикацию работа с данными с помощью REST-сервисов не доступна.

3.15.2. Элемент <ws>

Элемент содержит настройки публикации Web-сервисов, подчинен элементу <point>. Элемент <ws> может быть один или ни одного. Данный элемент может содержать произвольное количество элементов <point>.

Данный элемент может содержать следующие атрибуты.

Атрибут *enable*

Отвечает за возможность работы Web-сервисов в данной информационной базе. Если атрибут имеет значение `true` (или атрибут отсутствует), работа Web-сервисов возможна. В противном случае (атрибут имеет значение `false`) работа Web-сервисов невозможна.

Значение по умолчанию: `true` (работа Web-сервисов разрешена).

Атрибут `pointEnableCommon`

Отвечает за возможность использования Web-сервисов в данной информационной базе, которые опубликованы без явного указания разрешения использования (атрибут `enable` элемента `point`). Если атрибута имеет значение `true`, то все Web-сервисы, для которых явно не указано значение атрибута `enable` элемента `point` будут разрешены для использования. В противном случае использование таких Web-сервисов будет запрещено.

Значение по умолчанию: `true` (работа Web-сервисов разрешена).

3.15.2.1. Элемент `<point>`

Элемент содержит описание публикуемого Web-сервиса. Элемент подчинен элементу `<ws>`. Элементов `<point>` может быть не менее одного.

Если Web-сервис явно не указан в файле `default.vrd`, и разрешено использование Web-сервисов прикладного решения, то обращение к такому Web-сервису возможно только по имени Web-сервиса (свойство Web-сервиса *Имя*). Обращение по синониму (`alias`) будет недоступно даже в том случае, если этот синоним указан в свойстве Web-сервиса *Имя файла публикации*. Для того чтобы обращение к Web-сервису было доступно и по имени и по синониму (`alias`) – нужно явно указать требуемый Web-сервис в файле `default.vrd` (включая синоним).

Данный элемент может содержать следующие атрибуты.

Атрибут `name`

Имя публикуемого Web-сервиса. Обращение к сервису возможно как по ссылке, включающей имя Web-сервиса, так и по ссылке, включающей синоним Web-сервиса.

Для Web-сервиса, описанного строкой:

```
...
base="/demo"
...
<point name="OperationalData" alias="OperData"/>
```

[Копировать в буфер обмена](#)

Обращение возможно следующими способами:

```
http://host/demo/ws/OperationalData
http://host/demo/ws/OperData
```

[Копировать в буфер обмена](#)

СОВЕТ. Рекомендуется задавать имя Web-сервиса с использованием только символов US ASCII.

Атрибут `alias`

Синоним публикуемого Web-сервиса. Обращение к сервису возможно как по ссылке, включающей имя Web-сервиса, так и по ссылке, включающей синоним Web-сервиса (если синоним указан в файле `default.vrd`).

Для Web-сервиса, который опубликован следующим образом:

```
...
base="/demo"
...
<point name="OperationalData" alias="OperData"/>
```

[Копировать в буфер обмена](#)

Обращение возможно следующими способами:

```
http://host/demo/ws/OperationalData
http://host/demo/ws/OperData
```

[Копировать в буфер обмена](#)

СОВЕТ. Рекомендуется задавать синоним Web-сервиса с использованием только символов US ASCII.

Атрибут `enable`

Признак разрешения или запрета использования конкретного Web-сервиса.

Значение по умолчанию: `true` (публикация разрешена).

3.15.3. Элемент `<httpServices>`

Элемент содержит настройки публикации HTTP-сервисов, подчинен элементу `<point>`. Элемент `<httpServices>` может быть один или ни одного. Данный элемент может содержать произвольное количество элементов `<service>`.

Данный элемент может содержать следующие атрибуты.

Атрибут `publishByDefault`

Если данный атрибут не задан или установлен в значение `true`, то все добавляемые в конфигурацию HTTP-сервисы будут автоматически доступны для использования, если это явно не запрещено с помощью элемента `<service>`.

Значение по умолчанию: `true` (работа HTTP-сервисов разрешена).

3.15.3.1. Элемент `<service>`

Элемент содержит описание публикуемого HTTP-сервиса. Элемент подчинен элементу `<httpServices>`. Элементов `<service>` может быть не менее одного.

Данный элемент может содержать следующие атрибуты.

Атрибут `name`

Содержит имя HTTP-сервиса, как оно задано в конфигураторе. Для обращения к сервису данное имя не используется.

Атрибут `rootUrl`

Содержит значение свойства *Корневой URL* свойства HTTP-сервиса. Свойство используется для определения HTTP-сервиса, который должен обработать поступивший запрос.

Атрибут `enable`

Признак разрешения или запрета использования конкретного HTTP-сервиса.

Значение по умолчанию: `false` (использование запрещено).

3.15.4. Элемент `<pool>`

Элемент содержит настройки пула соединений с информационной базой. Элемент `<pool>` может быть один или ни одного.

Данный элемент может содержать следующие атрибуты:

Атрибут `size`

Размер пула – максимальное количество соединений в пуле.

Значение по умолчанию равно 100.

Атрибут `maxAge`

Время жизни соединения в пуле – максимальное время жизни соединения в пуле, в секундах. Если в течение указанного времени соединение не было востребовано, оно будет удалено из пула.

Значение по умолчанию равно 20 с.

Атрибут `attempts`

Максимальное количество попыток установки соединения с сервером «1С:Предприятия».

Значение по умолчанию равно 5.

Атрибут `attemptTimeout`

Время ожидания установления соединения с сервером «1С:Предприятия», в секундах.

Значение по умолчанию равно 0,5 с.

Атрибут `waitTimeout`

Время ожидания между попытками установления соединения с сервером «1С:Предприятия», в секундах.

Значение по умолчанию равно 0,5 с.

Пример:

```
<pool size="50" maxAge="10" attempts="2" attemptTimeout="1" waitTimeout="1"/>
```

[Копировать в буфер обмена](#)

3.15.5. Элемент `<debug>`

Атрибут `enable`

Указывает необходимость запуска в отладочном режиме:

- `enable="true"` – отладка включена;
- `enable="false"` – отладка выключена.

Атрибут `url`

Указывает адрес отладчика, к которому нужно автоматически подключиться для отладки, где `localhost` указывает на поиск на локальном компьютере, «1560» – номер сетевого порта. Если порт не указан, то будут проверяться все порты в диапазоне портов 1560–1591. Указание `tcp://` эквивалентно `tcp://localhost`. Если адрес отладчика не указан, в процессе исполнения кода на встроенном языке отладка производиться не будет.

Пример:

```
<debug enable="true" url="tcp://localhost"/>
```

[Копировать в буфер обмена](#)

3.15.6. Элемент `<zones>`

Элемент `<zones>` подчинен элементу `<point>` и может быть один или ни одного. Элементу `<zones>` подчинен один или несколько элементов `<zone>`.

Данный элемент не содержит атрибутов.

3.15.6.1. Элемент `<zone>`

Каждый элемент `<zone>` описывает один разделитель. Порядок следования элементов `<zone>` в элементе `<zones>` соответствует порядку следования разделителей в конфигураторе. При изменении порядка следования разделителей следует изменить файл `default.vrd`. Количество элементов `<zone>` не должно превышать количество разделителей. Если количество элементов будет больше, чем количество разделителей, то при подключении к информационной базе, опубликованной таким образом, будет вызвано исключение. Если количество элементов меньше количества разделителей, то для неуказанных разделителей значением будет являться значение по умолчанию для типа разделителя, а использование разделителя будет выключено.

Элемент `<zone>` может содержать следующие атрибуты.

Атрибут `safe`

Определяет возможность изменения значений объектов, связанных с механизмом разделения данных в том случае, если доступ к информационной базе осуществляется с помощью веб-клиента или тонкого клиента, подключенного через веб-сервер (безопасный режим разделения данных). Этот атрибут нужно использовать в том случае, когда нужна гарантия, что при доступе к информационной базе через Интернет будет невозможно получить доступ к другой области данных.

Значение по умолчанию: `false` (изменения разрешены).

Если значение атрибута равно `true`, то в сеансе, использующем данную публикацию информационной базы, запрещено:

- Выключать использование разделителя, если разделение не является условно выключенным.
- Изменять значение используемого разделителя, если разделение не является условно выключенным.
- Изменять объекты, которые управляют условным разделением:
 - Указанные для самого разделителя;
 - Указанные для объектов, входящих в состав разделителя.

Атрибут `specify`

Определяет необходимость присутствия значения данного разделителя в адресе публикуемой информационной базы.

Значение по умолчанию: `false` (разделитель не участвует в формировании адреса).

Атрибут `value`

Предназначен для явного указания значения разделителя, который находится на этой позиции.

Неуказанное значение атрибута `value` совместно с атрибутом `specify`, установленным в значение `false`, интерпретируется как отсутствие значения разделителя (соответствует значению «-» в значении параметра `Zn` строки соединения).

Если для атрибута `specify` указано значение `true` и для атрибута `value` указано значение, то это значение (без учета регистра символов) должно быть явно указано в строке адреса информационной базы в соответствующей позиции. В противном случае при попытке доступа к информационной базе будет получена ошибка с кодом `404` (не удалось найти веб-страницу).

Символы, недопустимые в URL ([RFC 1738](http://tools.ietf.org/html/rfc1738), <http://tools.ietf.org/html/rfc1738.html>), преобразуются в кодировку UTF-8 и кодируются в соответствии с разделом 2.2. URL Character Encoding Issues стандарта RFC 1738 (с помощью символа «%» и двух шестнадцатеричных символов).

Пример файла `default.vrd`:

```
<?xml version="1.0" encoding="UTF-8"?>
<point xmlns="http://v8.lc.ru/8.2/virtual-resource-system"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  base="/test"
  ib="File=&quot;c:\base&quot;;">
  <ws enable="false"/>
  <zones>
    <zone value="8214" safe="true"/>
    <zone specify="true" />
    <zone />
    <zone specify="true" />
    <zone value="last" specify="true" />
  </zones>
</point>
```

[Копировать в буфер обмена](#)

В данном примере в прикладном решении определено 5 разделителей. Адрес информационной базы будет выглядеть следующим образом:

```
http://hostname/test/01/20101231235959/last
```

[Копировать в буфер обмена](#)

Что будет интерпретировано следующим образом:

- <http://hostname/test> – адрес собственно информационной базы.
- Первый разделитель не должен быть указан в адресе (значение атрибута `specify` по умолчанию равно `false`), его значение равно `8214`, и нельзя программно управлять данным разделителем (значение атрибута `safe` равно `true`). Остальными разделителями допустимо управлять программно, т. к. значение атрибута `safe` для элементов `zone` не задано, а значение по умолчанию (`false`) разрешает программное управление.
- Второй разделитель должен быть указан в адресе (значение атрибута `specify` равно `true`), и его значение равно `01`.
- Третий разделитель отключен.
- Четвертый разделитель должен быть указан в адресе (значение атрибута `specify` равно `true`), и его значение равно `31-12-2010 23:59:59`.
- Последний разделитель должен быть указан в адресе, и его значение должно быть только `last`.

Такой вариант указания разделителей может использоваться для тонкого клиента, работающего через веб-сервер, в веб-клиенте и для Web-сервисов.

В случае одновременного указания значений разделителей разными способами значения разделителей, которые будут использоваться в сеансе, определяются следующим образом:

- Если в файле `default.vrd` задан элемент `<zones>`, то значения разделителей, указанных в адресе информационной базы, имеют наивысший приоритет. При этом:
 - Значения, указанные в параметре запуска (параметр `Z`), игнорируются.
 - Значения, указанные в строке соединения с информационной базой, игнорируются (параметр `Zn` в атрибуте `ib` элемента `<point>`).
- Если в файле `default.vrd` не задан элемент `<zones>`, то:
 - Выполняется попытка определить значения разделителей из параметра `Z` строки адреса.
 - Если параметр не указан, то выполняется попытка использовать значения, указанные в строке соединения с информационной базой (параметр `Zn` в атрибуте `ib` элемента `<point>`).
- В общем случае приоритет мест указания значений разделителей следующий (приоритет уменьшается сверху вниз):
 - Адрес информационной базы (если в файле `default.vrd` указан элемент `<zones>`).
 - Командная строка запуска (параметр `Z`).
 - Строка соединения информационной базы (параметр `Zn` в атрибуте `ib` элемента `<point>`).

3.15.7. Элемент `<openid>`

Данный элемент описывает настройки, связанные с OpenID-аутентификацией. Элемент `<openid>` подчинен элементу `<point>` и может быть один или ни одного. Элементу `<openid>` подчинены элементы `<rely>` и `<provider>`. Подчиненные элементы могут быть или в единственном числе или отсутствовать.

Данный элемент не содержит атрибутов.

3.15.7.1. Элемент `<rely>`

Элемент содержит адрес информационной базы, выступающей в роли OpenID-провайдера.

Атрибут `url`

Указывает URL информационной базы «1С:Предприятия», выступающей в роли OpenID-провайдера. Информационная база должна быть опубликована особым образом.

ВНИМАНИЕ! Взаимодействие с OpenID-провайдером осуществляется только по HTTPS-соединению.

ПРИМЕЧАНИЕ. URL OpenID-провайдера не должен завершаться символом «/». Правильно: <https://myserver.org/users-ib/e1cib/oid2op>, неправильно: <https://myserver.org/users-ib/e1cib/oid2op/>.

Пример:

```
<rely url="https://myserver.org/users-ib/elcib/oid2op"/>
```

[Копировать в буфер обмена](#)

3.15.7.2. Элемент <provider>

Элемент указывает, что данная информационная база выступает в роли OpenID-провайдера. Данному элементу подчинен элемент <lifetime>, который может быть один или ни одного.

Пример:

```
<?xml version="1.0" encoding="UTF-8"?>
<point xmlns="http://v8.1c.ru/8.2/virtual-resource-system"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  base="/demo"
  ib="Srvr=&quot;tcp://Server&quot;;Ref=&quot;demo&quot;;" enable="false">
  <openid>
    <provider/>
  </openid>
</point>
```

[Копировать в буфер обмена](#)

3.15.7.2.1. Элемент <lifetime>

Элемент указывает время жизни признака аутентифицированности идентификатора в секундах. Если не указан, то значением по умолчанию является 86 400 секунд (24 часа). Максимальное время жизни признака аутентифицированности идентификатора составляет 604 800 секунд (7 суток). При указании значения элемента *lifetime* более максимального значения, будет использовано максимальное значение.

Пример:

```
<?xml version="1.0" encoding="UTF-8"?>
<point xmlns="http://v8.1c.ru/8.2/virtual-resource-system"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  base="/demo"
  ib="Srvr=&quot;tcp://Server&quot;;Ref=&quot;demo&quot;;" enable="false">
  <openid>
    <provider>
      <lifetime>432000</lifetime>
    </provider>
  </openid>
</point>
```

[Копировать в буфер обмена](#)

3.16. inetcfg.xml

Файл *inetcfg.xml* позволяет задавать настройки прокси по умолчанию и имеет больший приоритет над настройками прокси по умолчанию в ОС Windows или параметрами, указанными в переменных окружения в ОС Linux.

Файл располагается в каталоге конфигурационных файлов системы «1С:Предприятие», и его наличие не является обязательным.

Для ОС Windows:

- При наличии файла – настройки берутся из файла;
- При отсутствии файла – настройки берутся из настроек Microsoft Internet Explorer.

Для ОС Linux:

- При наличии файла – настройки берутся из файла;
- При отсутствии файла – происходит попытка получения настроек из переменных окружения *http_proxy*, *https_proxy*, *ftp_proxy*, *ftps_proxy* (в соответствии с используемым протоколом). Если эти переменные среды не указаны, производится попытка получить настройки прокси из переменной окружения *all_proxy*. Если в переменных окружения указаны некорректные параметры – они игнорируются.

При настройке прокси можно использовать информацию *User-Agent* из HTTP-запроса:

- тонкий клиент – **1CV8C**;
- Web-сервис – **1C+Enterprise/8.3**;
- веб-клиент – данный параметр формирует веб-браузер.

Пример:

```
<InternetProxy
  protocols="http=10.1.0.8:8080 10.1.0.9:8080"
  user="proxyUser"
  password="proxyPassword"
  bypassOnLocal="true"
  bypassOnAddresses="127.0.0.1 *.master"
/>
```

[Копировать в буфер обмена](#)

Корневой элемент *InternetProxy*, задающий настройки прокси по умолчанию, имеет рассмотренную ниже структуру (атрибуты).

Атрибут ntlm необязательный

Тип: **Булево**. Признак использования механизма NTLM-аутентификации:

- **true** – NTLM-аутентификация включена;
- **false** – выключена.

По умолчанию NTLM-аутентификация включена.

Атрибут protocols необязательный

Тип: **Строка**. Задаёт имя и порт хоста для протоколов. Имеет следующий формат:

```
ПараметрыПроксиПротокола1ПараметрыПроксиПротокола2
...ПараметрыПроксиПротоколаN
ПараметрыПроксиПротокола:=[Протокол]"="хост":порт
```

[Копировать в буфер обмена](#)

Список параметров прокси-протоколов разделен пробелами. Каждый параметр состоит из необязательного имени протокола, знака "=", имени хоста и порта прокси-сервера, разделенных двоеточием. Если имя протокола не указано, то параметры прокси используются для всех протоколов, для которых они явно не указаны. Протоколы могут иметь следующие имена:

- `http`,
- `https`,
- `ftp`.

Регистр является значимым, другие имена протоколов не поддерживаются, например:

```
protocols="http=10.1.0.8:8080 10.1.0.9:8080"
```

[Копировать в буфер обмена](#)

Где:

- для протокола `http` определены параметры прокси: хост – `10.1.0.8`, порт – `8080`;
- для остальных протоколов (`https`, `ftp`): хост – `10.1.0.9`, порт – `8080`.

Атрибут `user` необязательный

Тип: **Строка**. Имя пользователя для аутентификации на прокси-сервере, например:

```
user="proxyUser"
```

[Копировать в буфер обмена](#)

Атрибут `password` необязательный

Тип: **Строка**. Пароль пользователя для аутентификации на прокси-сервере. Пример:

```
password="proxyPassword"
```

[Копировать в буфер обмена](#)

Атрибут `bypassOnLocal` необязательный

Тип: **Булево**. Признак того, использовать ли прокси-сервер для локальных адресов:

- `true` – не использовать;
- `false` – использовать.

Локальность адреса определяется по наличию точки в DNS имени адреса (т. е. все IP-адреса не являются локальными). Поэтому может получиться так, что фактически локальный адрес не распознается как локальный.

Например, `<пользователь>.<домен>` является в Windows XP локальным адресом, но не распознается как локальный. Для того чтобы запретить использовать прокси для адресов, которые воспринимаются как локальные, используется следующий параметр:

```
bypassOnLocal="true"
```

[Копировать в буфер обмена](#)

Для всех остальных адресов нужно использовать параметр `bypassOnAddresses`.

Атрибут `bypassOnAddresses` необязательный

Тип: **Строка**. Список адресов, для которых прокси не используется. Имеет следующий формат:

```
хост1 хост2 ... хостN
```

[Копировать в буфер обмена](#)

Имена хостов разделяются пробелами. Имя хоста может содержать специальные символы маски: `*` – любое количество символов, `?` – любой символ. Например, чтобы заблокировать прокси для всех хостов домена, нужно использовать: `*.<ИМЯ ДОМЕНА>`, например:

```
bypassOnAddresses="127.0.0.1 *.master"
```

[Копировать в буфер обмена](#)

В примере для адреса `127.0.0.1` (`localhost`) и для всех адресов домена `master` прокси не используется.

3.17. location.cfg

Файл `location.cfg` служит для указания каталога, который хранит файлы настроек пользователя и расположения файла программной лицензии. Для указания расположения каталога служит параметр `location`.

location

Путь к каталогу.

Например:

```
location=C:\Users\UserName\AppData\Roaming\1c\1cv82
```

[Копировать в буфер обмена](#)

3.18. logcfg.xml

Файл `logcfg.xml` служит для настройки параметров технологического журнала и механизма генерации дампов памяти при аварийном завершении работы системы «1С:Предприятие».

Файл располагается в каталоге конфигурационных файлов системы «1С:Предприятие», и его наличие не является обязательным.

Если файл не найден, то технологический журнал имеет следующие настройки по умолчанию:

- Технологический журнал **выключен**.
- Дампы сохраняются в каталог `%USERPROFILE%\Local Settings\Application Data\1C\1Cv8\dumps` (`%LOCALAPPDATA%\1C\1Cv8\dumps` для ОС Windows Vista и выше).
- Сохраняются **минимальные** дампы при аварийном завершении работы системы.
- Технологический журнал по умолчанию **включен**.
- Технологический журнал по умолчанию сохраняется в каталог `%USERPROFILE%\Local Settings\1C\1Cv8\logs` (`%LOCALAPPDATA%\1C\1Cv8\logs` для ОС Windows Vista и выше). Информация удаляется из технологического журнала по умолчанию через 24 часа.
- Уровень формирования событий технологического журнала по умолчанию установлен в `Error`.

Пример:

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
  <log location="c:\v8\logs" history="1">
    <event>
      <eq property="name" value="conn" />
    </event>
  </log>
  <dump location="c:\v8\dumps" create="1" type="2" />
</config>
```

Копировать в буфер обмена

Данный конфигурационный файл указывает на следующее:

- в технологическом журнале регистрируются все события установки и разрыва клиентского соединения с сервером;
- файлы технологического журнала располагаются в каталоге `c:\v8\logs`;
- файлы технологического журнала хранятся в течение одного часа;
- файлы дампа помещаются в каталог `c:\v8\dumps`;
- файлы дампа содержат всю доступную информацию (содержимое всей памяти процесса).

СОВЕТ. Для редактирования файла конфигурации технологического журнала удобно использовать специальный инструмент, расположенный на диске ИТС: [Технологическая поддержка – 1С:Предприятие 8 – Универсальные отчеты и обработки – Настройка технологического журнала \(http://its.1c.ru/db/metod8dev/content/3474/hdoc\)](http://its.1c.ru/db/metod8dev/content/3474/hdoc).

3.18.1. Структура конфигурационного файла

Корневым элементом конфигурационного файла является элемент `<config>`, который определяет настройки технологического журнала. Он может содержать несколько элементов `<log>`, один элемент `<dump>`, один элемент `<leaks>`, один элемент `<plansql>`, один элемент `<defaultlog>`, один или несколько элементов `<system>`:

```
<config...>
  <log...>...</log>
  <log...>...</log>
  <log...>...</log>
  <dump ... />
  <leaks>...</leaks>
  <mem />
  <plansql />
  <system ... />
  <system ... />
  <system ... />
  <defaultlog ... />
</config>
```

Копировать в буфер обмена

Эти элементы отвечают за следующее:

- Элемент `<log>` определяет каталог технологического журнала и его состав (см. [здесь](#)).
- Элемент `<dump>` определяет каталог для записи дампов аварийного завершения (см. [здесь](#)).
- Элемент `<leaks>` устанавливает отслеживание утечек памяти (см. [здесь](#)), которые могут быть вызваны ошибками в коде конфигурации. Отслеживание утечек памяти несколько снижает производительность.
- Элемент `<mem>` предназначен для учета используемой памяти (см. [здесь](#)).
- Элемент `<plansql>` предназначен для управления сбором планов запроса, формируемых при работе различных СУБД (см. [здесь](#)). Собственно планы запросов содержатся в свойстве `<planSQLText>` событий, связанных с СУБД.
- Элемент `<defaultlog>` определяет каталог и время жизни технологического журнала по умолчанию (см. [здесь](#)).
- Элемент `<system>` определяет настройки формирования системных событий (см. [здесь](#)).

Элементы можно разделить на несколько групп:

1. Элементы, которые управляют формированием того или иного события. К таким элементам относятся `<dump>`, `<leaks>`, `<mem>`, `<plansql>`, `<ftextupd>`, `<system>`. При этом, если в файле настройки технологического журнала не указан нужный элемент, то соответствующее событие не формируется системой. Другими словами, если, например, учет используемой памяти не включен элементом `<mem>`, то наличие отбора по событию `МЕМ` никак не повлияет на наполнение технологического журнала, т. к. событие просто не формируется.
2. Элементы, которые задают отбор для уже сформированных данных технологического журнала. К таким элементам относятся `<event>` и `<property>`. С помощью этих элементов можно «обрабатывать» те события, которые сформированы системой. С помощью этих элементов можно только уменьшить количество данных, записываемых в файлы технологического журнала.
3. Элементы, которые управляют расположением файлов с данными (технологический журнал и дампы). К таким элементам можно отнести `<log>`, `<defaultlog>`, `<dump>`.

3.18.1.1. Элемент `<log>`

Элемент `<log>` определяет каталог технологического журнала и условия отбора, по которым в технологический журнал помещаются ранее сформированные события.

Атрибуты элемента:

Атрибут location

Имя каталога, в котором будет размещаться технологический журнал.

ВНИМАНИЕ! Необходимо иметь в виду, что каталог технологического журнала не предназначен для хранения в нем файлов, которые не относятся к технологическому журналу. Поэтому не следует размещать в нем дампы или использовать каталог, который может содержать файлы, не относящиеся к технологическому журналу «1С:Предприятия». Если в каталоге, который указан в качестве каталога технологического журнала, имеются посторонние файлы, то указание каталога считается неверным, и технологический журнал не создается.

ПРИМЕЧАНИЕ. В элементах `<log>`, `<dump>` и `<defaultlog>` следует указывать разные каталоги в атрибутах `location`.

Атрибут history

Количество часов, через которое информация будет удаляться из технологического журнала.

В элемент `<log>` могут быть вложены элементы `<event>` и `<property>`, состав которых определяет условие записи в журнал каждого события и условия записи каждого свойства события.

Если этот элемент не содержит ни одного элемента `<event>`, то никакие события в журнал записываться не будут.

3.18.1.2. Элемент `<event>`

Последовательность элементов `<event>` определяет условие, при выполнении которого событие будет помещено в журнал. В журнал помещаются только такие события, которые удовлетворяют условию. Иначе говоря, если условие, определяемое последовательностью элементов `<event>`, принимает значение *Истина*, то событие будет записано в журнал. Событие включается в журнал, если оно удовлетворяет всем условиям внутри хотя бы одного из элементов `<event>`. То есть условия внутри `<event>` объединяются «по И», а элементы `<event>` объединяются «по ИЛИ».

Условия задаются элементами:

- `eq` – равно;
- `ne` – не равно;
- `gt` – больше;
- `ge` – больше или равно;
- `lt` – меньше;
- `le` – меньше или равно;
- `like` – соответствие маске.

Каждый из этих элементов, кроме элемента `like`, определяет простое сравнение значения параметра события (имя которого задается атрибутом `property`) со значением атрибута `value`.

Пример:

```
<event>
  <eq property="name" value="proc" />
</event>
```

[Копировать в буфер обмена](#)

В данном случае в технологическом журнале будут регистрироваться события, относящиеся к группе с именем `PROC`.

Доступны следующие имена групп событий.

`ADMIN`

Управляющие воздействия администратора кластера серверов системы «1С:Предприятие».

`ATTN`

Записи подсистемы мониторинга состояния кластера «1С:Предприятия».

`CALL`

Входящий удаленный вызов (удаленный вызов на стороне приемника вызова).

`CONN`

Установка или разрыв клиентского соединения с сервером.

`CLSTR`

Выполнение операций, изменяющих работу кластера серверов.

`DB2`

Исполнение операторов SQL СУБД DB2.

`DBMSSQL`

Исполнение операторов SQL СУБД Microsoft SQL Server.

`DBPOSTGRS`

Исполнение операторов SQL СУБД PostgreSQL.

`DBORACLE`

Исполнение операторов SQL СУБД Oracle Database.

`DEV8DBENG`

Исполнение операторов SQL файловой СУБД.

`EDS`

Работа с внешними источниками данных.

`EXCP`

Исключительные ситуации приложений системы «1С:Предприятие», которые штатно не обрабатываются и могут послужить причиной аварийного завершения серверного процесса или подсоединенного к нему клиентского процесса.

`EXCPCNTX`

События, которые начались, но не закончились в момент возникновения нештатной ситуации.

`FTEXTCheck`

Возникает при проверке файлов индекса полнотекстового поиска.

`FTEXTUpd`

Возникает во время обновления файлов индекса полнотекстового поиска.

`HASP`

Событие описывает одно обращение к аппаратному ключу защиты.

LEAKS

События, связанные с утечкой памяти, которые могут быть вызваны ошибками в коде конфигурации.

MAILPARSEERR

Событие, формируемое в том случае, если во время разбора почтового сообщения возникла ошибка.

MEM

События, связанные с увеличением объема памяти, занятой серверными процессами (**ragent**, **rmngr**, **rphost**).

PROC

События, относящиеся к процессу целиком и влияющие на дальнейшую работоспособность процесса. Например: старт, завершение, аварийное завершение и т. п.

QERR

События, связанные с обнаружением ошибок компиляции запроса или ограничения на уровне записей и полей базы данных.

SCALL

Исходящий удаленный вызов (исходящий вызов на стороне источника вызова).

SCOM

События создания или удаления серверного контекста, обычно связанного с информационной базой.

SDBL

События, связанные с исполнением запросов к модели базы данных системы «1С:Предприятие».

SESN

Действия, относящиеся к сеансу работы. Например: начало сеанса, окончание сеанса и т. д.

SYSTEM

Системные события механизмов платформы, предназначенные для анализа специалистами фирмы «1С».

TDEADLOCK

Обнаружена взаимоблокировка в управляемом режиме.

TIMEOUT

Превышение максимального времени ожидания транзакционной блокировки.

TLOCK

Управление транзакционными блокировками в управляемом режиме.

VRSCACHE

Работа кеша серверных вызовов.

VRSREQUEST

Запрос к серверу за некоторым ресурсом.

VRSRESPONSE

Ответ сервера.

Также следует заметить, что события из групп **PROC**, **SCOM**, **EXCP**, **CONN** и **ADMIN** возникают относительно редко и содержат небольшое количество информации, в то время как регистрация событий из групп **SDBL**, **DB2**, **DEMSSQL**, **DBPOSTGRS**, **DBORACLE** может приводить к значительному росту объема технологического журнала.

Элемент **like** определяет, соответствует ли свойство события технологического журнала некоторой маске. Маска представляет собой последовательность символов, некоторые из которых означают сами себя, а некоторые являются шаблонами и служат для описания группы символов.

Например, элемент `<like property="SDBL" value="%reference%"/>` означает проверку значения свойства **SDBL** события технологического журнала на соответствие маске **%reference%**.

К шаблонам относятся:

- **%** – 0 или более произвольных символов.
- **_** – 1 произвольный символ.
- **[...]** – один из перечисленных символов, причем **[...]** может содержать произвольные символы, а также диапазоны вида **c-c**, где **c** – начальный символ диапазона, **c** – конечный символ диапазона.
- **[^...]** – один любой символ, кроме перечисленных в скобках **[]**.
- **** – префиксный символ. Игнорируется и означает, что следующий за ним символ – это просто символ, означающий сам себя (а не шаблон).
- **Все другие символы** – это простые символы, которые означают сами себя. При сравнении простых символов регистры букв не различаются.

Примеры шаблонов:

- **шаблон** – строка с конкретным текстом. В этом случае сравнение **like** не отличается от сравнения **eq**. Регистры букв не различаются.
- **%reference%** – строка, содержащая контекст **reference** в произвольном месте. Регистры букв не различаются.
- **reference%** – строка, содержащая контекст **reference** в начале. Регистры букв не различаются.
- **%reference** – строка, содержащая контекст **reference** в конце. Регистры букв не различаются.
- **%[a-z]** – строка с маленькой английской буквой от **a** до **z** в конце.
- **%[^a-z]** – строка, содержащая хотя бы один символ, отличающийся от маленькой английской буквы.

ЗАМЕЧАНИЕ. Фильтрация событий по шаблонам медленнее, чем при использовании других элементов сравнения. Использование сложной фильтрации событий и свойств технологического журнала может несколько замедлить работу системы «1С:Предприятие».

Пример:

```
<log location="c:\logs" history="1">
  <event>
    <eq property="name" value="proc"/>
  </event>
  <event>
    <eq property="name" value="scom"/>
  </event>
  <event>
    <eq property="name" value="conn"/>
  </event>
  <event>
    <eq property="name" value="excp"/>
  </event>
  <event>
    <eq property="name" value="dbmssql"/>
  </event>
</log>
```

[Копировать в буфер обмена](#)

В данном примере указывается, что в технологическом журнале будут регистрироваться события, относящиеся к группам **PROC**, **SCOM**, **CONN**, **EXCP** и **DBMSSQL**.

3.18.1.3. Элемент <property>

Элемент <property> определяет условия попадания в журнал значения свойства события, имя которого является значением атрибута **name** при условии, что само событие попадает в технологический журнал. Условия задаются вложенными элементами <event> по таким же правилам, что и для событий.

Если элемент <property> с определенным именем отсутствует, то соответствующее свойство не пишется. Если элемент <property> не содержит вложенных элементов <event>, то определяемое им свойство пишется для всех событий, попадающих в журнал, в которых оно присутствует. Если элемент <property> содержит вложенные элементы <event>, то свойство будет записано только для событий, удовлетворяющих условию (если само событие в журнал записывается и событие имеет данное свойство).

Элемент <property name="all"> </property> включает записи в журнал всех свойств событий.

Приведенный ниже элемент <log> определяет запись в журнал событий: процесса, серверного контекста, соединения, исключений и исполнение операторов SQL. Причем текст оператора SQL будет помещен в журнал, только если он выполнялся более секунды. Журнал располагается в каталоге **c:\logs** и хранится 1 час.

Пример:

```
<log location="c:\logs" history="1">
  <event>
    <eq property="name" value="proc"/>
  </event>
  <event>
    <eq property="name" value="scom"/>
  </event>
  <event>
    <eq property="name" value="conn"/>
  </event>
  <event>
    <eq property="name" value="excp"/>
  </event>
  <event>
    <eq property="name" value="dbmssql"/>
  </event>
  <property name="sql">
    <event>
      <eq property="name" value="mssql"/>
      <gt property="duration" value="10000"/>
    </event>
  </property>
</log>
```

[Копировать в буфер обмена](#)

Каждое событие имеет набор свойств. Каждое свойство имеет имя. Возможно присутствие в событии нескольких свойств с одинаковыми именами. Имена свойств могут использоваться для фильтрации событий и свойств. Большие и малые буквы при сравнении имен не различаются. Пустое условие в элементе <property> будет означать, что свойство будет выводиться при любом условии.

ПРИМЕЧАНИЕ. Свойство события выводится, только если для него присутствует элемент <property>.

Далее перечислены основные свойства событий, которые могут потребоваться для настройки конфигурационного файла или просмотра технологического журнала:

- **Admin** – имя администратора кластера или центрального сервера.
- **All** – включает запись всех событий журнала.
- **ApplicationExt** – уточнение требования назначения функциональности (для события **CLSTR**).
- **Attempts** – количество попыток установки соединения с процессом, завершившихся ошибкой (для события **ATTN**).
- **AvMem** – Значение показателя **Доступная память** в момент вывода в технологический журнал (для события **FTEXTUpd**).
- **AvgExceptions** – среднее количество исключений за последние 5 минут по другим процессам (для события **ATTN**).
- **Body** – размер в байтах тела запроса/ответа (для событий **VRSREQUEST**, **VRSRESPONSE**).
- **BodyText** – полный текст запроса/ответа (для событий **VRSREQUEST**, **VRSRESPONSE**) для запросов, у которых MIME-тип равен **application/xml** или **application/json**.
- **Calls** – количество обращений клиентского приложения к серверному приложению через TCP.
- **Class** – имя класса, в котором сгенерировано событие (для события **SYSTEM**).
- **Cluster** – номер основного порта кластера серверов.
- **cn** – количество занятых процессом фрагментов динамической памяти на момент вывода события **MEM**.
- **cn2** – изменение количества занятых процессом фрагментов динамической памяти с момента вывода предыдущего события **MEM**.
- **Component** – имя компоненты платформы, в которой сгенерировано событие (для события **SYSTEM**).
- **Connection** – номер соединения с информационной базой.
- **ConnLimit** – установленное максимальное количество соединений на один рабочий процесс (для события **CLSTR**).

- **Context** – контекст исполнения.
 - **CurExceptions** – количество исключений в процессе за последние 5 минут (для события **ATTN**).
 - **Dbms** – СУБД внешнего источника данных (для события **EDS**). Может принимать значения:
 - **DBMSSQL** – Microsoft SQL Server;
 - **DBOracle** – Oracle Database;
 - **DB2** – IBM DB2;
 - **DBPOSTGRS** – PostgreSQL;
 - **DBMySQL** – MySQL;
 - **DBUnkn** – прочие СУБД.
 - **DBConnStr** – строка соединения с внешним источником данных (для события **EDS**).
 - **DBUser** – имя пользователя СУБД внешнего источника данных (для события **EDS**).
 - **DBConnID** – идентификатор соединения с СУБД внешнего источника данных (для события **EDS**).
 - **dbpid** – строковое представление идентификатора соединения сервера системы «1С:Предприятие» с сервером баз данных в терминах сервера баз данных (для событий **DBMSSQL**, **DBPOSTGRS**, **DB2**, **DBORACLE**).
 - **DeadlockConnectionIntersections** – список пар транзакций, образующих взаимную блокировку (для события **TDEADLOCK**).
 - **Descr** – пояснения к программному исключению. Для события **ATTN** содержит описание события. В зависимости от события запись технологического журнала содержит разный набор свойств:
 - **Process excess memory limit** – рабочий процесс превысил ограничения по памяти. Доступны свойства: **Host**, **PID**, **Name**, **MemSize**, **MaxMemSize**.
 - **Process not respond** – рабочий процесс не отвечает. Доступны свойства: **Host**, **PID**, **Name**, **Timeout**, **Attempts**.
 - **Process has generated too big amount of exceptions** – рабочий процесс формирует очень большое количество ошибок или исключений. Доступны свойства: **Host**, **PID**, **Name**, **CurExceptions**, **AvgExceptions**.
 - **Process will be killed** – процесс будет принудительно завершен. Доступны свойства: **Host**, **PID**, **Name**.
 - **Abandoned process was alive too long time** – рабочий процесс «завис» в памяти. Доступны свойства: **Host**, **PID**, **Name**, **Timeout**.
 - **dumpError** – описание ошибки, произошедшей в процессе построения дампа.
 - **DumpFile** – имя файла с дампом.
 - **Duration** – длительность события в сотнях микросекунд.
 - **Durationus** – длительность события в микросекундах.
 - **Err** – тип консольного сообщения: 0 – информационное; 1 – об ошибке.
 - **Event** – содержит наименование действия, выполняемого кластером серверов (для события **CLSTR**) и определяет наличие других свойств в данном событии. Ниже приведены значения свойства и то, какие свойства будут установлены в этом событии дополнительно:
 - **distrib obsolete** – кеш назначений функциональности кластера устарел в текущем рабочем процессе.
 - **current version older** – активный экземпляр сервиса получил репликацию с новой версией состояния сервиса, он должен стать резервным;
 - **current version newer** – активный экземпляр сервиса получил репликацию со старой версией состояния сервиса и отверг ее.
- Для события **CLSTR**, у которого свойство **Event** равно одному из вышеуказанных значений, имеют смысл следующие свойства события:
- **ServiceName** – наименование сервиса кластера;
 - **Ref** – имя информационной базы;
 - **SessionID** – номер сеанса;
 - **MyVer** – текущая версия состояния сервиса;
 - **SrcVer** – полученная версия состояния сервиса;
 - **NeedResync** – требуется синхронизация данных сервиса (для события **current version older**).
 - **service assign require** – сервис недоступен, требуется переназначение. Для события **CLSTR**, у которого свойство **Event** равно этому значению, имеют смысл следующие свойства события:
 - **ServiceName** – наименование сервиса кластера;
 - **Ref** – имя информационной базы.
 - **working process not found** – не найден рабочий процесс для соединения с информационной базой. Для события **CLSTR**, у которого свойство **Event** равно этому значению, имеют смысл следующие свойства события:
 - **Ref** – имя информационной базы;
 - **SrcURL** – предпочтительный адрес рабочего процесса;
 - **ApplicationExt** – уточнение требования назначения функциональности.
 - **process unavailable** – рабочий процесс не может быть использован для соединения с информационной базой. Для события **CLSTR**, у которого свойство **Event** равно этому значению, имеют смысл следующие свойства события:
 - **Reason** – описывает причину недоступность рабочего процесса:
 - **IBLimit** – достигнуто максимальное количество информационных баз на рабочий процесс;
 - **ConnLimit** – достигнуто максимальное количество соединений на рабочий процесс.

- **IBLimit** – установленное максимальное количество информационных баз на рабочий процесс;
- **ConnLimit** – установленное максимальное количество соединений на рабочий процесс.
- **data replication start** – начало репликации данных из текущего активного экземпляра сервиса в резервный экземпляр. Для события **CLSTR**, у которого свойство **Event** равно этому значению, имеют смысл следующие свойства события:
 - **ServiceName** – наименование сервиса кластера серверов;
 - **Ref** – имя информационной базы;
 - **SessionID** – номер сеанса.
- **destination version older** – репликация была передана в активный экземпляр сервиса со старой версией состояния сервиса;

destination version newer – репликация была передана в активный экземпляр сервиса с новой версией состояния сервиса, репликация была отвергнута и текущий сервис должен стать резервным.

Для события **CLSTR**, у которого свойство **Event** равно одному из вышеуказанных значений, имеют смысл следующие свойства события:

- **ServiceName** – наименование сервиса кластера серверов;
- **Ref** – имя информационной базы;
- **SessionID** – номер сеанса.
- **finish replication** – репликация закончена. Для события **CLSTR**, у которого свойство **Event** равно этому значению, имеют смысл следующие свойства события:
 - **ServiceName** – наименование сервиса кластера серверов;
 - **Ref** – имя информационной базы;
 - **SessionID** – номер сеанса.
- **register rphost** – регистрация рабочих процессов кластера.
- **register rmngr** – регистрация менеджеров кластера.
- **unregister rphost** – отмена регистрации рабочих процессов кластера.
- **unregister rmng** – отмена регистрации менеджеров кластера.
- **main rmngr is down** – ошибка вызова сервиса кластера на главном менеджере. Рабочий процесс должен завершиться. Для события **CLSTR**, у которого свойство **Event** равно этому значению, имеют смысл следующие свойства события:
 - **ServiceName** – имя сервиса, на вызове которого обнаружилось, что главный менеджер кластера недоступен.
- **Exception** – наименование программного исключения.
- **Finish** – причина завершения процесса.
- **FindByString** – имя объекта конфигурации, для которого выполняется ввод по строке.
- **File** – имя файла, в котором было сгенерировано событие (для события **SYSTEM**) или в котором обнаружена проблема при проверке индекса полнотекстового поиска (для события **FTEXTCheck**).
- **Folder** – Каталог, в котором выполняется действие (для события **FTEXTUpd**). Может быть временным каталогом или каталогом расположения файлов с индексом полнотекстового поиска. Свойство формируется только если атрибут **logfiles** элемента **ftextupd** установлено в значение **True** (см. [здесь](#)).
- **Files** – список файлов в формате «имя файла ... размер файла» в каталоге, в котором выполняется действие (для события **FTEXTUpd**). Свойство формируется только если атрибут **logfiles** элемента **ftextupd** установлено в значение **True** (см. [здесь](#)).
- **FilesCount** – количество файлов в каталоге, в котором выполняется действие (для события **FTEXTUpd**). Свойство формируется только если атрибут **logfiles** элемента **ftextupd** установлено в значение **True** (см. [здесь](#)).
- **FilesTotalSize** – размер файлов в каталоге, в котором выполняется действие (для события **FTEXTUpd**). Свойство формируется только если атрибут **logfiles** элемента **ftextupd** установлено в значение **True** (см. [здесь](#)).
- **Func** – наименование выполняемого действия:
 - **connect** – соединение с внешним источником данных.
 - **disconnect** – разрыв соединения с внешним источником данных.
 - **beginTransaction** – начало транзакции (событие типа **SDBL** выводится в журнал в момент начала транзакции и не имеет длительности).
 - **transaction** – начало транзакции (событие типа **SDBL** начинается при начале транзакции, заканчивается при завершении транзакции).
 - **commitTransaction** – фиксация транзакции.
 - **rollbackTransaction** – отмена транзакции.
 - **setRollbackOnly** – установка флажка наличия в транзакции ошибки (ее можно только откатить).
 - **getTransactionSplitter** – получение разделителя итогов.
 - **quickInsert** – быстрая вставка данных в таблицу базы данных.
 - **insertRecords** – добавление записи в таблицу базы данных.
 - **suspendIndexing** – отмена индексирования таблиц базы данных.
 - **resumeIndexing** – восстановление индексирования таблиц базы данных.
 - **holdConnection** – удержание соединения.
 - **saveObject** – сохранение объекта.
 - **restoreObject** – восстановление объекта.

- `readFile` – чтение файла.
- `createFile` – создание файла.
- `deleteFile` – удаление файла.
- `searchFile` – поиск файла.
- `modifyFile` – обновление файла.
- `isProperLocale` – проверка национальных настроек, установленных для базы данных.
- `changeLocale` – изменение национальных настроек базы данных.
- `takeKeyVal` – получение значения ключа записи табличной части.
- `lockRecord` – блокировка записи.
- `serializeTable` – сохранение данных таблицы в файл.
- `deserializeTable` – восстановление данных таблицы базы данных из файла.
- `xlockTables` – установка исключительной блокировки на таблицу.
- `xlockTablesShared` – установка разделяемой блокировки на таблицу.
- `copyMoveFile` – копирование/перемещение фрагмента конфигурации между записями таблиц базы данных.
- `moveFile` – перемещение файла.
- `securedInsert` – вставка записей с наложением ограничений доступа к данным.
- `selectFileName` – выбор имени файла.
- `setSingleUser` – установка монопольного режима.
- `insertIBRegistry` – создание кластера.
- `eraseIBRegistry` – удаление кластера.
- `setRegMultiProcEnable` – установка значения флажка поддержки кластером многих рабочих процессов.
- `setServerProcessCapacity` – установка значения пропускной способности рабочего процесса.
- `agentAuthenticate` – аутентификация администратора центрального сервера.
- `insertAgentUser` – добавление администратора центрального сервера.
- `eraseAgentUser` – удаление администратора центрального сервера.
- `setRegSecLevel` – установка уровня безопасности кластера.
- `setRegDescr` – установка описания кластера.
- `setInfoBaseDescr` – установка описания информационной базы.
- `insertServerProcess` – добавление рабочего процесса.
- `eraseServerProcess` – удаление рабочего процесса.
- `regAuthenticate` – аутентификация администратора кластера.
- `insertRegUser` – добавление администратора кластера.
- `eraseRegUser` – удаление администратора кластера.
- `setServerProcessEnable` – установка значения флажка разрешения запуска рабочего процесса.
- `insertRegServer` – добавление рабочего сервера.
- `eraseRegServer` – удаление рабочего сервера.
- `updateRegServer` – изменение параметров рабочего сервера.
- `authenticateAdmin` – аутентификация администратора информационной базы.
- `createInfoBase` – создание информационной базы.
- `dropInfoBase` – удаление информационной базы.
- `killClient` – разрыв соединения клиента с кластером серверов системы «1С:Предприятие».
- `authenticateSrvrUser` – аутентификация администратора кластера в рабочем процессе.
- `setInfoBaseConnectingDeny` – установка режима блокировки установки соединений с информационной базой.
- `lookupTmpTable` – получение/создание временной таблицы базы данных.
- `returnTmpTable` – освобождение временной таблицы базы данных.
- `start` – начало сеанса (событие типа `SESN` выводится в журнал в момент начала сеанса и не имеет длительности).
- `finish` – окончание сеанса (событие типа `SESN` выводится в журнал в момент окончания сеанса, и длительность события равна длительности всего сеанса).
- `attach` – назначение сеанса соединению (событие типа `SESN` выводится в журнал в момент отмены назначения соединению сеанса). Длительность показывает, сколько времени сеанс был назначен соединению.
- `busy` – сеанс уже назначен соединению (событие типа `SESN` выводится при попытке назначения соединению сеанса, который уже назначен). Не имеет длительности.

- `wait` – ожидание назначения (событие типа `SESN` выводится в момент окончания ожидания назначения сеанса соединению). Длительность события равна времени ожидания соединения. Если соединению назначается сеанс, который уже назначен, то текущий поток текущего соединения ожидает отмены назначения сеанса другому соединению.
- `setSrcProcessName` – означает создание общих данных информационной базы в рабочем процессе и назначение им общего имени. Событие записывается при подключении первого пользователя к информационной базе через данный рабочий процесс или при выполнении динамического обновления конфигурации информационной базы.
- `FtextMgrIndexChanges` – выполняется обновление индекса полнотекстового поиска в файловом варианте информационной базы.
- `FtextMgrRHostIndexChanges` – выполняется обновление индекса полнотекстового поиска в клиент-серверном варианте информационной базы.
- `GetDataForIndexing` – получить список измененных объектов для включения в индекс полнотекстового поиска.
- `IndexObjects` – выполняется индексация порции объектов.
- `AcceptPartialIndex` – принять частичные индексы.
- `MergeSynchro` – объединить файлы с индексами полнотекстового поиска.
- `CheckIndexes` – выполняется проверка индексов полнотекстового поиска.
- `Headers` – HTTP-заголовки запроса/ответа (для событий `VRSREQUEST`, `VRSRESPONSE`).
- `Method` – HTTP-метод обращения к ресурсу (для событий `VRSREQUEST`, `VRSRESPONSE`) или метод объекта `ИнтернетПочта`, во время которого произошла ошибка разбора почтового сообщения (для события `MAILPARSEERR`). При разборе почтового сообщения может принимать значения:
 - `GET` – проблема обнаружена во время исполнения метода `Выбрать()`.
 - `GETHEADERS` – проблема обнаружена во время исполнения метода `ПолучитьЗаголовки()`.
 - `SETRAW` – проблема обнаружена во время исполнения метода `УстановитьИсходныеДанные()`.
- `Host` – имя компьютера.
- `Ib` – имя информационной базы в клиент-серверном варианте.
- `IBLimit` – установленное максимальное количество информационных баз на один рабочий процесс (для события `CLSTR`).
- `IName` – имя передаваемого интерфейса, метод которого вызывается удаленно (для событий `SCALL` и `CALL`).
- `Info` – информация о сбое (для события `FTEXTCheck`).
- `InBytes` – количество данных, прочитанных с диска за время вызова (в байтах).
- `Level` – уровень важности события (для события `SYSTEM`). Возможные значения события приведены в описании элемента `<system>` файла настройки технологического журнала `logcfg.xml` (см. [здесь](#)).
- `Line` – номер строки в файле, в которой было сгенерировано событие `SYSTEM`.
- `Locks` – список управляемых транзакционных блокировок (для события `TLOCK`).
- `Method` – имя вызываемого метода для события `CALL` удаленного вызова метода, отличного от метода `call`. Для удаленного вызова метода `call` в событии `CALL` записываются свойства `Interface` – идентификатор интерфейса и `Method` – номер метода интерфейса.
- `Memory` – объем памяти в байтах, занятой, но не освобожденной за серверный вызов.
- `MemoryPeak` – пиковое значение за вызов памяти в байтах, занятой, но не освобожденной.
- `MessageUid` – уникальный идентификатор почтового сообщения при разборе которого возникла ошибка. Значение равно свойству `Идентификатор` объекта `ИнтернетПочтовоеСообщение`.
- `Name` – имя события.

Для события `ATTN` содержит имя серверного процесса: `rmngr` или `rphost`.

- `NParams` – количество параметров оператора SQL для файлового варианта информационной базы (для события `DBV8DBENG`). Параметры, количество которых указано в данном свойстве, используются для передачи длинных двоичных данных.
- `MaxMemSize` – максимальный объем памяти, которую может занимать процесс, указанный в настройках кластера, в байтах (для события `ATTN`).
- `MemSize` – объем памяти, занимаемой процессом, в байтах (для события `ATTN`).
- `MName` – имя удаленно вызываемого метода (для событий `SCALL` и `CALL`).
- `MyVer` – текущая версия состояния сервера (для события `CLSTR`).
- `NeedResync` – требуется синхронизация данных сервера (для события `CLSTR`, у которого свойство `Event` равно значению `current version older`).
- `OSException` – описание исключения операционной системы.
- `OutBytes` – количество данных, записанных на диск за время вызова (в байтах).
- `PID` – идентификатор процесса операционной системы.
- `Phrase` – текстовая фраза, соответствующая коду состояния (для событий `VRSRESPONSE`).
- `planSQLText` – план запроса, содержащегося в свойстве `sql` (для событий `DBV8DBENG`, `DBMSSQL`, `DBPOSTGRS`, `DB2`, `DBORACLE`, `EDS`).
- `Process` – наименование приложения, как его представляет операционная система (имя файла загрузочного модуля приложения).
- `p:processName` – имя серверного контекста, который обычно совпадает с именем информационной базы.
- `Port` – номер основного сетевого порта процесса.
- `ProcessName` – наименование процесса.
- `Ref` – имя информационной базы.
- `Reason` – причина недоступности рабочего процесса (для события `CLSTR`).

- **Regions** – имена пространств управляемых транзакционных блокировок (для события **TLOCK**).
- **Report** – имя объекта метаданных выполняемого отчета (выполняемого в фоновом задании).
- **Result** – результат проверки файлов индекса полнотекстового поиска: 1 – нет ошибок, 0 – есть ошибки (для события **FTEXTCheck**).
- **Rows** – количество полученных записей базы данных.
- **RowsAffected** – количество измененных записей базы данных.
- **RunAs** – режим запуска процесса (приложение или сервис).
- **Sdbl** – текст запроса на встроенном языке модели базы данных.
- **Separation** – включено или нет разделение (для события **FTEXTCheck**).
- **SepId** – индекс области разделения, если разделение включено (для события **FTEXTCheck**).
- **ServerComputerName** – имя рабочего сервера.
- **ServiceName** – имя сервиса кластера серверов (для события **CLSTR**).
- **SessionID** – номер сеанса, назначенный текущему потоку. Если текущему потоку не назначен никакой сеанс, то свойство не добавляется.
- **Status** – код состояния HTTP (для событий **VRSRESPONSE**).
- **State** – фиксируется начало или завершение операции обновления индекса полнотекстового поиска (для события **FTEXTUpd**).
- **srcProcessName** – записывается при освобождении общих данных информационной базы рабочим процессом. Значением свойства **ProcessName** является имя общих данных на момент освобождения. Значением свойства **srcProcessName** является имя общих данных информационной базы на момент их создания.
- **Sql** – текст оператора SQL.
- **SrcVer** – полученная версия состояния кластера серверов (для события **CLSTR**).
- **SrcURL** – предпочтительный адрес рабочего сервера (для события **CLSTR**).
- **SyncPort** – номер вспомогательного сетевого порта процесса.
- **sz** – объем занятой процессом динамической памяти на момент вывода события **MEM** (в байтах).
- **szd** – изменение объема динамической памяти, занятой процессом, с момента вывода предыдущего события **MEM** (в байтах).
- **t:applicationName** – идентификатор клиентской программы.
- **t:clientID** – идентификатор соединения с клиентом по TCP.
- **t:computerName** – имя клиентского компьютера.
- **t:connectID** – идентификатор соединения с информационной базой.
- **Time** – время вывода записи в технологический журнал (для события **FTEXTUpd**).
- **Timeout** – в зависимости от назначения события **ATTN** (свойство **Descr**) может описывать следующее:
 - Время активности рабочего процесса, который удален из реестра кластера, в секундах.
 - Тайм-аут установки TCP-соединения с рабочим процессом, в миллисекундах.
- **Trans** – идентификатор активности транзакции на момент начала события:
 - 0 – транзакция не была открыта;
 - 1 – транзакция была открыта.
- **Txt** – текст информационного сообщения.

Для события **HASP** данное свойство содержит исходные данные и результат обращения к ключу в следующем формате: **<Операция><Список входных параметров>-><Список выходных параметров>**. При этом:

- **<Операция>** – операция, выполняемая в данном обращении к ключу.
- **<Список входных параметров>** – список входных параметров операции и их значения, перечисленные через запятую.
- **<Список выходных параметров>** – список выходных параметров операции и их значения, перечисленные через запятую.

Полный список операций, их параметров и результатов, содержится в книге **Руководстве разработчика HASP** (<http://sentineldiscussion.safenet-inc.com/viewFile.do?fileId=4316100000036014&forumGroupId=431610000003001>).

Для события **CLSTR** данное свойство содержит значения параметров, участвующих в вычислении доступной производительности рабочего процесса в виде **Параметр:Значение**, разделенные пробелом.

- **URI** – ресурс, к которому производится обращение (для событий **VRSREQUEST**, **VRSRESPONSE**).
- **Usr** – имя пользователя информационной базы (если в информационной базе не определены пользователи, это свойство будет иметь значение **DefUser**). Значение свойства берется из назначенного сеанса.
- **IB** – имя информационной базы (для события **SESN**).
- **Nmb** – номер сеанса (для события **SESN**).
- **Val** – значение, смысл зависит от значения параметра **Func**.
- **Word** – слово, если определено (для события **FTEXTCheck**).
- **WaitConnections** – список соединений, с которыми идет столкновение по управляемым транзакционным блокировкам (для событий **TLOCK** и **TTIMEOUT**).

Используя свойства элемента **<property>**, в технологический журнал можно записывать контекст исполнения. Контекст исполнения может быть двух видов: контекст встроенного языка и интерфейсный контекст. Контекст встроенного языка представляет собой список операторов встроенного языка и содержит в себе:

- название модуля;
- номер строки модуля;
- текстовое представление элемента списка вызова встроенного языка соответствующей строки модуля.

Интерфейсный контекст включает в себя:

- полное имя формы;
- тип активного элемента формы;
- имя активного элемента формы;
- имя кнопки командной панели (если она была нажата);
- действие, выполняемое элементом формы;

Например, контекст встроенного языка в файле технологического журнала может иметь следующий вид:

[Копировать в буфер обмена](#)

```
Документ.ПриходнаяНакладная:23:Движения.УчетНоменклатуры.Записать();
МодульПриложения:18:ПроверитьПодключениеОбработчикаОжидания(Истина);
МодульПриложения:230:ЕслиПолучитьЗначениеПоУмолчанию(глТекущийПользователь,"ИспользоватьНапоминания")
ОбщийМодуль.нпНастройкаПользователей:481:Выборка=Запрос.Выполнить().Выбрать();
```

Интерфейсный контекст в файле технологического журнала может иметь такой вид:

[Копировать в буфер обмена](#)

```
{Документ.Документ1.ФормаСписка}/{ТабличноеПоле:
ДокументСписок}/{ОбновлениеОтображения}
{Документ.Документ1.Форма.ФормаДокумента}/{КоманднаяПанель:
ОсновныеДействияФормы}/{ОсновныеДействияФормыOK}
{Документ.Документ1.Форма.ФормаДокумента}/{Кнопка:
Кнопка1}/{Нажатие}
```

Чтобы включить запись контекста, нужно среди фильтров свойств записать элемент `<property name="Context">` или элемент `<property name="all">`.

Если нужно записывать события **SDBL** (SDBL-запросы) и **DBMSSQL** (операторы SQL к СУБД MS SQL Server) с контекстом исполнения, то содержимое файла настройки технологического журнала будет выглядеть следующим образом:

[Копировать в буфер обмена](#)

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
  <log location="c:\v8\logs" history="1">
    <event>
      <eq property="name" value="sdbl"/>
    </event>
    <event>
      <eq property="name" value="dbmssql"/>
    </event>
    <property name="context">
    </property>
  </log>
</config>
```

Чтобы записывать события **SDBL** (SDBL-запросы) и **DBMSSQL** (операторы SQL к СУБД MS SQL Server) без контекста исполнения, файл настройки технологического журнала нужно заполнить следующим образом:

[Копировать в буфер обмена](#)

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
  <log location="c:\v8\logs" history="1">
    <event>
      <eq property="name" value="sdbl"/>
    </event>
    <event>
      <eq property="name" value="dbmssql"/>
    </event>
  </log>
</config>
```

Чтобы записывать события **SDBL** (SDBL-запросы) и **DBMSSQL** (операторы SQL к СУБД MS SQL Server) без контекста исполнения, но со всеми другими свойствами, файл настроек должен содержать:

[Копировать в буфер обмена](#)

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
  <log location="c:\v8\logs" history="1">
    <event>
      <eq property="name" value="sdbl"/>
    </event>
    <event>
      <eq property="name" value="dbmssql"/>
    </event>
    <property name="all">
    </property>
    <property name="context">
      <eq property="name" value="" />
    </property>
  </log>
</config>
```

Для того чтобы записывать события **SDBL** (SDBL-запросы) с контекстом исполнения и **DBMSSQL** (операторы SQL к СУБД MS SQL Server) без контекста исполнения, содержимое файла настройки должно иметь следующий вид:

[Копировать в буфер обмена](#)

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
  <log location="c:\v8\logs" history="1">
    <event>
      <eq property="name" value="sdbl" />
    </event>
    <event>
      <eq property="name" value="dbmssql"/>
    </event>
    <property name="context">
    <event>
      <eq property="name" value="sdbl"/>
    </event>
  </property>
  </log>
</config>
```

Наличие элемента `<property name="Context">` означает, что для записываемых в журнал событий при выполнении условий, указанных в данном элементе, будет записана информация о контексте. После этого в каждое событие технологического журнала будет добавлена информация о контексте исполнения в текущем процессе, а после

события будет добавлено мгновенное событие, несущее информацию о контексте исполнения клиентского процесса.

В технологический журнал могут быть записаны сообщения об исключительных ситуациях, связанных с менеджером блокировок. Для этого файл конфигурации должен иметь примерно следующий вид:

[Копировать в буфер обмена](#)

```
<config xmlns="http://v8.lc.ru/v8/tech-log">
  <log location="c:\v8\logs" history="7">
    <event>
      <eq property="name" value="excp" />
    </event>
    <event>
      <eq property="name" value="tlock" />
      <gt property="duration" value="100000" />
    </event>
    <property name="all" />
    <property name="context">
      <event>
        <eq property="name" value="" />
      </event>
    </property>
  </log>
  <dump location="c:\v8\dumps" create="1" type="2" />
</config>
```

В приведенном примере будут регистрироваться все исключительные ситуации, связанные с блокировками (в частности, **DEADLOCK** – взаимные блокировки соединений и **TIMEOUT** – истечение предопределенного времени, при этом в обоих случаях в текст сообщения об исключительной ситуации включается номер соединения, которое вызвало эту исключительную ситуацию), и ожидания, превысившие 10 секунд. При этом будет записана информация по всем свойствам, кроме **Context**.

3.18.1.4. Элемент <dump>

Элемент <dump> определяет параметры дампа, создаваемого при аварийном завершении приложения. Чтобы отключить запись дампов, нужно в элементе <dump> установить значение параметра `create = "0"` или `create = "false"`. Если элемент <dump> отсутствует, то для записи дампов будет использоваться каталог `%USERPROFILE%\Local Settings\Application Data\1C\1Cv8\Dumps` (для ОС Windows).

ВНИМАНИЕ! Для ОС Linux настройка формирования дампов выполняется средствами ОС. Поэтому элемент <dump> игнорируется. Подробнее о настройке формирования дампов в ОС Linux см. [здесь](#).

Атрибуты элемента:

Атрибут location

Имя каталога, в который будут помещаться файлы дампов.

ПРИМЕЧАНИЕ. В элементах <log>, <dump> и <defaultlog> следует указывать разные каталоги в атрибутах `location`.

Атрибут create

Создавать или не создавать файл дампа.

- 0 ("false") – не создавать;
- 1 ("true") – создавать.

Атрибут type

Тип дампа, произвольная комбинация приведенных ниже флажков, представленная в десятичной или шестнадцатеричной системе (сложение значений флагов). Представление в шестнадцатеричной системе должно начинаться с символа 'x', например, x0002.

Доступны следующие значения:

- 0 (x0000) – минимальный;
- 1 (x0001) – дополнительный сегмент данных;
- 2 (x0002) – содержимое всей памяти процесса;
- 4 (x0004) – данные хэндлов;
- 8 (x0008) – оставить в дампе только информацию, необходимую для восстановления стеков вызовов;
- 16 (x0010) – если стек содержит ссылки на память модулей, то добавить флажок флаг 64 (0x0040);
- 32 (x0020) – включить в дамп память из-под выгруженных модулей;
- 64 (0x0040) – включить в дамп память, на которую есть ссылки;
- 128 (x0080) – добавить в дамп подробную информацию о файлах модулей;
- 256 (0x0100) – добавить в дамп локальные данные потоков;
- 512 (0x0200) – включение в дамп памяти из всего доступного виртуального адресного пространства.

СОВЕТ. Для большинства случаев достаточно в качестве значения атрибута `type` использовать значение 3, например, `type="3"`.

Атрибут prtscrn

Создавать или нет файл копии экрана при аварийном завершении клиентской части системы «1С:Предприятие». Имя файла совпадает с именем дампа, но имеет расширение `png`. Файлы копий экрана создаются в том же каталоге, что и дампы (см. атрибут `location`).

- 0 ("false") – не создавать;
- 1 ("true") – создавать.

При аварийном завершении программы «1С:Предприятие» система выдает диалог с информацией о процессе записи дампа, который автоматически закрывается после завершения записи дампа.

Атрибут externaldump только для ОС Windows

Управляет формированием дампа аварийного завершения работы, в том случае, если система функционирует под управлением ОС Windows. Атрибут может принимать следующие значения:

- 0 ("false") – дамп формируется процессом, который завершается аварийно (значение по умолчанию).

- 1 ("true") – дамп формируется внешним приложением [dumper.exe](#), входящий в комплект поставки «1С:Предприятия». При использовании внешней программы исключена возможность зависания в процессе создания дампа.

Если внешняя программа не обнаружена или в процессе ее запуска выявлены проблемы, то будет использован обычный режим создания дампов (с помощью аварийно завершающегося процесса).

Рекомендуется использовать внешнюю программу формирования дампов для серверов «1С:Предприятия», которые работают без ежесуточного обслуживания.

3.18.1.5. Элемент <leaks>

Элемент <leaks> устанавливает отслеживание утечек памяти, вызванных проблемами кода конфигурации. По умолчанию отслеживание утечек выключено и не влияет на производительность системы.

Для того чтобы включить сбор данных об утечках, в файле [logcfg.xml](#) следует добавить элемент <leaks>: <leaks collect="1"> или <leaks collect="true">.

Чтобы отключить отслеживание утечек памяти, следует изменить элемент <leaks>: <leaks collect="0"> или <leaks collect="false">.

Если отслеживание утечек включено, то в разрезе пользователей контролируется создание и удаление следующих объектов:

- Форма,
- УправляемаяФорма,
- ФиксированнаяСтруктура,
- ФиксированноеСоответствие,
- ДанныеФормыСтруктура,
- ДанныеФормыКоллекция,
- ДанныеФормыСтруктураСКоллекцией,
- ДанныеФормыЭлементКоллекции,
- ДанныеФормыДерево,
- ДанныеФормыКоллекцияЭлементовДерева,
- ДанныеФормыЭлементДерева,
- РегистрБухгалтерийМенеджер,
- РегистрБухгалтерииНаборЗаписей,
- ПланСчетовМенеджер,
- ПланСчетовОбъект,
- ПланОбменаМенеджер,
- ПланОбменаОбъект,
- ХранилищаНастроекМенеджер,
- РегистрНакопленияМенеджер,
- РегистрНакопленияНаборЗаписей,
- ПланВидовХарактеристикМенеджер,
- ПланВидовХарактеристикОбъект,
- КонстантаМенеджер,
- ДокументМенеджер,
- ДокументОбъект,
- ПеречислениеМенеджер,
- ВнешняяОбработка,
- ВнешнийОтчет,
- РегистрСведенийМенеджер,
- РегистрСведенийНаборЗаписей,
- ОбработкаМенеджер,
- Обработка,
- СправочникМенеджер,
- СправочникОбъект,
- ОтчетМенеджер,
- Отчет,
- ПоследовательностьНаборЗаписей,
- БизнесПроцессМенеджер,
- БизнесПроцессОбъект,
- ЗадачаМенеджер,

- ЗадачаОбъект,
- ПланВидовРасчетаМенеджер,
- ПланВидовРасчетаОбъект,
- РегистрРасчетаМенеджер,
- РегистрРасчетаНаборЗаписей,
- ПерерасчетНаборЗаписей,
- COMSafeArray,
- КлючиЗначение,
- Массив,
- ФиксированныйМассив,
- Соответствие,
- Структура,
- ЭлементСпискаЗначений,
- СписокЗначений,
- ТаблицаЗначений,
- СтрокаТаблицыЗначений,
- ДеревоЗначений,
- СтрокаДереваЗначений.

Утечки отслеживаются между начальной и конечной контрольной точкой в коде. В начальной контрольной точке выполняется очистка данных об утечках для текущего пользователя. В конечной контрольной точке выполняется формирование и вывод в технологический журнал события LEAKS, в котором для каждого неосвобожденного экземпляра объекта будет указан стек встроенного языка на момент его создания.

В качестве контрольных точек могут использоваться:

- начало и конец исполнения встроенного языка на клиенте или на сервере;
- вызов процедуры/функции встроенного языка и возврат из процедуры/функции;
- начало выполнения одной строки кода встроенного языка и окончание выполнения другой строки кода встроенного языка.

Начальную и конечную контрольную точку определяет элемент `<point>`. При этом вложение контрольных точек друг в друга допускается, но игнорируется – подсчет утечек ведется только по внешним контрольным точкам. Например, если в процессе исполнения кода конфигурации были пройдены контрольные точки `Начальная1`, `Начальная2`, `Конечная1`, `Конечная2`, то утечки будут отслеживаться между точками `Начальная1` и `Конечная2`.

Элемент `<point>` может иметь один из следующих форматов:

```
<point call=«client»/>, <point call=«server»/>
```

Определяет контрольные точки в начале/в конце исполнения встроенного языка на клиенте или на сервере, т. е.: начальная точка будет установлена в начале исполнения встроенного языка на сервере/клиенте, конечная точка – в конце исполнения встроенного языка на сервере/ клиенте.

```
<proc=«ИмяМодуля»/«ИмяМетода» />
```

Определяет контрольные точки при вызове и возврате определенного метода встроенного языка. `<ИмяМодуля>` – содержит полное имя объекта метаданных, которому принадлежит модуль (без названия конфигурации). В таком же формате имена модулей показывает отладчик. `<ИмяМетода>` содержит наименование метода. Если аргумент `<ИмяМетода>` не установлен, то контрольные точки будут определены в начале/окончании исполнения тела модуля. Примеры имен модулей:

- МодульСеанса.Модуль – модуль сеанса;
- МодульПриложения.Модуль – модуль приложения;
- МодульУправляемогоПриложения.Модуль – модуль управляемого приложения;
- МодульВнешнегоСоединения.Модуль – модуль внешнего соединения;
- ОбщийМодуль.Глобальный.Модуль – общий модуль Глобальный;
- Справочник.Контрагенты.МодульОбъекта – модуль элемента справочника Контрагенты;
- Обработка.Обработка1.Форма.Формал.Форма – модуль формы Формал обработки Обработка1;
- Обработка.Обработка2.Форма.ОсновнаяФорма.Форма – модуль формы ОсновнаяФорма обработки Обработка2.

```
<point on=«ИмяМодуля»/«НомерСтроки» Off=«ИмяМодуля»/«НомерСтроки» />
```

Определяет начальную и конечную контрольные точки явным указанием строк кода. Начальная контрольная точка соответствует началу исполнения кода строки, указанной в атрибуте `On`. Конечная контрольная точка соответствует окончанию исполнения кода строки, указанной в атрибуте `Off`. Нумерация строк начинается с 1. Если начальная контрольная точка достигнута на сервере, то и конечная контрольная точка должна быть достигнута на сервере. Конечной контрольной точкой не может быть последняя строка кода процедуры, функции или тела модуля.

Пример элемента `<leaks>`:

```
<leaks collect="1">
  <point call="client" />
  <point call="server" />
  <point proc="МодульПриложения" />
  <point proc="ОбщийМодуль.ОбработкаПодключений.Модуль/НаСервереБезУтечки" />
  <point on="ОбщийМодуль.Сервисы.Модуль/9" off="ОбщийМодуль.Сервисы.Модуль/11" />
</leaks>
```

[Копировать в буфер обмена](#)

В данном случае сбор данных об утечках включен. Контрольные точки установлены:

- в начале и в конце исполнения встроенного языка на клиенте;

- в начале и в конце исполнения встроенного языка на сервере;
- в начале и при окончании исполнения тела модуля приложения;
- при вызове и возврате метода `НаСервереБезУтечки()` из общего модуля `ОбработкаПодключений`;
- на девятой и одиннадцатой строках общего модуля `Сервисы`.

Предположим, процедура со следующим текстом вызывает утечку памяти:

```
Процедура НаСервереСУтечкой() Экспорт
М=НовыйМассив;
М.Добавить(НовыйМассив);
М[0].Добавить(НовыйМассив);
М[0][0].Добавить(М);
КонецПроцедуры
```

[Копировать в буфер обмена](#)

Чтобы ее обнаружить, можно включить отслеживание утечек в технологическом журнале при помощи следующей настройки:

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
<log location="C:\ProgramFiles\1cv8\logs" history="24">
<event>
<eq property="name" value="call"/>
</event>
<event>
<eq property="name" value="leaks"/>
</event>
<property name="all">
</property>
</log>
<leaks collect="1">
<point call="server"/>
</leaks>
</config>
```

[Копировать в буфер обмена](#)

Тогда при вызове сервера или выполнении регламентного задания, если утечки не происходит, фрагмент технологического журнала будет выглядеть следующим образом:

```
59:44.4562-2840,CALL,5,process=rphost,p:processName=t76346,t:clientID=428,t:applicationName=JobScheduler,Func=Execute,Module=ОбщийМодуль2,Meth=Регламентное;
59:49.4581-2700,CALL,5,process=rphost,p:processName=t76346,t:clientID=430,t:applicationName=JobScheduler,Func=Execute,Module=ОбщийМодуль2,Meth=Регламентное;
```

[Копировать в буфер обмена](#)

А если происходит, то таким образом:

```
59:48.4768-2885,CALL,5,process=rphost,p:processName=t76346,t:clientID=429,t:applicationName=JobScheduler,Func=Execute,Module=ОбщийМодуль2,Meth=Регламентное;
59:48.4769-0,LEAKS,5,process=rphost,Descr='
Array:
ОбщийМодуль.ОбщийМодуль2:2:НаСервереСУтечкой();
ОбщийМодуль.ОбщийМодуль1:4:М[0].Добавить(НовыйМассив);

Array:
ОбщийМодуль.ОбщийМодуль2:2:НаСервереСУтечкой();
ОбщийМодуль.ОбщийМодуль1:2:М=НовыйМассив;

Array:
ОбщийМодуль.ОбщийМодуль2:2:НаСервереСУтечкой();
ОбщийМодуль.ОбщийМодуль1:3:М.Добавить(НовыйМассив);
```

[Копировать в буфер обмена](#)

В приведенном фрагменте при выполнении в качестве регламентного задания (`t:applicationName=JobScheduler, Func=Execute`) метода `РегламентноеЗаданиеСУтечкой()` модуля `ОбщийМодуль2` были созданы и не освобождены три объекта `Array` (`Массив`). При этом указаны стеки вызовов встроенного языка в момент создания каждого из объектов.

3.18.1.6. Элемент `<mem>`

Если элемент `<mem>` присутствует, то процессы сервера «1С:Предприятия» ведут подсчет:

- количества выделенных и не освобожденных фрагментов памяти;
- суммарного объема выделенных и не освобожденных фрагментов памяти.

Если между моментами времени, когда процесс сервера не выполнял ни одного вызова и ни одного регламентного задания, увеличилось количество выделенных, но не освобожденных фрагментов памяти, то формируется событие типа `MEM` со следующими свойствами:

- `sz` – общий объем выделенных процессом, но не освобожденных фрагментов памяти;
- `szd` – его изменение с момента вывода предыдущего события `MEM`;
- `cn` – общее количество выделенных процессом, но не освобожденных фрагментов памяти;
- `cmd` – его изменение с момента вывода предыдущего события `MEM`.

Длительность события `MEM` равна периоду времени между последним и предпоследним моментами времени, когда процесс сервера не выполнял ни одного вызова и ни одного регламентного задания. Именно в течение этого времени увеличилось количество занятых процессом фрагментов памяти.

ВНИМАНИЕ! Указание элемента `<mem>` в конфигурационном файле технологического журнала несколько снижает производительность «1С:Предприятия», особенно при параллельной работе нескольких пользователей.

Например, при следующей конфигурации объем распределенной памяти не собирается и события `MEM` не выводятся:

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
<log location="C:\ProgramFiles\1Cv8\logs" history="24">
<event>
<eq property="name" value="mem"/>
</event>
<property name="all"/>
</log>
</config>
```

[Копировать в буфер обмена](#)

Следующая конфигурация технологического журнала собирает объем распределенной памяти и при его росте выводит события `MEM`:

```
<config xmlns="http://v8.1c.ru/v8/tech-log">
<log location="C:\ProgramFiles\1Cv8\logs" history="24">
<event>
<eq property="name" value="mem"/>
</event>
```

[Копировать в буфер обмена](#)

```

    <property name="all"/>
  </log>
</mem/>
</config>

```

3.18.1.7. Элемент <ftextupd>

Элемент <ftextupd> включает формирование расширенной информации о процессе обновления индексов полнотекстового поиска (событие FTTEXTUPd). Если элемент отсутствует в файле, то расширенная информация не включается в технологический журнал.

Атрибуты элемента:

Атрибут *logfiles*

Наличие расширенной информации в событии FTTEXTUPd:

- 0 ("false") – не включать;
- 1 ("true") – включать.

3.18.1.8. Элемент <plansql>

Если элемент <plansql> присутствует, то будет включен сбор планов запросов, которые генерируют СУБД при выполнении запросов «1С:Предприятия». Сами планы запросов расположены в свойстве <planSQLText> событий, связанных с исполнением запросов конкретной СУБД (см. [здесь](#)).

СОВЕТ. Рекомендуется вместе со свойством <planSQLText> включать в состав регистрируемых свойств также и свойство <SQL>, содержащее запрос, план которого будет зарегистрирован.

[Копировать в буфер обмена](#)

```

<?xml version="1.0"?>
<config xmlns="http://v8.1c.ru/v8/tech-log">
  <log location="c:\log" history="24">
    <event>
      <eq property="name" value="dbmssql"/>
    </event>
    <property name="sql"/>
    <property name="plansqltext"/>
  </log>
</plansql />
</config>

```

В приведенном выше примере для СУБД Microsoft SQL Server (выражение <eq property="name" value="dbmssql"/>) включен сбор планов запросов (элемент <plansql />) и их запись в журнал регистрации (выражение <property name="plansqltext"/>) вместе с текстами самих запросов (выражение <property name="sql"/>) на языке запросов системы «1С:Предприятие».

ВНИМАНИЕ! Получение планов запросов замедляет выполнение запросов к СУБД. Для некоторых СУБД такое замедление может быть существенным. Не следует получать планы запросов в обычном режиме работы «1С:Предприятия». Планы запросов следует собирать только при анализе производительности запросов.

ПРИМЕЧАНИЕ. Получение планов запросов для внешних источников данных (событие <EDS>) возможно только в том случае, если в качестве СУБД внешнего источника данных выступают IBM DB2, Microsoft SQL Server, Oracle Database, PostgreSQL. Для прочих СУБД планы запросов не получаются, в технологический журнал записывается только текст запроса.

3.18.1.8.1. Информация по планам запросов СУБД

Информация по работе с планами запросов конкретных СУБД приводится в документации к этим СУБД:

- Microsoft SQL Server 2000:
 - [http://msdn.microsoft.com/en-us/library/aa178417\(v=SQL.80\).aspx](http://msdn.microsoft.com/en-us/library/aa178417(v=SQL.80).aspx);
 - [http://msdn.microsoft.com/en-us/library/aa259207\(v=SQL.80\).aspx](http://msdn.microsoft.com/en-us/library/aa259207(v=SQL.80).aspx);
 - <http://msdn.microsoft.com/en-us/library/Aa259203.aspx>.
- Microsoft SQL Server 2005:
 - [http://msdn.microsoft.com/en-US/library/ms176005\(v=SQL.90\).aspx](http://msdn.microsoft.com/en-US/library/ms176005(v=SQL.90).aspx);
 - [http://msdn.microsoft.com/en-us/library/ms187735\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms187735(SQL.90).aspx).
- Microsoft SQL Server 2008:
 - [http://msdn.microsoft.com/en-us/library/ms176005\(v=SQL.100\).aspx](http://msdn.microsoft.com/en-us/library/ms176005(v=SQL.100).aspx);
 - [http://msdn.microsoft.com/en-us/library/ms187735\(v=SQL.100\).aspx](http://msdn.microsoft.com/en-us/library/ms187735(v=SQL.100).aspx).
- Microsoft SQL Server 2008 R2:
 - [http://msdn.microsoft.com/en-us/library/ms176005\(v=SQL.105\).aspx](http://msdn.microsoft.com/en-us/library/ms176005(v=SQL.105).aspx);
 - [http://msdn.microsoft.com/en-us/library/ms187735\(v=SQL.105\).aspx](http://msdn.microsoft.com/en-us/library/ms187735(v=SQL.105).aspx).
- Microsoft SQL Server 2012:
 - [http://msdn.microsoft.com/en-us/library/ms187735\(v=SQL.11\).aspx](http://msdn.microsoft.com/en-us/library/ms187735(v=SQL.11).aspx).
- PostgreSQL 8.1 Windows/Linux:
 - <http://www.postgresql.org/docs/8.1/static/performance-tips.html>.
- PostgreSQL 8.2 Windows/Linux:
 - <http://www.postgresql.org/docs/8.2/static/using-explain.html>.
- PostgreSQL 8.3 Windows/Linux:
 - <http://www.postgresql.org/docs/8.3/static/using-explain.html>.
- PostgreSQL 8.4 Windows/Linux:
 - <http://www.postgresql.org/docs/8.4/static/using-explain.html>.
- PostgreSQL 9.0 Windows/Linux:

- <http://www.postgresql.org/docs/9.0/static/using-explain.html>.
- PostgreSQL 9.1 Windows/Linux:
 - <http://www.postgresql.org/docs/9.1/static/using-explain.html>.
- PostgreSQL 9.2 Windows/Linux:
 - <http://www.postgresql.org/docs/9.2/static/using-explain.html>.
- IBM DB2 версия 9.1 Windows/Linux:
 - <http://publib.boulder.ibm.com/infocenter/db2luw/v9r1/topic/com.ibm.db2.udb.admin.doc/doc/c0005739.htm>;
 - <http://publib.boulder.ibm.com/infocenter/db2luw/v9r1/topic/com.ibm.db2.udb.admin.doc/doc/r0008441.htm>.
- IBM DB2 версия 9.5 Windows/Linux:
 - <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.admin.explain.doc/doc/r0052023.html>;
 - <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.admin.perf.doc/doc/c0005739.html>;
 - <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.sql.ref.doc/doc/r0008441.html>.
- IBM DB2 версия 9.7 Windows/Linux:
 - <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.explain.doc/doc/r0052023.html>;
 - <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.perf.doc/doc/c0005739.html>;
 - <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.sql.ref.doc/doc/r0008441.html>.
- Oracle DB2 10g R2:
 - http://download.oracle.com/docs/cd/B19306_01/server.102/b14211/optimops.htm.
- Oracle DB2 11g R1:
 - http://download.oracle.com/docs/cd/B28359_01/server.111/b28274/optimops.htm.

Для PostgreSQL и Oracle Database формат плана запроса в точности соответствует формату, описанному в документации на соответствующую СУБД. Формат планов запросов для Microsoft SQL Server и IBM DB2 упрощен относительно оригинального формата. При этом сохранены оригинальные названия полей. Интерпретация информации в этих полях также выполняется в соответствии с информацией на конкретную СУБД. Эти изменения отражены в следующих разделах.

3.18.1.8.2. Формат плана запроса MS SQL Server

Поле `planSQLText` для СУБД Microsoft SQL Server состоит из нескольких записей (строк), каждая из которых состоит из следующих полей (в терминах СУБД), идущих в порядке описания:

- `Rows`,
- `Executes`,
- `EstimateRows`,
- `EstimateIO`,
- `EstimateCPU`,
- `AvgRowSize`,
- `TotalSubtreeCost`,
- `EstimateExecutions`,
- `StmtText`.

Поля разделены запятыми. Последнее поле описания плана запроса (`StmtText`) следует читать до конца строки, игнорируя возможные символы ";". Строки разделены переводом строки.

3.18.1.8.3. Формат плана запроса IBM DB2

Поле `planSQLText` для СУБД IBM DB2 состоит из нескольких записей (строк), каждая из которых состоит из следующих полей, идущих в порядке описания. Имена полей в точности соответствуют полям из таблиц объяснений (`explain tables`), т. е. текст `IO_COST (EXPLAIN_OPERATOR)` означает, что в план запроса будет помещено поле `IO_COST` из таблицы объяснения `EXPLAIN_OPERATOR`:

- `OPERATOR_TYPE (EXPLAIN_OPERATOR)`,
- `TOTAL_COST (EXPLAIN_OPERATOR)`,
- `STREAM_COUNT (EXPLAIN_STREAM)`,
- `IO_COST (EXPLAIN_OPERATOR)`,
- `CPU_COST (EXPLAIN_OPERATOR)`,
- `COMM_COST (EXPLAIN_OPERATOR)`,
- `BUFFERS (EXPLAIN_OPERATOR)`,
- `PREDICATE_TEXT (EXPLAIN_PREDICATE)`.

Поля разделены запятыми. Последнее поле описания плана запроса (`PREDICATE_TEXT`) следует читать до конца строки, игнорируя возможные символы ";". Строки разделены переводом строки.

В конце описания плана запроса добавляется строка, начинающаяся с текста `Optimized query:`, которая содержит в себе текст запроса, сформированный оптимизатором СУБД. Оригинальный текст запроса приведен в свойстве SQL события технологического журнала. Запрос завершается концом строки. Идентификаторы из оптимизированного варианта запроса используются в данных, размещенных в колонке `PREDICATE_TEXT`.

3.18.1.8.4. Формат плана запроса файловой версии

План запроса файловой версии имеет следующий формат:

<План запроса>

```
[CONST <Условия>]
<Поля списка выборки>
[<Описание источника> [<Описание связи> [...]]]
[WITHOUT DUPLICATES]
[GROUPING]
[SORTING [CUTTING TOP]]
[UNION [ALL] <План запроса>]
```

[Копировать в буфер обмена](#)

В данном описании:

- **WITHOUT DUPLICATES** – обозначает, что нужно получать данные без дубликатов.
- **GROUPING** – обозначает, что нужно выполнить группировку результата.
- **SORTING** – обозначает, что нужно выполнить сортировку результата.
- **CUTTING TOP** – обозначает, что после сортировки будет получена только часть записей.

<Условия>

```
WHERE [(POST) | (END)] <Условие> [AND <Условие> [...]]
```

[Копировать в буфер обмена](#)

В данном описании:

- **(POST)** – обозначает, что условия проверяются после выполнения соединения.
- **(END)** – обозначает, что условия проверяются после того, как будут выполнены соединения между всеми таблицами.

<Поля списка выборки>

```
Fields:(<Выражение из списка выборки> [, <Выражение из списка выборки>])
```

[Копировать в буфер обмена](#)

<Описание сканирования>

```
{(NOT SCAN) | {FULL SCAN} | {DISTINCT SCAN} | {RANGE SCAN}} {UNTIL FIRST NOT NULL} [USING [REVERSE] INDEX (<Имя индекса>)] (<Количество используемых полей и
```

[Копировать в буфер обмена](#)

В данном описании:

- **NOT SCAN** – обозначает, что содержимое таблицы не будет сканироваться.
- **FULL SCAN** – обозначает, что содержимое таблицы будет просканировано полностью.
- **DISTINCT SCAN** – обозначает, что будут обходиться различные значения по индексу.
- **UNTIL FIRST NOT NULL** – обозначает, что записи будут сканироваться до тех пор, пока не будет получена первая запись со значением не **NULL**.
- **USING INDEX** – обозначает, что для обхода будет использоваться индекс.
- **REVERSE** – обозначает, что индекс будет использоваться в обратной последовательности.

<Описание источника>

```
<Имя таблицы> [(TWICE)] <Описание сканирования> |
{NESTED SELECT <Описание сканирования> (<План запроса>)}
<Условия>
```

[Копировать в буфер обмена](#)

<Описание связи>

```
{ { NESTED [OUTER] LOOP <Имя таблицы> [(TWICE)] <Описание сканирования> } |
{ NESTED [OUTER] LOOP BY SELECT <Описание сканирования> (<План запроса>)} }
<Условия> [<Условия>]
```

[Копировать в буфер обмена](#)

В данном описании:

- **(TWICE)** – обозначает, что таблица используется в запросе несколько раз.
- **NESTED LOOP** – обозначает, что для каждой записи таблицы слева будет выполнен цикл обхода записей таблицы справа.
- **OUTER** – обозначает, что если в правой таблице не будет найдено подходящей по условию связи записи, то запись целиком не пропадет.

3.18.1.9. Элемент <defaultlog>

Элемент <defaultlog> определяет параметры технологического журнала по умолчанию. Данный журнал имеет фиксированный фильтр событий, определяемых системой «1С:Предприятие». Этот фильтр не может быть изменен и его можно представить следующим файлом настройки:

```
<log location="C:\Users\<UserName>\AppData\Local\1C\1Cv8\logs" history="24" >
  <event>
    <eq property="name" value="system"/>
    <eq property="level" value="error"/>
  </event>
  <property name="all"/>
</log>
```

[Копировать в буфер обмена](#)

В этот журнал записываются события, критические с точки зрения функционирования системы. Состав событий не документируется. Детализация формирования событий настраивается с помощью элемента <system>.

Атрибуты элемента:

Атрибут location

Имя каталога, в котором будет размещаться технологический журнал по умолчанию. Если атрибут не задан, то технологический журнал по умолчанию сохраняется в следующие каталоги:

- OC Windows:

- Windows XP: %USERPROFILE%\Local Settings\Cv8\logs.
- Windows Vista (и старше): %LOCALAPPDATA%\Cv8\logs.
- ОС Linux: \$HOME/.1cv8/logs.

ВНИМАНИЕ! Необходимо иметь в виду, что каталог технологического журнала (любого) не предназначен для хранения в нем файлов, которые не относятся к технологическому журналу. Поэтому не следует размещать в нем дампы или использовать каталог, который может содержать файлы, не относящиеся к технологическому журналу «1С:Предприятия». Если в каталоге, который указан в качестве каталога технологического журнала, имеются посторонние файлы, то указание каталога считается неверным, и технологический журнал не создается.

ПРИМЕЧАНИЕ. В элементах `<log>`, `<dump>` и `<defaultlog>` следует указывать разные каталоги в атрибутах `location`.

Атрибут `history`

Количество часов, через которое информация будет удаляться из технологического журнала. Если значение этого атрибута установить в значение 0 – будет отключена запись технологического журнала по умолчанию.

Значение по умолчанию: 24.

В отличие от файлов технологического журнала, файлы технологического журнала по умолчанию формируются только тогда, когда возникает соответствующее событие.

3.18.1.10. Элемент `<system>`

Для управления формированием событий `SYSTEM` в технологическом журнале предназначен элемент `<system>`. В файле настройки технологического журнала (`logcfg.xml`) может не содержать такого элемента, а также содержать один или несколько таких элементов.

Если элемент `<system>` отсутствует в файле `logcfg.xml`, то технологический журнал по умолчанию настроен следующим образом – уровень формирования системных событий для всех компонентов системы определен как `Error`.

События `SYSTEM` будут одновременно попадать во все настроенные технологические журналы (в том числе и в технологический журнал по умолчанию).

Атрибуты элемента:

Атрибут `level`

Задаёт минимальное значение уровня формируемых системой событий. Возможные значения (в порядке увеличения значимости):

- `Trace` – максимально подробный уровень.
- `Debug` – уровень отладочной информации. Предназначен для событий необходимых в целях отладки механизмов платформы или расследования особо трудных в обнаружении ошибок.
- `Info` – информационный уровень. Предназначен для событий, свидетельствующих о нормальном функционировании того или иного механизма платформы.
- `Warning` – уровень предупреждений. Предназначен для событий, информирующих о возникновении нештатных, но не критичных, с точки зрения механизма платформы, ситуаций.
- `Error` – уровень ошибок. Предназначен для событий, информирующих о возникновении ошибочных, с точки зрения механизма платформы, ситуаций.
- `None` – отключение фиксации системных событий.

Задание этого атрибута приведет к тому, что система «1С:Предприятие» не будет формировать события, не соответствующие заданному уровню.

Так, если в файле `logcfg.xml`, находится конструкция вида `<system level="info" />`, то это будет означать, что система «1С:Предприятие» будет формировать события с уровнями `Info`, `Warning` и `Error`.

Атрибут `component`

Определяет имя компоненты, для которой настраивается формирование системных событий. Имя компоненты является регистрозависимым.

Атрибут `class`

Определяет имя класса, для которого настраивается формирование системных событий. Имя класса является регистрозависимым.

В качестве примера, рассмотрим ситуацию, когда файл `logcfg.xml` содержит следующий фрагмент:

```
<system level="info"/>
<system level="debug" class="core::FileSystem" />
<system level="warning" component="core82" />
```

[Копировать в буфер обмена](#)

Эта настройка означает:

- Следует формировать события `SYSTEM` с уровнем `Info` (и выше) для всех объектов системы.
- Однако, для класса `core::FileSystem` следует формировать события с уровнем `Debug`.
- Для всех классов компоненты `core82` следует формировать события с уровнем `Warning` и выше.

3.18.2. Вывод контекстов исключений

Контекст исключения представляет собой последовательность событий технологического журнала типа `EXCPCTX`. Каждое событие типа `EXCPCTX` является одним из длительных событий, которые начались, но не закончились в момент возникновения нештатной ситуации в работе системы «1С:Предприятие». При этом события выводятся в порядке убывания уровня вложенности. Тип события, исходного для события `EXCPCTX`, становится значением свойства `SrcName` события `EXCPCTX`.

Контекст исключения выводится в технологический журнал, если технологический журнал включен (в файле `logcfg.xml` имеется хотя бы один элемент `log`) и произошла одна из следующих нештатных ситуаций:

- При работе системы «1С:Предприятие» произошло исключение операционной системы, процесс (клиент или сервер) завершен аварийно, построен дамп аварийного завершения.
- Возникло исключение в базе данных, приводящее к отображению сообщения об ошибке и закрытию приложения системы «1С:Предприятие».

При возникновении любой ошибки базы данных в технологический журнал записывается событие типа `EXCP`, если оно удовлетворяет условиям, записанным в конфигурационном файле технологического журнала (`logcfg.xml`).

3.18.3. Вывод информации о взаимных блокировках

При любом обращении к СУБД, но не чаще чем 1 раз в 2 секунды, выполняется дополнительное обращение к СУБД с запросом, какой поток был заблокирован и каким потоком. Результатом такого запроса является таблица пар («жертва блокировки», «источник блокировки»), где:

- **Жертва блокировки** – идентификатор соединения с СУБД, которое ожидает блокировки.
- **Источник блокировки** – идентификатор соединения с СУБД, которое установило блокировку.

Если в кластере несколько рабочих процессов, то запрос выполняется одним из них. Запросы о взаимных блокировках нумеруются.

Данные из полученной таблицы добавляются к контексту каждого потока, которому соответствуют полученные идентификаторы соединений с СУБД, и будут отображены в качестве значения блокировочных свойств очередного события технологического журнала. После того как в потоке, к контексту которого добавлена информация о блокировках, будет завершено очередное событие технологического журнала, к этому событию будут добавлены блокировочные свойства. При этом если поток был жертвой блокировки, то события блокировки будут очищены после вывода. Если поток был источником, то очистка выполняется при закрытии или откате транзакции.

Информация о блокировках добавляется к потокам в следующем порядке:

- Если поток-жертва еще не знает об этом, то ему устанавливается номер запроса и идентификатор потока-источника блокировки.
- К потоку-источнику блокировки добавляется номер запроса, только если у него есть жертвы, которые об этом еще не знали.

Информация о блокировках:

- поток является источником, момент обнаружения;
- поток является жертвой, момент обнаружения;
- номер запроса (если поток является жертвой);
- список номеров запросов (если поток является источником);
- номер соединения источника (если поток является жертвой).

Блокировочные свойства событий:

- `lka='1'` – поток является источником блокировки.
- `lkp='1'` – поток является жертвой блокировки.
- `lkpid` – номер запроса к СУБД, «кто кого заблокировал» (только для потока-жертвы блокировки). Например, `'423'`.
- `lkaid` – список номеров запросов к СУБД, «кто кого заблокировал» (только для потока-источника блокировки). Например, `'271,273,274'`.
- `lksrc` – номер соединения источника блокировки, если поток является жертвой, например, `'23'`.
- `lkpto` – время в секундах, прошедшее с момента обнаружения, что поток является жертвой. Например: `'15'`.
- `lkato` – время в секундах, прошедшее с момента обнаружения, что поток является источником блокировок. Например, `'21'`.

Таким образом, для анализа блокировок необходимо найти в технологических журналах процессов `rphost` первое событие со свойствами `lka` и `lkp`, узнать значения свойств `lkaid`, `lkpid` и найти все события с этими значениями свойств в журналах всех рабочих процессов кластера. По найденной группе событий можно установить, кто кого заблокировал, на сколько времени и что они при этом делали.

Также в свойстве `Txt` события `TLOCK` в технологическом журнале может быть отображено пространство имен, в котором наложена блокировка.

3.19. logui.txt

Файл `logui.txt` содержит список интерактивных действий пользователя, которые выполнялись за время журналирования.

Расположение файла:

- Для ОС Windows: `%APPDATA%\1C\1cv8\<Уникальный идентификатор информационной базы>`.
- Для ОС Linux: `~/1cv8/1C/1cv8/<Уникальный идентификатор информационной базы>`.

Одна запись файла содержит описание одного действия пользователя. Общий формат строки следующий:

- дата и время события;
- описание события (`Event`);
- имя объекта, с которым произошло событие;
- время (в миллисекундах) от начала работы программы (`t`);
- префиксы `beg` или `end` (аналогично открывающей и закрывающей скобке), идентифицирующие начало и конец события;
- детализация (`Detail`).

С целью сбора статистики длительности выполнения действий в протоколе фиксируется начало и окончание действия. Начало действия содержится в записи помеченным как `beg`, окончание действия – в записи помеченным как `end` (данные признаки отображаются в самом конце строки журнала).

Также фиксируется длительность действия, помеченная как `t`. Отсчет осуществляется с момента запуска программы и ведется в миллисекундах.

Между строками лога, содержащими `beg` и `end`, могут находиться как вложенные действия, содержащие `beg` и `end`, так и строки, отражающие какие-либо действия, подлежащие фиксации записи в журнал.

Протоколирование ведется для всех элементов формы и элементов глобального командного интерфейса, доступных в режиме 1С:Предприятие.

Протоколируются следующие действия:

- Нажатие кнопки на клавиатуре. Данные, вводимые пользователем, заменяются в протоколе звездочкой (`Event Key_<клавиша>` или `Event key_*`).
- Нажатие левой (`Event_LClick`), правой (`Event_RClick`) и средней кнопкой мыши (`Event_MBtndn`).
- Двойной щелчок левой кнопкой мыши (`Event_LBtndbl`).
- Прокрутка колеса на мышке (`Event_Wheel`).

Для некоторых элементов используются специальные, присущие только им события:

- Форма:
 - активация окна формы:

"Event FormActivate", "Name <имяформы>"

- Панель подсистем (**PartitionPanel**):
 - выбор подсистемы или рабочего стола при помощи мыши:

[Копировать в буфер обмена](#)

"Event LClick", "Name PartitionPanel"

- активация сочетанием клавиш:

[Копировать в буфер обмена](#)

"Event PanelActivate", "Name SubsystemsPanel"

- Панель навигации (**FormNavigationPanel**):
 - исполнение команды:

[Копировать в буфер обмена](#)

"Event LClick", "Name FormNavigationPanel", "Detail Execute<имя команды>"

- разворачивание/сворачивание группы команд:

[Копировать в буфер обмена](#)

"Event LClick", "Name FormNavigationPanel", "Detail Close<имя группы команд>"

- активация сочетанием клавиш:

[Копировать в буфер обмена](#)

"Event PanelActivate", "Name FormNavigationPanel"

- Заголовок окна (**WindowCaptionText**):
 - нажатие на заголовок:

[Копировать в буфер обмена](#)

"Event LClick", "Type WindowCaptionText", "Detail Обработка"

- Панель действий (**ActionsPanel**):
 - исполнение команды:

[Копировать в буфер обмена](#)

"Event LClick", "Name ActionsPanel", "Detail <имя команды>"

- активация сочетанием клавиш:

[Копировать в буфер обмена](#)

"Event PanelActivate", "Name ActionsPanel"

- Область информационной панели, в которой отображается список последних оповещений (**NotificationHistoryPanel**):
 - исполнение команды:

[Копировать в буфер обмена](#)

"Event Key_SPACE", "Name NotificationHistoryPanel", "Detail<имя>"

- активация сочетанием клавиш:

[Копировать в буфер обмена](#)

"Event PanelActivate", "Name NotificationHistoryPanel"

- Окно состояния (**StatusWindow**):
 - закрытие:

[Копировать в буфер обмена](#)

"Event CloseWindow"

- перемещение:

[Копировать в буфер обмена](#)

"Event MoveWindow offset=<dx,dy> pos=<x,y,w,h>"

- Окно оповещения (**NotificationWindow**):
 - закрытие:

[Копировать в буфер обмена](#)

"Event CloseWindow"

- нажатие на ссылку:

[Копировать в буфер обмена](#)

"Event LClick", "Name NotificationWindow", "Detail Hyperlink"

- перемещение:

[Копировать в буфер обмена](#)

"Event MoveWindow offset=<dx,dy> pos=<x,y,w,h>"

- Окно проверки (подсказка об ошибках ввода) (**CheckWindow**):
 - нажатие мышкой кнопки [Следующее сообщение](#):

[Копировать в буфер обмена](#)

"Event LClick", "Name CheckWindow", "Detail NextButton"

- нажатие мышкой кнопки [Предыдущее сообщение](#):

[Копировать в буфер обмена](#)

"Event LClick", "Name CheckWindow", "Detail PrevButton"

- нажатие мышкой кнопки [Закреть](#):

[Копировать в буфер обмена](#)

"Event LClick", "Name CheckWindow", "Detail CloseButton"

Пример записей лога:

```
"17.12.200816:41:55", "Event Key_SPACE",
"Name HistoryPanel", "t=465562", "beg"
"17.12.200816:41:55", "Event FormActivate",
"Name Справочник.Товары.ФормаСписка", "t=465562"
"17.12.200816:41:56", "Event Key_SPACE", "Name HistoryPanel",
"Detail Жирность", "t=466281", "end"
"17.12.200816:07:05", "Event PanelActivate",
"Name HistoryPanel", "t=918188"
```

[Копировать в буфер обмена](#)

3.20. nethasp.ini

Для настройки параметров взаимодействия системы «1С:Предприятие» с HASP License Manager используется конфигурационный файл [nethasp.ini](#).

Файл располагается в каталоге конфигурационных файлов системы «1С:Предприятие», и его наличие не является обязательным. Если данный файл отсутствует в каталогах конфигурационных файлов, то выполняется поиск этого файла в следующих каталогах:

- Для ОС Windows:
 - Каталог исполняемых файлов запущенной версии «1С:Предприятия»;
 - Текущий каталог;
 - Каталог `%SYSTEMROOT%\Windows\System32` (для 32-разрядной ОС Windows) или `%SYSTEMROOT%\Windows\SysWOW64` (для 64-разрядной ОС Windows).
 - Каталог `%SYSTEMROOT%\Windows\System`;
 - Каталог `%SYSTEMROOT%\Windows`;
 - Каталоги, перечисленные в переменной окружения `PATH`.
- Для ОС Linux:
 - Текущий каталог;
 - Домашний каталог пользователя;
 - Каталог `/etc`.

Файл [nethasp.ini](#) содержит четыре секции:

- [\[NH_COMMON\]](#) – для общих настроек (см. [здесь](#));
- [\[NH_IPX\]](#) – для протокола IPX (см. [здесь](#));
- [\[NH_NETBIOS\]](#) – для протокола NetBIOS (см. [здесь](#));
- [\[NH_TCPIP\]](#) – для протокола TCP/IP (см. [здесь](#)).

В секции [\[NH_COMMON\]](#) содержатся глобальные настройки для всех разделов файла конфигурации. Во всех остальных разделах содержатся настройки, влияющие на выполнение операций с конкретным протоколом.

В каждой секции можно использовать параметры, специфические для данной секции или общие для всех секций. Указание общего для всех секций параметра в секции для одного из трех протоколов обладает большим приоритетом, чем настройка в секции [\[NH_COMMON\]](#) (по отношению к этому протоколу).

Для определения дополнительных настроек конкретного протокола следует использовать параметры, специфические для конкретной секции.

В файле конфигурации могут присутствовать комментарии. Комментарий начинается с символа «;» (точка с запятой) и продолжается до конца строки.

Регистр букв в именах параметров не имеет значения.

Ниже приводится список параметров и их допустимые значения, которые могут приводиться в тех или иных секциях файла [nethasp.ini](#).

При установке системы «1С:Предприятие» образец файла [nethasp.ini](#) копируется в каталог `conf` установочного каталога системы «1С:Предприятие». Данный файл практически полностью состоит из закомментированных строк и никак не переопределяет значений параметров по умолчанию, но при этом он содержит наиболее полный список параметров, которые могут использоваться для настройки взаимодействия системы «1С:Предприятие» с HASP License Manager.

Далее подробно описаны параметры каждой секции конфигурационного файла.

3.20.1. Секция [\[NH_COMMON\]](#)

[NH_IPX](#)

Возможные значения: `Enabled`, `Disabled`. Использовать или не использовать (соответственно) протокол IPX для связи с HASP License Manager.

Значение по умолчанию: `Enabled`.

[NH_NETBIOS](#)

Возможные значения: `Enabled`, `Disabled`. Использовать или не использовать (соответственно) протокол NetBIOS для связи с HASP License Manager.

Значение по умолчанию: `Enabled`.

[NH_TCPIP](#)

Возможные значения: `Enabled`, `Disabled`. Использовать или не использовать (соответственно) протокол TCP/IP для связи с HASP License Manager.

Значение по умолчанию: `Enabled`.

[NH_SESSION](#)

Возможные значения: `<Число>`. Задаёт интервал в секундах, в течение которого программа пытается установить соединение с HASP License Manager.

Значение по умолчанию: 2 секунды.

[NH_SEND_RCV](#)

Возможные значения: `<Число>`. Задаёт интервал в секундах, в течение которого программа пытается установить соединение с HASP License Manager.

Значение по умолчанию: 1 секунда.

3.20.2. Секция [NH_IPX]*NH_USE_SAP*

Возможные значения: **Enabled**, **Disabled**. Использовать или не использовать службу **SAP** для поиска в сети **HASP License Manager**.

Значение по умолчанию: **Enabled**.

NH_USE_BROADCAST

Возможные значения: **Enabled**, **Disabled**. Использовать только механизм **Broadcast** для поиска в сети **HASP License Manager**. Данную возможность имеет смысл использовать при работе с протоколом **IPX** в сетях, отличных от **Novell NetWare**. Значение по умолчанию: **Enabled**.

NH_BC_SOCKET_NUM

Возможные значения: **<Число>**. Определяет номер сокета для широковещательного механизма. Число указывается в шестнадцатеричном виде.

Значение по умолчанию: **7483H**.

NH_SERVER_NAME

Возможные значения: **localnet**, **Internet**. Определяет, будет ли приложение обмениваться данными только с **HASP LM**, находящимся в локальной сети, или с любыми другими **HASP LM**.

Значение по умолчанию: **Internet**.

NH_DATFILE_PATH

Возможные значения: **<Путь>**. Путь, по которому будет производиться поиск файлов **haspaddr.dat** и **newhaddr.dat**, содержащих сетевой адрес **HASP License Manager**. Данный параметр в основном имеет смысл использовать только при установках **NH_USE_SAP=Disabled** и **NH_USE_BROADCAST=Disabled**, так как в противном случае адрес **HASP License Manager** может быть определен автоматически.

NH_SESSION

Возможные значения: **<Число>**. Задаёт интервал в секундах, в течение которого программа пытается установить соединение с **HASP License Manager**.

Значение по умолчанию: **2** секунды.

NH_SEND_RCV

Возможные значения: **<Число>**. Устанавливает для **HASP License Manager** максимальное время получения или отправки пакета.

Значение по умолчанию: **1** секунда.

3.20.3. Секция [NH_NETBIOS]*NH_NBNAME*

Возможные значения: **<Имя>**. Задаёт имя **HASP License Manager** (длина имени – до 8 символов).

NH_USELANANUM

Возможные значения: **<Число>**. Устанавливает номер коммуникационного канала, который будет использоваться в качестве коммуникационного канала.

NH_SESSION

Возможные значения: **<Число>**. Задаёт интервал в секундах, в течение которого программа пытается установить соединение с **HASP License Manager**.

Значение по умолчанию: **2** секунды.

NH_SEND_RCV

Возможные значения: **<Число>**. Устанавливает для **HASP License Manager** максимальное время получения или отправки пакета.

Значение по умолчанию: **1** секунда.

3.20.4. Секция [NH_TCPIP]*NH_SERVER_ADDR*

Возможные значения: **<Адрес1>**, **<Адрес2>**. Устанавливает IP-адреса всех **HASP License Manager**. Возможно использование неограниченных адресов и множественных строк.

IP-адрес: **192.114.176.65**.

Имя локального узла: **ftp.aladdin.co.il**.

NH_SERVER_NAME

Возможные значения: **<Имя1>**, **<Имя2>**. Обменивается данными с **HASP LM** с определенным именем. Максимально – 6 имен; каждое имя может состоять максимум из 7 символов.

NH_PORT_NUMBER

Возможные значения: **<Число>**. Устанавливает номер сетевого порта.

Значение по умолчанию: **475**.

NH_TCPIP_METHOD

Возможные значения: **TCP**, **UDP**. Посылает пакет **TCP** или **UDP**.

Значение по умолчанию: **UDP**.

ПРИМЕЧАНИЕ. Установка параметра в значение **TCP** игнорируется. Обращение к **HASP License Manager** всегда выполняется по **UDP**.

NH_USE_BROADCAST

Возможные значения: **Enabled**, **Disabled**. Использовать широковещательный механизм **UDP**.

Значение по умолчанию: **Enabled**.

NH_SESSION

Возможные значения: <Число>. Задаёт интервал в секундах, в течение которого программа пытается установить соединение с HASP License Manager.

Значение по умолчанию: 2 секунды.

NH_SEND_RCV

Возможные значения: <Число>. Устанавливает для HASP License Manager максимальное время получения или отправки пакета.

Значение по умолчанию: 1 секунда.

3.21. nhsrv.ini

Некоторые настройки HASP License Manager могут задаваться при помощи файла конфигурации [nhsrv.ini](#).

Расположение файла:

- Для ОС Windows: поиск данного файла выполняется в различных каталогах в следующей последовательности:
 - каталог, в котором размещается исполняемый файл HASP License Manager;
 - текущий каталог Windows;
 - системный каталог Microsoft Windows (%SystemRoot%\system32 – для 32-разрядной версии и %SystemRoot%\system – для 64-разрядной версии);
 - каталог Microsoft Windows (каталог %SystemRoot%);
 - каталоги, перечисленные в переменной окружения PATH (только в случае установки HASP License Manager как приложения Microsoft Windows).

В ОС Windows рекомендуется размещать файл [nhsrv.ini](#), если это необходимо, в каталоге, в котором размещается исполняемый файл HASP License Manager. Проверка того, что HASP License Manager нашел и прочитал файл конфигурации, возможна с помощью журнала [Activity Log/Server Activity Log](#).

- Для ОС Linux: размещение конфигурационного файла [nhsrv.ini](#) следует указать с помощью параметра `-c`. Расположение файла конфигурации по умолчанию не определено.

Настройка HASP License Manager осуществляется при помощи установки тех или иных значений параметров в секции `[NHS_SERVER]` файла [nhsrv.ini](#):

NHS_IP_LIMIT

Возможные значения: <ipAddr>, <ipAddr>,...

Определяет диапазон сетевых станций, обслуживаемых HASP LM. Например: `10.1.1.1`, `10.1.1.*`, `10.1.1.1/32`, `10.1.1.1/24`.

NHS_ADAPTER

Возможные значения: <ipAddrSubMask>, <ipAddrSubMask>,...

Определяет IP-адрес одной или более сетевых карт, которые будут обслуживать HASP License Manager. Применяется при использовании HASP License Manager с Win32. Например: `10.1.1.111`, `255.255.0.0`.

NHS_USERLIST

Максимальное количество пользователей, одновременно подключенных к HASP License Manager. Значение по умолчанию: 250.

3.22. srv1cv83

Конфигурационный файл `/etc/sysconfig/srv1cv83` (для RPM-системы) используется для задания параметров запуска агента сервера «1С:Предприятия» с помощью скрипта `/etc/init.d/srv1cv83`. Если установка выполнялась для DEB-системы, то нижеперечисленные параметры следует редактировать в файле `/etc/init.d/srv1cv83`.

ВНИМАНИЕ! Данный конфигурационный файл используется только в случае запуска сервера «1С:Предприятия» в операционной системе Linux.

С помощью этого конфигурационного файла можно настраивать следующие параметры:

SRV1CV8_KEYTAB

Путь к файлу закрытого ключа Kerberos.

SRV1CV8_DATA

Каталог, в котором будут расположены служебные файлы кластера серверов (в том числе список кластеров и список информационных баз кластера).

SRV1CV8_PORT

Номер главного порта агента кластера. Этот порт используется агентом кластера для обращения к центральному серверу. Порт агента кластера также указывается в качестве сетевого порта рабочего сервера.

SRV1CV8_REGPORT

Номер сетевого порта кластера, создаваемого по умолчанию при первом запуске [ragent](#).

SRV1CV8_RANGE

Диапазоны сетевых портов для динамического выбора. Из них выбираются служебные порты процессов кластера при невозможности их выбора из настроек соответствующего рабочего сервера.

SRV1CV8_DEBUG

Запуск в режиме отладки:

- `0` – без режима отладки (по умолчанию);
- `1` – в режиме отладки.

SRV1CV8_SECLEV

Уровень безопасности соединений;

- `0` – выключено (по умолчанию);
- `1` – установка соединения;

3.23. swpuser.ini

Файл `swpuser.ini` предназначен для переопределения пользователей, от имени которых будут выполняться рабочие процессы и менеджер кластера. По умолчанию рабочий процесс и менеджер кластера выполняются от имени того же пользователя, что и агент сервера.

Файл `swpuser.ini` располагается в каталоге реестра кластера (подробнее об устройстве кластера серверов см. книгу «1С:Предприятие 8.3. Клиент-серверный вариант. Руководство администратора»), с которым работает главный менеджер кластера, и имеет следующий формат:

[Копировать в буфер обмена](#)

```
user=<имя пользователя для rphost>
password=<пароль пользователя для rphost>
[rnmngr_user=<имя пользователя для rnmngr>
[rnmngr_pass=<пароль пользователя для rnmngr>]]
[registry=<каталог реестра кластера>]
[<port>:
[user=<имя пользователя для rphost>
[password=<пароль пользователя для rphost>]]
[rnmngr_user=<имя пользователя для rnmngr>
[rnmngr_pass=<пароль пользователя для rnmngr>]]
[registry=<каталог реестра кластера>]]
```

С помощью данного файла можно указать:

- Пользователя (и его пароль) от имени которого будут выполняться рабочие процессы во всех кластерах на данном компьютере (параметры `user` и `password`). Если не указано, то рабочий сервер работает от имени того же пользователя, что и агент сервера.
- Пользователя (и его пароль) от имени которого будут выполняться менеджеры кластера во всех кластерах данного компьютера (параметры `rnmngr_user` и `rnmngr_pass`). Если не указано, то менеджер кластера работает от имени того же пользователя, что и агент сервера.
- Каталог реестра кластера для всех рабочих серверов в кластере (параметр `registry`). Если не указан, то будет использоваться каталог вида `reg_<порт>` в каталоге со списком кластеров главного менеджера кластера (указывается в параметре `-d` при запуске агента кластера).

При необходимости смены каталога с реестром кластера следует помнить, что пользователь, от имени которого работает менеджер кластера, должен иметь полные права на этот каталог.

- Если необходимо для каждого кластера указать свой набор параметров, то для этого в файле `swpuser.ini` создается раздел с номером порта менеджера кластера и в этом разделе можно указать все вышеперечисленные параметры.

Если для кластера указывается индивидуальный реестр кластера, то пользователь, от имени которого работает менеджер этого кластера, должен иметь полные права в указанном каталоге.

ВНИМАНИЕ! Пароли в примере указаны исключительно в демонстрационных целях. Настоятельно не рекомендуется указывать такие (или аналогичные) пароли в системах, находящихся в промышленной эксплуатации.

Рассмотрим пример использования файла `swpuser.ini`. В рамках этого примера на компьютере работают два кластера (с портами `1541` и `1641`), для главного менеджера кластера указан параметр `-d"d:\cluster\main"`. В каталоге `d:\cluster\main` размещен файл `swpuser.ini` следующего содержания:

[Копировать в буфер обмена](#)

```
user=svr_1c_rphost
password=123
rnmngr_user=svr_1c_agent
rnmngr_pass=123
registry=d:\cluster\common
1541:
user=svr_1c_rphost_1541
password=123
rnmngr_user=svr_1c_agent_1541
rnmngr_pass=123
registry=d:\cluster\one
```

В результате:

- Для кластера с портом менеджера кластера `1541` будет использоваться каталог реестра кластера `d:\cluster\1541` и пользователи `svr_1c_rphost_1541` и `svr_1c_agent_1541` для работы рабочего процесса и менеджера кластера соответственно. Пользователь `svr_1c_agent_1541` должен иметь полный доступ в каталог `d:\cluster\one`, фактический каталог с данными реестра кластера будет `d:\cluster\one\reg_1541`.
- Кластер с портом менеджера кластера `1641` (и все кластеры, которые могут быть добавлены в будущем, если не будет изменен приведенный файл `swpuser.ini`) будут использовать для хранения своих реестров кластера подкаталоги каталога `d:\cluster\common` и пользователей `svr_1c_rphost` и `svr_1c_agent` для работы рабочих процессов и менеджеров кластера (соответственно). Фактическими каталогами с реестрами кластеров будут каталоги `d:\cluster\common\reg_1641` и т. д.

При использовании файла `swpuser.ini` пользователь, от имени которого работает главный агент кластера (`ragent`) должен иметь **административные права**. В силу этого, настоятельно рекомендуется явным образом указывать пользователей, от имени которых будут работать рабочие процессы и менеджеры кластера. В противном случае весь кластер будет работать от имени администратора, что крайне небезопасно.

Пользователи, от чьего имени будут работать рабочие процессы и менеджеры кластера, должны обладать следующими привилегиями:

- **Вход в систему как сервис (Log on as a service),**
- **Вход в систему как пакетное задание (Log on as a batch job).**

А также следует включить этих пользователей в группу **Пользователи журналов производительности (Performance Log Users)**.

3.24. testcfg.xml

Файл `testcfg.xml` предназначен для настройки диапазона портов, используемого при автоматизированном тестировании прикладных решений, работающих в веб-клиенте.

Файл располагается в каталоге конфигурационных файлов системы «1С:Предприятие», которая выступает в роли, и его наличие не является обязательным.

Если файл не найден, то для взаимодействия используются порты из стандартного диапазона (`1538-1539`).

Пример:

[Копировать в буфер обмена](#)

```
<config xmlns="http://v8.1c.ru/v8/testcfg">
  <testports range="1538:1539"/>
</config>
```

Элемент `testports` имеет атрибуты, которые описаны ниже.

Атрибут range

Тип: **Строка**. Содержит диапазон портов, используемых веб-сервером для организации взаимодействия между менеджером тестирования и клиентом тестирования.

Приложение 4. Утилита тестирования chdbfl

Утилита предназначена для автономной проверки и исправления файлов базы данных.

ВНИМАНИЕ! Перед использованием данной утилиты следует обязательно сделать резервную копию файла базы данных.

Данная утилита предназначена для работы только с файловой базой данных. Файловая база данных используется:

- для хранения информационной базы в файловом варианте;
- хранения хранилища конфигурации.

Для запуска утилиты в каталоге установки системы «1С:Предприятие» нужно запустить приложение **chdbfl**. На экран выводится окно (см. [рис. 85](#)).

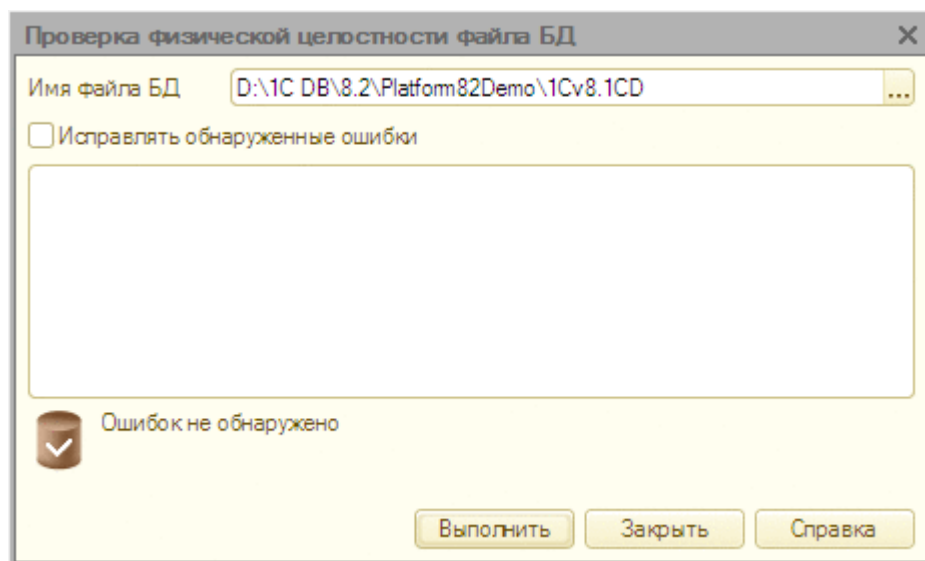


Рис. 85. Утилита тестирования и исправления информационных баз

В поле **Имя файла БД** указать или выбрать имя файла информационной базы.

Установить флажок **Исправлять обнаруженные ошибки**, если требуется исправлять обнаруженные при проверке ошибки.

Для начала работы утилиты нужно нажать кнопку **Выполнить**. К этому моменту выбранная информационная база не должна быть открыта конфигуратором или в режиме 1С:Предприятие.

Сообщения о найденных ошибках выводятся в текстовое поле.

Ниже текстового поля выводятся сообщения о результатах работы утилиты.

Также эта утилита может использоваться и для проверки хранилища конфигурации.

Рекомендуется использовать следующую последовательность шагов при использовании утилиты:

- Создать резервную копию файла базы данных.
- Выполнить тестирование без установленного флажка **Исправлять обнаруженные ошибки**.
- Если при тестировании не выявлено проблем, то можно выполнить тестирование с установленным флажком **Исправлять обнаруженные ошибки** – при этом будет выполнена операция реиндексации информационной базы или информационной базы хранилища. Такую операцию рекомендуется выполнять регулярно.

• Если при тестировании обнаружены проблемы, то попытаться выполнить исправление информационной базы. Если в процессе работы утилита сообщила о том, что какие-либо данные потеряны, то работать с такой информационной базой не рекомендуется. Из получившейся информационной базы можно:

- получить сохранившиеся данные и конфигурацию для создания на их основе новой информационной базы;
- получить последнюю версию конфигурации, на основании которой создать другое хранилище.

Приложение 5. Обработка ошибок

При формировании текста сообщения об ошибке используется следующий алгоритм:

- Из цепочки возникших исключений извлекается самое вложенное исключение (т. е. самое первое исключение).
- Текст описания самого первого исключения используется для краткого представления ошибки:
 - При этом для ошибок компиляции модулей в краткое представление включен текст, связанный с местоположением возникновения ошибки (строка в модуле).
 - Для краткого представления ошибок времени выполнения используется текст без подробностей возникновения ошибки (строки в модуле).
- Краткое представление ошибки отображается в диалоге, отображаемом пользователю. В этом диалоге возможно наличие кнопки [Подробно...](#), если выполняются следующие условия:
 - включен режим отладки,
 - ошибка встроенного языка,
 - наличие более одного исключения в цепочке.
- Подробное представление ошибки формируется из описаний всех исключений цепочки.

Если диалог с ошибкой не содержит кнопку [Подробно...](#) и ошибка является критической (работа программы не может быть продолжена), то в этом диалоге будет находиться гиперссылка [Показать информацию для технической поддержки](#).

При нажатии кнопки [Подробно...](#) будет открыто дополнительное окно, в котором будет содержаться подробное представление ошибки и дополнительная кнопка [Конфигуратор...](#), если в диалоге отображается ошибка встроенного языка.

Во время работы тонкого клиента или веб-клиента в клиент-серверном варианте работы обрабатываются следующие ошибочные ситуации:

- Если при начале работы системы обнаружена установленная блокировка соединений с информационной базой, то отображается сообщение об ошибке с возможностью повтора соединения.
- В случае несовпадения версии сервера и клиентского приложения отображается сообщение об ошибке с возможностью выполнить перезапуск.
- При отсутствии прав на запуск тонкого клиента поведение зависит от использования ключа запуска `-AppAutoCheckMode`:
 - Если ключ был указан при запуске, производится автоматическая попытка запуска толстого клиента.
 - Если ключ не был указан при запуске, отображается окно с сообщением об ошибке (без возможности перезапуска).
- При невозможности соединения с сервером «1С:Предприятия» (или веб-сервером) для запросов, допускающих повторную отправку, и для запросов, выполняемых до начала сеанса, отображается сообщение об ошибке с возможностью повторить запрос или отказаться от попытки. При отказе от повторения запроса «1С:Предприятие» не завершает свою работу. Если же запрос не допускает повторной отправки, отображается окно с сообщением об ошибке и возможность выполнить перезапуск.
- Если ошибка произошла в процессе обработки запроса на сервере, отображается окно с сообщением об ошибке и возможность выполнить перезапуск.

- При ошибке работы с сеансом (например, сеанс был удален администратором) отображается окно с сообщением об ошибке, и предоставляется возможность выполнить перезапуск.

При возникновении внутренней ошибки платформы веб-клиент формирует ошибку вида: **Неизвестная ошибка: <описание ошибки>**.

Если для подключения к информационной базе используется веб-клиент или тонкий клиент, подключенный через веб-сервер, то при возникновении нестандартных ситуаций используются следующие коды ошибок:

- **400 Bad Request** – описывает ошибки прикладного решения, которые не имеют критических последствий для работы клиентского приложения (включая все исключения времени выполнения).
Нельзя перехватывать для отображения пользователю.
- **500 Internal Server Error** – описывает ошибки прикладного решения, имеющие критические последствия для клиентского приложения, например:
 - исключения базы данных, являющиеся невозможными;
 - исключения, связанные с удалением сеанса или отсутствием данных сеанса.
- **502 Bad gateway** – описывает ошибки кластера серверов или регламентных операций, имеющие критические последствия для приложения, например, если невозможно установить соединение с сервером «1С:Предприятия».
- **503 Service Unavailable** – описывает ошибки кластера серверов или регламентных операций, имеющие критические последствия для клиентского приложения, например, невозможно установить соединение с сервером СУБД.

ВНИМАНИЕ! Для сохранения работоспособности системы, не следует перехватывать ошибки с кодом **400**. Ошибки с кодами **50x** можно перехватывать для отображения более дружелюбного сообщения об ошибке.

Приложение 6. Интернет-сервисы получения списка общих информационных баз и дистрибутива клиентского приложения

6.1. Получение списка общих информационных баз

При удаленной работе (например, через веб-сервер) возникает необходимость получения списка общих информационных баз. В этом случае параметр `CommonInfoBases` конфигурационного файла `1cestart.cfg` не позволяет получить этот список. Для получения этого списка можно воспользоваться возможностью публикации списка общих информационных баз через Интернет-сервис. При этом доступно получение списка как через HTTP-запросы и через Web-сервисы.

Если получение списка общих информационных баз выполняется с помощью https-соединения, то проверка сертификата сервера (с которого выполняется получение списка) выполняется с помощью сертификатов корневых удостоверяющих центров, получаемых из файла `cacert.pem` (который находится в каталоге исполняемых файлов платформы) и из хранилища сертификатов ОС Windows.

6.1.1. Получение через Web-сервис

Для получения списка общих информационных баз через Web-сервис необходимо опубликовать специальный Web-сервис, который будет возвращать этот список. Рассмотрим работу этого Web-сервиса более подробно.

6.1.1.1. Описание функционирования

Интерактивная программа запуска (`1cv8s`) может получить список общих баз, как из локальной сети, так и через Интернет. Получение списка баз через Интернет выполняется только в случае интерактивного запуска и если указан адрес для получения списка общих информационных баз (параметр `InternetService` или `WebCommonInfoBases` файла `1cestart.cfg`).

Для нормального функционирования, механизм получения списка общих информационных баз должен удовлетворять следующим требованиям:

- Вызов метода `WebCommonInfoBases.CheckInfoBases()` должен быть анонимным;
- Вызов метода `WebCommonInfoBases.GetInfoBases()` должен выполняться с аутентификацией;
- Информационная база, которая возвращает списки общих баз, должна содержать список пользователей, которые могут обращаться к ней за списком баз.

Первым шагом осуществляется вызов метода `WebCommonInfoBases.CheckInfoBases()` (анонимно). Если для этого компьютера и для этого пользователя интерактивная программа запуска вызывается впервые, то в качестве параметров `ClientID` и `InfoBaseCheckCode` передается значение `00000000-0000-0000-0000-000000000000`. Если вызов не первый, то в качестве параметров передаются код клиента и код, идентифицирующий текущий список общих информационных баз. Метод Web-сервиса должен определить, нужно обновлять список общих информационных баз для данного клиента или нет. Если такая необходимость есть – то выходной параметр `InfoBasesChanged` должен получить значение *Истина*, а параметр `URL` должен содержать адрес Web-сервиса, в котором реализован метод `WebCommonInfoBases.GetInfoBases()` (требует аутентификации). В противном случае параметр `InfoBasesChanged` должен получить значение *Ложь*, а параметр `URL` должен содержать пустую строку.

Алгоритм, используемый для проверки неизменности списка общих информационных баз, не регламентируется и может быть произвольным. Необходимо понимать, что интерактивная программа запуска не занимается расчетом значения кода, идентифицирующего список общих информационных баз, а просто хранит значение, которое было передано при предыдущем вызове Web-сервиса.

Если вызов метода `WebCommonInfoBases.CheckInfoBases()` указал на необходимость обновления списка, интерактивная программа запуска вызывает метод Web-сервиса `WebCommonInfoBases.GetInfoBases()`. Web-сервис расположен по адресу, который вернула функция `WebCommonInfoBases.CheckInfoBases()` в параметре `URL`. Метод `GetInfoBases()` должен сопоставить пользователя, от имени которого выполнена

аутентификация Web-сервиса с каким-либо кодом клиента. Сопоставление может быть «персональным» – пользователь идентифицирует себя своим персональным именем пользователя и паролем и получает свой, персональный список общих информационных баз. Также сопоставление может быть «ролевым» – пользователь идентифицирует свою принадлежность к некоторой роли, например [Оператор](#), [Кладовщик](#) и т. д., и получает список общих информационных баз, который общий для всех пользователей, выполняющих одну роль. При этом необходимо понимать, что в первом случае информационная база, реализующая метод `GetInfoBases()`, должна содержать список всех пользователей, которые могут запускать интерактивную программу запуска ([1cv8s](#)), подключенную в Web-сервису. Во втором случае в списке пользователей достаточно завести имена ролей.

Метод `GetInfoBases()` должен вернуть три значения:

- Код клиента (если он не задан);
- Список общих информационных баз в формате [v8i](#) (см. [здесь](#));
- Значение кода, идентифицирующий переданный список общих информационных баз. Это значение будет передано в метод `WebCommonInfoBases.CheckInfoBases()` при следующей проверке необходимости обновления списка общих информационных баз.

Если список общих информационных баз получается впервые, то код клиента (параметр `ClientID`) будет равно значению `00000000-0000-0000-0000-000000000000`.

Также следует понимать следующие особенности:

- Информационная база, в которой реализуется Web-сервис `WebCommonInfoBases`, должна публиковаться в две разные публикации – это вытекает из необходимости обеспечить разный уровень аутентификации для вызова метода `CheckInfoBases()` и `GetInfoBases()`.
- Организация анонимного доступа выполняется с помощью явного указания пользователя, от имени которого выполняется доступ, в файл `default.vrd`.
- Пользователь, от имени которого организуется анонимный доступ, не должен иметь возможность вызывать метод получения списка информационных баз, а только указывать, изменился список для переданного значения `ClientID` или нет.
- Ни одна публикация, обслуживающая Web-сервис `WebCommonInfoBases`, не должна предоставлять возможность работать с помощью веб-клиента.

6.1.1.2. Описание Web-сервиса

Имя Web-сервиса: `WebCommonInfoBases`. Тайм-аут на выполнение любого метода Web-сервиса составляет 3 секунды.

Методы Web-сервиса перечислены ниже.

CheckInfoBases

Описание:

Данный метод используется интерактивной программой запуска ([1cv8s](#)) для определения необходимости получения списка общих информационных баз.

Параметры:

ClientID входной

Тип: [Строка](#). Содержит идентификатор клиента, для которого проверяется актуальность списка общих информационных баз.

InfoBaseCheckCode входной

Тип: **Строка**. Код, идентифицирующий список информационных баз. Код должен однозначно идентифицировать текущий список информационных баз. При любом изменении списка код должен быть изменен и не должен совпадать ни с одним, ранее используемым для данного идентификатора клиента, кодом.

InfoBasesChanged **выходной**

Тип: **Булево**. Признак того, что список общих информационных баз необходимо получить повторно.

URL **выходной**

Тип: **Строка**. URL, по которому следует запрашивать список общих информационных баз в том случае, если этот список поменялся с прошлого обращения.

Возвращаемое значение:

Тип произвольный, значение игнорируется.

GetInfoBases

Описание:

Параметры:

ClientID **входной/выходной**

Тип: **Строка**. Содержит идентификатор клиента, для которого выполняется получение списка общих информационных баз. Если идентификатор клиента не задан (равен значению 00000000-0000-0000-0000-000000000000), то метод должен назначить идентификатор клиента и вернуть его в этом параметре.

InfoBaseCheckCode **выходной**

Тип: **Строка**. Значение кода, идентифицирующего список общих информационных баз, который возвращается данным методом в параметре **InfoBases**.

InfoBases **выходной**

Тип: **Строка**. Список общих информационных баз в формате **v8i** (см. [здесь](#)).

Возвращаемое значение:

Тип произвольный, значение игнорируется.

6.1.1.3. Пример реализации

Рассмотрим пример Web-сервиса для получения списка общих информационных баз.

ПРИМЕЧАНИЕ. Пример, приведенный в данном разделе, не является законченным. Он предназначен для демонстрации работы механизма.

В качестве Web-сервиса используется простая конфигурация, которая содержит один справочник и один Web-сервис.

Справочник имеет следующие характеристики:

- Имя **СписокОбщихБаз**.
- Тип кода **Строка**, длина 36 символов.
- Реквизиты:

- Имя `КодСписка`, тип `УникальныйИдентификатор`.
- Имя `СписокИБ`, тип `Строка`, неограниченной длины.
- Остальные параметры заданы по умолчанию.

Данный справочник будет хранить список идентификаторов клиентов (стандартный реквизит `Код`), список общих информационных баз (реквизит `СписокИБ`) и текущая версия списка информационных баз (реквизит `КодСписка`), рассчитанной при последнем получении списка для данного клиента. Версия списка является уникальным идентификатором, который изменяется при каждом сохранении элемента справочника. Для этого в модуле объекта определен обработчик `ПередЗаписью`:

[Копировать в буфер обмена](#)

Процедура `ПередЗаписью(Отказ)`

```
КодСписка = Новый УникальныйИдентификатор;
```

КонецПроцедуры

Также в конфигурации должен быть создан Web-сервис `WebCommonInfoBases`, для которого должны быть определены операции:

- `CheckInfoBases`, свойство `Тип возвращаемого значения` установлено в значение `string`, флажок `Возможно пустое значение` установлен. Остальные свойства установлены в значения по умолчанию. Параметры метода и их типы см. [здесь](#).
- `GetInfoBases`, свойство `Тип возвращаемого значения` установлено в значение `string`, флажок `Возможно пустое значение` установлен. Остальные свойства установлены в значения по умолчанию. Параметры метода и их типы см. [здесь](#).

Текст операций Web-сервиса:

[Копировать в буфер обмена](#)

Функция `CheckInfoBases(ClientID, InfoBaseCheckCode, InfoBaseChanged, URL)`

```
Если ClientID = "00000000-0000-0000-0000-000000000000"
```

```
И InfoBaseCheckCode = "00000000-0000-0000-0000-000000000000" Тогда
```

```
// это первое обращение клиента
```

```
InfoBaseChanged = Истина;
```

```
URL = "/listservice2/ws/WebCommonInfoBases";
```

```
Возврат "";
```

```
КонецЕсли;
```

```
Клиент = Справочники.СписокОбщихБаз.НайтиПоКоду(ClientID);
```

```
Если Клиент.Пустая() Тогда
```

```
// нет такого клиента
```

```
InfoBaseChanged = Ложь;
```

```
Иначе
```

```
// проверим, что список на стороне клиента и наш список не различаются
```

```
Если InfoBaseCheckCode = Клиент.КодСписка Тогда
```

```
// список не поменялся
```

```
InfoBaseChanged = Ложь;
```

```
URL = "";
```

```
Иначе
```

```
// список изменился
```

```
InfoBaseChanged = Истина;
```

```
URL = "/listservice2/ws/WebCommonInfoBases";
```

```
КонецЕсли;
```

```
КонецЕсли;
```

```
Возврат "";
```

```
КонецФункции
```

Функция `GetInfoBases(ClientID, InfoBaseCheckCode, InfoBases)`

```
Если ClientID = "00000000-0000-0000-0000-000000000000" Тогда
```

```
ТекПользователь = ПользователиИнформационнойБазы.ТекущийПользователь();
```

```
// необходимо завести нового клиента
```

```
// кодом элемента справочника будет уникальный идентификатор
```

```
// пользователя информационной базы
```

```
Объект = Справочники.СписокОбщихБаз.СоздатьЭлемент();
```

```
Объект.Код = Строка(ТекПользователь.УникальныйИдентификатор);
```

```
// именем клиента будет имя пользователя
```

```
Объект.Наименование = ТекПользователь.Имя;
```



```

// список ИБ пустой при первом обращении
Объект.СписокИБ = "";
Объект.Записать();
// сформируем возвращаемые значения Web-сервиса
InfoBaseCheckCode = Объект.КодСписка;
InfoBases = Объект.СписокИБ;
ClientID = Объект.Код;
Иначе
// здесь получим данные для существующего кода клиента
Клиент = Справочники.СписокОбщихБаз.НайтиПоКоду(ClientID);
Если Клиент.Пустая() Тогда
// нет такого клиента
InfoBaseCheckCode = "";
InfoBases = "";
Иначе
InfoBaseCheckCode = Клиент.КодСписка;
InfoBases = Клиент.СписокИБ;
КонецЕсли;
КонецЕсли;
Возврат "";
КонецФункции

```

После создания конфигурации следует дважды опубликовать Web-сервис на веб-сервере (см. [здесь](#)). Затем необходимо запомнить адреса опубликованных Web-сервисов. Предположим, что Web-сервисы опубликованы по адресам:

- <http://localhost/listservice> – анонимный Web-сервис;
- <http://localhost/listservice2> – Web-сервис, требующий аутентификации;

В информационной базе должны существовать пользователи **Анонимный**, и, например, пользователи с именами **Оператор**, **Кладовщик**, **Бухгалтер**.

Файл **default.vrd**, который описывает публикацию по адресу <http://localhost/listservice> имеет следующий вид:

[Копировать в буфер обмена](#)

```

<?xml version="1.0" encoding="UTF-8"?>
<point xmlns="http://v8.1c.ru/8.2/virtual-resource-system"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  base="/listservice"
  ib="File=X:\DB\ListBase;Usr=Анонимный"
  enable="false">
  <ws>
    <point name="WebCommonInfoBases"
      enable="true"/>
  </ws>
</point>

```

Файл **default.vrd**, который описывает публикацию по адресу <http://localhost/listservice2> имеет следующий вид:

[Копировать в буфер обмена](#)

```

<?xml version="1.0" encoding="UTF-8"?>
<point xmlns="http://v8.1c.ru/8.2/virtual-resource-system"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  base="/listservice2"
  ib="File= X:\DB\ListBase;"
  enable="false">
  <ws>
    <point name="WebCommonInfoBases"
      enable="true"/>
  </ws>
</point>

```

В настройках веб-сервера, для которого выполнены публикации Web-сервиса получения списка общих

информационных баз, должна быть запрещена обработка HTTP-запроса **HEAD** (по крайней мере для виртуальных каталогов, через которые выполняется обращение к Web-сервису). В противном случае Web-сервис не будет использоваться.

В форме настройки окна запуска (см. [здесь](#)) необходимо добавить Интернет-сервис с указанным выше адресом, с указанием суффикса **ws**: <http://localhost/listservice/ws/>.

После выполнения настройки следует запустить программу запуска. На предложение ввести имя пользователя и пароль доступа к Web-сервису «1С:Предприятия», следует вводить имена **Оператор**, **Кладовщик**, **Бухгалтер**. В справочнике **СписокОбщихБаз** появятся соответствующие записи. Если в реквизит **СписокИБ** каждого элемента справочника поместить свой список в формате **v8i**, то этот список будет добавляться в список информационных баз программы запуска, после выполнения аутентификации.

6.2. Получение дистрибутива клиентского приложения

При удаленной работе (через веб-сервер) возникает необходимость автоматического получения дистрибутива клиентского приложения в тех случаях, когда на сервере «1С:Предприятия» (или веб-сервере) выполнена замена версии системы. В этом случае поиск новой версии с помощью параметра **DistributiveLocation** конфигурационных файлов может не дать результат. Для получения дистрибутива можно воспользоваться возможностью публикации дистрибутива клиентского приложения через Интернет-сервис. При этом доступно получение списка как через HTTP-запросы и через Web-сервисы.

Если получение дистрибутива клиентского приложения выполняется с помощью **https**-соединения, то проверка сертификата сервера (с которого выполняется получение дистрибутива) выполняется с помощью сертификатов корневых удостоверяющих центров, получаемых из файла **cacert.pem** (который находится в каталоге исполняемых файлов платформы) и из хранилища сертификатов ОС Windows.

6.2.1. Получение через Web-сервис

Для получения дистрибутива клиентского приложения через Web-сервис необходимо опубликовать специальный Web-сервис, который будет возвращать этот дистрибутив. Рассмотрим работу этого Web-сервиса более подробно.

6.2.1.1. Описание функционирования

Если тонкий клиент (**1cv8c**) запущен с ключом **/AppAutoCheckVersion**, то выполняется попытка подобрать версию тонкого клиента в том случае, если она не совпадает с версией сервера «1С:Предприятия» или расширения веб-сервера. Для этого используются три механизма (по порядку использования):

- Поиск дистрибутива в локальной сети – с помощью параметров конфигурационных файлов (**1cestart.cfg** и **1cescmn.cfg**) **DistributiveLocation**.
- Получение дистрибутива клиентского приложения по **URL**, указанных в конфигурационных файлах **default.vrd** (атрибут элемента **point**, см. [здесь](#)) или **conf.cfg** (параметр **PublishDistributiveLocation**, см. [здесь](#)). При этом приоритетным является значение, указанное в файле **default.vrd**.
- Получение файла через Web-сервис получения дистрибутива клиентского приложения. Для этого адрес Web-сервиса необходимо указать в конфигурационном файле **1cestart.cfg** (параметр **InternetService** или **WebDistributiveLocation**, подробнее см. [здесь](#)) или в диалоге настройке окна запуска, подробнее см. [здесь](#).

Тонкий клиент анализирует результат обращения к Web-сервису. Если Web-сервис возвращает **0** в параметре **Size**, то считается, что необходимого дистрибутива необходимого клиентского приложения нет, и формируется ошибка о несоответствии версий клиентского приложения и сервера. В противном случае пользователю выдается запрос на скачивание и установку клиентского приложения, с указанием размера получаемого дистрибутива. При положительном ответе выполняется скачивание и установка новой версии, после чего происходит перезапуск нужной версии клиентского приложения. При выполнении загрузки файла дистрибутива тайм-аут на выполнение операции составляет **600** секунд. При выполнении загрузки дистрибутива не поддерживается **redirect** на стороне веб-сервера.

6.2.1.2. Описание Web-сервиса

Имя Web-сервиса: `WebDistributiveLocation`. Тайм-аут на выполнение любого метода Web-сервиса составляет 3 секунды.

Методы Web-сервиса перечислены ниже.

GetDistributiveInfo

Описание:

Данный метод используется тонким клиентом (`1cv8c`) для получения дистрибутива клиентского приложения нужной версии в следующих случаях:

- Несовпадение версии клиентского приложения и сервера, в клиент-серверном варианте, при подключении через веб-сервер.
- Несовпадение версии клиентского приложения и расширения веб-сервера, в файловом варианте, при подключении через веб-сервер.

Параметры:

OS входной

Тип: **Строка**. Тип операционной системы, для которой необходимо получить дистрибутив клиентского приложения.

Возможные значения: [Windows](#), [Linux](#).

Arch входной

Тип: **Строка**. Архитектура операционной системы, для которой необходимо получить дистрибутив клиентского приложения.

Возможные значения: [x86](#), [x86_64](#).

Version входной

Тип: **Строка**. Номер версии клиентского приложения, дистрибутив которого необходимо получить.

Size выходной

Тип: **Число**. Размер дистрибутива клиентского приложения (в байтах). Если запрашиваемый дистрибутив отсутствует, необходимо вернуть значение 0.

URL выходной

Тип: **Строка**. URL для скачивания дистрибутива клиентского приложения. При формировании URL следует помнить, что файл с дистрибутивом должен быть доступен веб-серверу, и у пользователя, который будет получать дистрибутив, также должны быть права на скачивание этого файла.

Дистрибутив клиентского приложения представляет из себя zip-архив файлов дистрибутива, без иерархии.

Возвращаемое значение:

Тип произвольный, значение игнорируется.

6.2.1.3. Пример реализации

Рассмотрим пример Web-сервиса для получения дистрибутива клиентского приложения.

ПРИМЕЧАНИЕ. Пример, приведенный в данном разделе, не является законченным. Он предназначен для демонстрации работы механизма.

В качестве Web-сервиса используется простая конфигурация, которая реализует собственно Web-сервис и не содержит никаких других объектов конфигурации. Дистрибутивы клиентского приложения будут расположены в специальном каталоге, к которому должен иметь доступ веб-сервер. В конфигурации должен быть создан Web-сервис [WebDistributiveLocation](#), для которого должна быть определены операция [GetDistributiveInfo](#), свойство [Тип возвращаемого значения](#) установлено в значение `string`, флажок [Возможно пустое значение](#) установлен. Остальные свойства установлены в значения по умолчанию. Параметры метода и их типы см. [здесь](#).

Текст операций Web-сервиса:

[Копировать в буфер обмена](#)

```
Функция GetDistributiveInfo(OS, Arch, Version, Size, URL)
    КаталогДистрибутивов = "C:\inetpub\Distribs\";
    URLДистрибутивов = "http://host/site/distribs/";
    // сформируем имя файла с архивом
    ИмяФайла = "tc-" + НРег(OS) + "-" + Arch + "-" + Version + ".zip";
    Архив = Новый Файл(КаталогДистрибутивов + ИмяФайла);
    Если Архив.Существует() Тогда
        Size = Архив.Размер();
        URL = URLДистрибутивов + ИмяФайла;
    Иначе
        Size = 0;
        URL = "";
    КонецЕсли;
    Возврат "";
КонецФункции
```

Следует указать в переменных `КаталогДистрибутивов` и `URLДистрибутивов` правильные значения, соответствующие реальному имени каталога с дистрибутивами клиентского приложения (при обращении к нему из Web-сервиса – переменная `КаталогДистрибутивов` и при обращении к нему через веб-сервис – переменная `URLДистрибутивов`).

Имя файла с дистрибутивом должно иметь имя `tc-windows-x86-8.3.3.100.zip` или аналогичное (в зависимости от вида ОС и архитектуры запрашиваемого клиентского приложения). Имя файла с архивом определяется программным кодом приведенного выше демонстрационного Web-сервиса.

После создания конфигурации следует опубликовать Web-сервис на веб-сервере (см. [здесь](#)). Затем необходимо запомнить адрес опубликованного Web-сервиса. Предположим, что Web-сервис опубликован по адресу <http://localhost/getdistr>.

В форме настройки окна запуска (см. [здесь](#)) необходимо добавить Интернет-сервис с указанным выше адресом, с указанием суффикса `ws`: <http://localhost/getdistr/ws>. Теперь при необходимости получения дистрибутива будет выполнен запрос у Web-сервису. Если в каталоге `КаталогДистрибутивов` (на компьютере в Web-сервисом) будет zip-архив с нужным дистрибутивом – этот файл будет передан на компьютер, который запросил дистрибутив.

Приложение 7. Параметры командной строки запуска «1С:Предприятия»

7.1. Выбор режима запуска

При запуске может использоваться один из режимов запуска, перечисленных ниже. Одновременное использование нескольких режимов не допускается.

7.1.1. Запуск системы «1С:Предприятие» в режиме 1С:Предприятие

[Копировать в буфер обмена](#)

```
1cv8 ENTERPRISE [<параметры запуска>]
```

Параметры запуска могут включать в себя:

- Общие параметры запуска – подробнее см. [здесь](#).

7.1.2. Запуск системы «1С:Предприятие» в режиме Конфигуратора

[Копировать в буфер обмена](#)

```
1cv8 DESIGNER [<параметры запуска>]
```

Параметры запуска могут включать в себя:

- Общие параметры запуска – подробнее см. [здесь](#).
- Один из следующих наборов параметров:
 - Параметры запуска конфигулятора – подробнее см. [здесь](#).
 - Параметры создания файлов поставки и обновления – подробнее см. [здесь](#).
 - Параметры работы с хранилищем конфигурации – подробнее см. [здесь](#).
 - Регистрация «1С:Предприятие» в качестве OLE-Automation-сервера – подробнее см. [здесь](#).

7.1.3. Запуск системы «1С:Предприятие» в режиме создания информационной базы

[Копировать в буфер обмена](#)

```
1cv8 CREATEINFORMBASE <строка соединения> [/AddInList [<имя ИБ>]] [/UseTemplate <имя файла шаблона>]
```

При запуске приложения используются следующие параметры:

- <Строка соединения> – строка, задающая параметры доступа к базе данных. Подробное описание строки соединения см. [здесь](#).
- /AddInList – параметр, указывающий, что создаваемую информационную базу необходимо добавить в список баз. База добавляется под именем <имя ИБ>. Если <имя ИБ> не указано, то используется имя по умолчанию, аналогичное тому, какое формируется при интерактивном добавлении информационной базы в список баз.

Если параметр не указан, база не будет добавлена в список.

- /UseTemplate – создание информационной базы осуществляется по шаблону, указанному в <имя файла шаблона>. В качестве шаблонов могут быть файлы конфигурации (.cf) или файлы выгрузки информационной базы (.dt). Если шаблон не указан, параметр игнорируется.

7.2. Общие параметры запуска

```
/F <каталог>
```

Каталог, в котором расположен файл базы данных 1Cv8.1CD.

```
/S <адрес>
```

Адрес информационной базы, хранящейся на сервере «1С:Предприятия». Формируется как

<Имя компьютера, работающего сервером приложений>\<Ссылочное имя информационной базы, известное в рамках сервера "1С:Предприятия">.

```
/WS <url>
```

Строка ws-соединения.

```
/IBName <имя информационной базы>
```

Запуск информационной базы по имени базы в списке информационных баз (подробнее см. [здесь](#)). При необходимости имя должно быть взято в двойные кавычки. При этом если имя содержит двойные кавычки, то нужно заменить каждые двойные кавычки на две двойные кавычки. Если в списках баз будет найдено более чем одна база с указанным именем, то пользователю будет выдана ошибка.

```
/IBConnectionString
```

Позволяет задать строку соединения с информационной базой целиком в том виде, в котором ее возвращает функция `СтрокаСоединенияИнформационнойБазы()`. Части строки соединения могут быть переопределены ранее существовавшими параметрами. Для этого нужно, чтобы `/IBConnectionString` находился в командной строке раньше них. Передавая строку соединения в качестве параметра командной строки, нужно помнить о том, что строка соединения содержит кавычки. Поэтому требуется взять всю строку в кавычки, а содержащиеся внутри кавычки удвоить.

Дополнительные параметры строки соединения в режиме тонкого клиента:

- `wsn` – имя пользователя для аутентификации на веб-сервере;

- `wsp` – пароль пользователя для аутентификации на веб-сервере;
- `wspauto` – использовать автоматические настройки прокси-сервера;
- `wspsrv` – адрес прокси-сервера;
- `wspport` – порт прокси;
- `wspuser` – имя пользователя для прокси с авторизацией;
- `wsppwd` – пароль для прокси с авторизацией.

`/N <имя>`

Имя пользователя. Должно быть указано так же, как в списке пользователей, создаваемом в Конфигураторе.

`/P <пароль>`

Пароль пользователя, имя которого указано в параметре `/N`. Если у пользователя нет пароля, этот параметр можно опустить.

`/AppAutoCheckVersion`

Выполнять автоматический подбор нужной версии для каждой информационной базы. Подробнее см. [здесь](#).

`/AppAutoCheckMode`

Выполнять автоматическое определение приложения на основе данных информационной базы (основного режима запуска для конфигурации и режима запуска для пользователя). Подробнее см. [здесь](#).

`/NoProxy`

Запретить использование прокси (только для ws-соединения).

`/Proxy -PSrv <адрес прокси> -PPort <порт> [-PUser <имя пользователя прокси> [-PPwd <пароль>]]`

Использовать указанные настройки прокси, игнорируя умолчания (только для ws-соединения).

`/LogUI`

Выполнить логирование действий пользователя.

`/WA<режим>`

Определяет режим использования аутентификации операционной системы при запуске «1С:Предприятие». Если параметр `/WA` не указывается, то подразумевается, что используется параметр командной строки `/WA+`.

`<режим>` может принимать следующие значения:

- `+` – установка обязательного применения аутентификации операционной системы при старте «1С:Предприятие».
- `-` – запрет применения аутентификации операционной системы при старте «1С:Предприятия».

`/WSA<режим>`

Определяет режим использования аутентификации операционной системы на веб-сервере. Если параметр `/WSA` не указывается, то подразумевается, что используется параметр командной строки `/WSA+`.

`<режим>` может принимать следующие значения:

- `+` – принудительное выполнение аутентификации средствами операционной системы на веб-сервере (по умолчанию).
- `-` – запрет применения аутентификации средствами операционной системы на веб-сервере.

`/WSN <имя>`

Имя пользователь операционной системы, которое будет использовать системой для выполнения аутентификации на веб-сервере в случае указания параметра `/WSA+`. Пароль пользователя указывается в параметре `/WSP`.

`/WSP <пароль>`

Пароль пользователя, имя которого указано в параметре `/WSN`, для аутентификации на веб-сервере.

`/OIDA<режим>`

Применение сквозной аутентификации пользователя между разными информационными базами и/или внешними ресурсами для тонкого и веб-клиентов. Если при запуске клиента параметр `/OIDA` не задан, или задан параметр `/OIDA+`, то производится попытка аутентификации через OpenID-провайдера, адрес которого задан в файле `deafault.vrd` публикации этой информационной базы.

Если OpenID-провайдер требует интерактивной аутентификации (происходит первое обращение или истекло время жизни признака аутентифицированности), клиент отображает диалог для ввода имени и пароля пользователя.

Аутентификация происходит по списку пользователей информационной базы OpenID-провайдера.

Аутентифицируемый пользователь информационной базы, использующих OpenID-аутентификацию, должен соответствовать по имени пользователю информационной базы OpenID-провайдера.

`<режим>` может принимать следующие значения:

- `+` – использовать OpenID-аутентификацию (по умолчанию).
- `-` – не использовать OpenID-аутентификацию.

`/Authoff`

Выполняет операцию OpenID logout (завершение сеанса работы пользователя). Завершение сеанса работы выполняется вне зависимости от используемого в дальнейшем метода аутентификации.

`/UseHwLicenses<режим>`

Определяет режим поиска локального ключа защиты.

`<режим>` может принимать следующие значения:

- `+` – поиск локального ключа защиты выполняется.
- `-` – поиск локального ключа защиты не выполняется.

`/Out <имя файла> [-NoTruncate]`

Установка файла для вывода служебных сообщений. Если задан параметр `-NoTruncate` (через пробел), файл не очищается (не используется в тонком клиенте).

Во время исполнения пакетных команд файл сообщений можно открыть для просмотра. Запись сообщений в файл не буферизуется (сообщения записываются сразу).

`/L<код языка>`

Указывается код языка интерфейса платформы. Поддерживаемые языки интерфейса (`<код языка>`) см. [здесь](#).

`/VL<код локализации сеанса>`

Указывается код локализации сеанса, используемый при форматировании данных типа `Число` и `Дата`, а также в методах `ЧислоПропись()` и `ПредставлениеПериода()`.

`/DisableStartupMessages`

Подавляет стартовые сообщения:

- [Конфигурация базы данных не соответствует сохраненной конфигурации. Продолжить?](#);
- [Возможностей Вашего компьютера недостаточно для редактирования справки по конфигурации. Для редактирования справки необходимо установить Microsoft Internet Explorer версии 7.0 или выше.](#);
- [Возможностей Вашего компьютера недостаточно для редактирования html-документов, в том числе разделов справки. Для редактирования html-документов необходимо установить Microsoft Internet Explorer версии 7.0 или выше. В данном запуске редактирование html-документов будет недоступно..](#)

`/C <строка текста>`

Передача параметра в прикладное решение.

`/DisableStartupDialogs`

Подавляет вызов стартового диалога и диалогов аутентификации. При этом:

- Если командной строки недостаточно для выбора информационной базы или определения режима запуска, выдается ошибка.
- Если командной строки недостаточно для авторизации пользователя в информационной базе, выдается ошибка.
- Если в командной строке отсутствует авторизация в хранилище конфигурации, конфигуратор запускается без связи с хранилищем.
- Если в командной строке ошибочная авторизация в хранилище конфигурации, выдается ошибка.

`/AllowExecuteScheduledJobs -Off|-Force`

Управление запуском регламентных заданий. Регламентные задания начинают выполняться на первом запущенном по порядку клиенте, у которого не `/AllowExecuteScheduledJobs -Off`. После завершения сеанса этого клиента, выполнение переходит к какому-либо из других запущенных сеансов. Если запускается сеанс с `/AllowExecuteScheduledJobs -Force`, то регламентные задания начинают выполняться на нем, не зависимо от наличия других сеансов.

`/RunModeOrdinaryApplication`

Запуск толстого клиента в обычном режиме, несмотря на настройки конфигурации и пользователя, от имени которого будет выполняться запуск. Не используется в тонком клиенте.

`/RunModeManagedApplication`

Запуск толстого клиента в режиме управляемого приложения, при этом учитывается настройка клиента в списке информационных баз (настройка [Основной режим запуска](#), см. [здесь](#)):

- [Выбирать автоматически](#) – запускается тонкий клиент;

- **Тонкий клиент** – запускается тонкий клиент;
- **Веб-клиент** – запускается веб-клиент;
- **Толстый клиент** – запускается толстый клиент в режиме управляемого приложения.

При этом при запуске клиента, отключается механизм автоматического подбора клиентского приложения.

/EnableCheckModal

Включает режим строгой проверки использования модальных методов. Более подробную информацию о работе параметра можно получить в книге «1С:Предприятие 8.3. Руководство разработчика».

/EnableCheckExtensionsAndAddInsSyncCalls

Включает режим строгой проверки использования синхронных вызовов расширений работы с файлами и криптографией и внешних компонент. Более подробную информацию о работе параметра можно получить в книге «1С:Предприятие 8.3. Руководство разработчика».

/UC <код доступа>

Позволяет выполнить установку соединения с информационной базой, на которую установлена блокировка установки соединений. Если при установке блокировки задан непустой код доступа, то для установки соединения необходимо в параметре */UC* указать этот код доступа.

Не используется при работе тонкого клиента через веб-сервер.

/RunShortcut <имя файла>

Позволяет запустить систему «1С:Предприятие» со списком информационных баз, полученным с помощью указанного файла. В качестве файла может быть указан файл списка общих информационных баз (*.v8i), или файл ярлыка информационных баз (*.v8l).

/TestManager

Запуск толстого и тонкого клиента для управления клиентами тестирования с помощью специализированной объектной модели.

/TestClient [-TPort<Номер tcp-порта>]

Запуск толстого и тонкого клиента как клиента тестирования.

-TPort<Номер tcp-порта> указывает номер порта для взаимодействия клиента и менеджера тестирования. По умолчанию используется порт **1538**.

/Z "<Общий реквизит 1>,<Общий реквизит 2>,...,<Общий реквизит N>"

Установка разделителей при запуске клиентского приложения.

Подробное описание приведено в книге «1С:Предприятие 8.3. Руководство разработчика».

/itdi

Запуск в режиме интерфейса с использованием закладок.

/isdi

Запуск в режиме интерфейса с использованием отдельных окон.

/iTaxi

Запуск в режиме интерфейса **Такси**.

/HttpsCert [-windows] [-recent] [-auto] [-choose] [-file <path>] [-pwd <password>] [-none]

Указывает источник клиентского сертификата. Если указан ключ */HttpsNSS*, то данный ключ игнорируется. Если данный ключ и ключ */HttpsNSS* не указаны, то источник клиентского сертификата определяется настройками диалога настройки соединения с информационной базой.

Параметры ключа:

- *-windows* – указывает, что при соединении нужно использовать клиентский сертификат из системного хранилища сертификатов операционной системы Microsoft Windows. Данный параметр игнорируется, если указан хотя бы один из следующих параметров ключа: *-file* или *-none*.
- *-recent* – выбирать или использовать ранее выбранный клиентский системный сертификат Microsoft Windows.

Если в системном хранилище пользовательских сертификатов присутствует более одного подходящего сертификата, то пользователю предоставляется выбор используемого сертификата через системный диалог выбора сертификата. В дальнейшем выбранный сертификат используется автоматически.

Данный способ выбора клиентского сертификата выбран по умолчанию для параметра ключа *-windows*, если параметры *-auto* и *-choose* не указаны.

- *-auto* – использовать автоматически выбранный клиентский сертификат из установленных в системном хранилище сертификатов операционной системы Microsoft Windows. Данный параметр игнорируется, если у ключа отсутствует параметр *-windows*.
- *-choose* – всегда выбирать используемый клиентский сертификат Microsoft Windows.

Если в системном хранилище пользовательских сертификатов присутствует более одного подходящего сертификата, то пользователю предоставляется выбор используемого сертификата через системный диалог выбора сертификата независимо от того, был ли ранее выбран какой-либо сертификат. Выбранный сертификат в дальнейшем может использоваться автоматически с опцией *-recent*.

Данную параметр можно указать, если необходимо избежать автоматического использования ранее выбранного клиентского сертификата из

системного хранилища сертификатов операционной системы Microsoft Windows, и выбрать новый сертификат из установленных в системе сертификатов подходящих для данного соединения. Данный параметр игнорируется, если у ключа отсутствует параметр `-windows` или установлен параметр `-auto`.

- `-file <path>` – указывает, что необходимо использовать клиентский сертификат и приватный ключ из указанного файла. Данный параметр игнорируется, если у ключа установлен параметр `-none`.
- `-pwd <password>` – указывает пароль файла, содержащего клиентский сертификат и его приватный ключ. Если сервер требует предоставления клиентского сертификата и файл сертификата защищен паролем, то соединение возможно только при правильно указанном пароле. Данный параметр игнорируется, если у данного ключа не указан параметр `-file`.
- `-none` – указывает, что клиентский сертификат не используется. Соединение возможно только с серверами, не требующими проверки клиентского сертификата.

Если ни один из параметров `-windows`, `-file` или `-none` не указан, то ключ `/HttpsCert` игнорируется.

```
/HttpsCA [-windows] [-file <path>] [-pwd <password>] [-none]
```

Указывает источник сертификатов удостоверяющих центров, используемых для проверки сертификата сервера. Если указан ключ `/HttpsNSS`, то данный ключ игнорируется. Если данный ключ и ключ `/HttpsNSS` не указаны, то источник сертификатов удостоверяющих центров определяется настройками диалога настройки соединения с информационной базой.

Параметры ключа:

- `-windows` – указывает, что для проверки сертификата сервера при соединении нужно использовать сертификаты удостоверяющих центров из системного хранилища сертификатов операционной системы Microsoft Windows. Данный параметр игнорируется, если установлен хотя бы один из параметров ключа `-file` или `-none`.
- `-file <path>` – указывает, что для проверки сертификата сервера при соединении нужно использовать сертификаты удостоверяющих центров, загружаемые из указанного файла. Данный параметр игнорируется, если установлен параметр ключа `-none`.
- `-pwd <password>` – пароль файла, содержащего корневые сертификаты. Если файл сертификата защищен паролем, то соединение возможно только при правильно указанном пароле. Данный параметр игнорируется, если у данного ключа не указан параметр `-file`.
- `-none` – указывает, что корневые сертификаты не используются, и сертификат сервера не проверяется.

Если ни один из параметров `-windows`, `-file` или `-none` не установлен, то ключ `/HttpsCA` игнорируется.

```
/HttpsNSS [-path <path>] [-pwd <password>] [-certName <name>] [-verifySrvCert]
```

Настройки хранилища сертификатов NSS (<https://developer.mozilla.org/ru/docs/NSS>, на английском языке). Указывает, что необходимо использовать клиентский сертификат и (опционально) корневые сертификаты из хранилища сертификатов NSS.

Параметры ключа:

- `-path <path>` – путь каталога хранилища сертификатов NSS, содержащего файлы баз сертификатов и ключей NSS (`cert9.db` и `key4.db`, или `cert8.db` и `key3.db`). Если данный параметр не указан или его значение пустая строка, то ключ `/HttpsNSS` игнорируется.

Для баз сертификатов веб-браузера Mozilla Firefox по умолчанию используются каталоги:

- в ОС Microsoft Windows: `%AppData%\Mozilla\Firefox\Profiles\[profile name]`;
- в ОС Linux: `~/mozilla/firefox/[profile name]`.

`[profile name]` – это имя каталога профиля пользователя веб-браузера Mozilla Firefox.

Для баз сертификатов веб-браузера Chromium в операционной системе Linux это каталог `~/pki/nssdb`.

- `-pwd <password>` – указывает пароль для хранилища сертификатов NSS. Если открывается защищенная паролем база, то соединение возможно только с серверами, не требующими предоставления клиентского сертификата.
- `-certName <name>` – имя конкретного клиентского сертификата в хранилище NSS. Если не указано, то автоматически выбирается подходящий для данного соединения клиентский сертификат. Для использования хранилищ сертификатов веб-браузеров Mozilla Firefox и Chromium устанавливать данную опцию не рекомендуется, т. к. имя сертификата зависит от используемого веб-браузера и может быть нечитаемым.
- `-verifySrvCert` – указывает необходимость проверки сертификата сервера, используя корневые сертификаты установленные в данном хранилище сертификатов NSS.

```
/AppAutoInstallLastVersion<режим>
```

Управляет возможность автоматической установки новых версий клиентского приложения.

`<режим>` может принимать следующие значения:

- `+` – установка новых версий включена.
- `-` – установка новых версий выключена.

```
/URL <адрес>
```

Указывает необходимость перехода по ссылке. Поддерживаются ссылки формата `e1c` и `http(s)`:

- Если указана внешняя ссылка – выполняется поиск запущенного клиентского приложения с той же строкой соединения, которая указана в параметре. В найденном клиентском приложении не должно быть открыто модальное или блокирующее окно. После этого выполняется попытка перехода по локальной ссылке из исходной навигационной ссылки и активизируется основное окно приложения. В случае неудачи клиентское приложение продолжает работу. Если исходная навигационная ссылка не содержит локальной ссылки (содержит только адрес информационной

базы), то попытка перехода не выполняется, активируется основное окно найденного клиентского приложения.

- Если подходящего клиентского приложения не найдено, строка соединения определяется из параметра командной строки `/URL`.
- Если указана локальная ссылка – клиентское приложение запускается в общем порядке. После запуска выполняется попытка перехода по переданной локальной ссылке.

Для ссылок формата [`http\(s\)`](http(s)) всегда запускается (или находится активный) тонкий клиент.

```
/HttpsForceSSLv3
```

Указывает на необходимость принудительного использования протокола SSL версии 3.0 тонким клиентом при работе через веб-сервер с помощью протокола [`https`](https).

7.3. Параметры командной строки пакетного режима запуска

Параметры, перечисленные в данном разделе (и его подразделах) не могут комбинироваться в рамках одной командной строки запуска, если об этом явно не сказано обратного. Параметры доступа к хранилищу конфигурации (`/ConfigurationRepositoryF`, `/ConfigurationRepositoryN` и `/ConfigurationRepositoryP`) могут сочетаться с другими параметрами данного раздела.

В том случае, когда параметры командной строки пакетного режима запуска конфигулятора требуют указания имени файла, следует учитывать, что при указании имени файла с полным путем, все каталоги, входящие в состав пути, должны существовать. В противном случае операция не будет выполнена.

7.3.1. Параметры запуска конфигулятора

В том случае, если параметр командной строки поддерживает ключи `-Extension` и `-AllExtensions`, одновременное указание обоих ключей не поддерживается и поведение системы в таком случае не определено.

При работе с расширениями (ключи `-Extension` и `-AllExtension`) при успешном завершении работы код возврата будет установлен в значение 0, в противном случае код возврата будет принимать значение 1.

7.3.1.1. Выгрузка информационной базы

```
/DumpIB <имя файла>
```

Выполнить выгрузку информационной базы в файл.

```
/RestoreIB <имя файла>
```

Выполнить загрузку информационной базы из файла.

7.3.1.2. Конфигурация и расширения

```
/DumpCfg <имя cf/cfe файла> [-Extension <Имя расширения>]
```

Выполнить сохранение конфигурации или расширения конфигурации в файл. Сохранение конфигурации расширения будет выполняться при корректном задании ключа `-Extension`.

```
/LoadCfg <имя cf/cfe файла> [-Extension <Имя расширения>]
```

Выполнить загрузку конфигурации или расширения конфигурации из файла. Загрузка конфигурации расширения будет выполняться при корректном задании ключа `-Extension`. Если в момент загрузки расширение отсутствует в информационной базе – оно будет создано с указанным именем.

```
/UpdateDBCfg [-Dynamic<Режим>] [-BackgroundStart] [-BackgroundCancel] [-BackgroundFinish [-Visible]] [-BackgroundSuspend] [-BackgroundResume] [-WarningsAsErrors] [-Server] [-Extension <Имя расширения>]
```

Выполнить обновление конфигурации базы данных.

Допустимо использовать следующие ключи:

- `-Dynamic<Режим>` – признак использования динамического обновления. Режим может принимать следующие значения:
 - `--` – явно запрещает динамическое обновление.
 - `+` – разрешает динамическое обновление. Вначале выполняется попытка выполнить обычное обновление, если попытка не удалась – происходит попытка выполнить динамическое обновление. Динамическое обновление будет разрешено также без указания ключа `-Dynamic+` или при использовании ключа `-Dynamic` без указания режима.
- `-BackgroundStart` – запускает фоновое обновление конфигурации базы данных и завершает работу. Если дополнительно указан ключ `-Dynamic` или `-Dynamic+`, то вначале будет выполнена попытка динамического обновления и если эта попытка завершилась неудачно, то будет запущено фоновое обновление.
- `-BackgroundCancel` – отменяет ранее запущенное фоновое обновление конфигурации базы данных.
- `-BackgroundFinish` – завершает фоновое обновление конфигурации базы данных (выполняет фазу принятия изменений): выполняется попытка наложить монопольную блокировку базы данных и выполнить финальную фазу. При указании флага `-Visible` выдается диалоговое окно с кнопками [Отмена](#), [Повторить](#), [Завершить сеансы и повторить](#) в случае невозможности выполнить завершение фонового обновления (перейти к фазе принятия изменений). Если флаг не указан – выполнение завершается с ошибкой.
- `-BackgroundSuspend` – приостанавливает фоновое обновление конфигурации базы данных.
- `-BackgroundResume` – продолжает фоновое обновление конфигурации базы данных, приостановленное ранее.
- `-WarningsAsErrors` – все предупредительные сообщения будут трактоваться как ошибки.

• **-Server** – обновление будет выполняться на сервере (имеет смысл только в клиент-серверном варианте работы). Если ключ используется вместе с фоновым обновлением, то:

- Фаза актуализации всегда выполняется на сервере.
- Фаза обработки и фаза принятия изменений могут выполняться как на клиенте, так и на сервере.
- Допускается запуск фонового обновления на стороне клиента, а завершение на стороне сервера (как и наоборот).
- **-Extension <Имя расширения>** – будет выполнено обновление указанного расширения.

Параметр **/UpdateDBCfg** допускается указывать **после** следующих параметров:

- **/LoadCfg;**
- **/UpdateCfg;**
- **/ConfigurationRepositoryUpdateCfg;**
- **/LoadConfigFiles;**
- **/MAUpdatePublication;**
- **/MAWriteFile.**

/DumpDBCfg <имя cf/cfe файла> [**-Extension** <Имя расширения>]

Выполнить сохранение конфигурации базы данных или конфигурации расширения, сохраненного в базу данных, в файл. Сохранение конфигурации расширения будет выполняться при корректном задании ключа **-Extension**.

/DumpDBCfgList [**-Extension** <Имя расширения>] [**-AllExtensions**]

Выводит имя основной конфигурации (если не указан не один ключ) или имя расширения(-ий). Допустимо использование следующих ключей:

- **-Extension** – выводит имя указанного расширения.
- **-AllExtensions** – выводит имена всех расширений.

/RollbackCfg [**-Extension** <Имя расширения>]

Выполнить возврат к конфигурации базы данных.

В случае указания ключа **-Extension** будет выполнен возврат к конфигурации, сохраненной в базе данных, для указанного расширения.

/DeleteCfg [**-Extension** <Имя расширения>] [**-AllExtensions**]

Выполняет удаление расширения с указанным именем. При указании ключа **-AllExtensions** будет выполнено удаление всех расширений. Использование параметра без ключа не допускается.

/DumpConfigFiles <каталог выгрузки> [**-Module**] [**-Template**] [**-Help**] [**-AllWritable**] [**-Picture**] [**-Right**] [**-Extension** <Имя расширения>]

Позволяет осуществлять выгрузку некоторых свойств объектов конфигурации (модулей, макетов, картинки, права доступа и справочной информации) в файлы. Допустимо указывать следующие каталоги и использовать следующие ключи:

- <каталог выгрузки> – каталог расположения файлов свойств;
- **-Module** – признак необходимости выгрузки модулей;
- **-Template** – признак необходимости выгрузки макетов;
- **-Help** – признак необходимости выгрузки справочной информации;
- **-AllWritable** – признак выгрузки свойств только доступных для записи объектов;
- **-Picture** – признак выгрузки общих картинок;
- **-Right** – признак выгрузки прав.
- **-Extension** – выгрузка будет выполнена для указанного расширения.

/LoadConfigFiles <каталог выгрузки> [**-Module**] [**-Template**] [**-Help**] [**-AllWritable**] [**-Picture**] [**-Right**] [**-Extension** <Имя расширения>]

Позволяет осуществлять загрузку некоторых свойств объектов конфигурации (модулей, макетов, картинки, права доступа и справочной информации) из файлов. Допустимо указывать следующие каталоги и использовать следующие ключи:

- <каталог выгрузки> – каталог расположения файлов свойств;
- **-Module** – признак необходимости выгрузки модулей;
- **-Template** – признак необходимости выгрузки макетов;
- **-Help** – признак необходимости выгрузки справочной информации;
- **-AllWritable** – признак выгрузки свойств только доступных для записи объектов;

- `-Picture` – признак выгрузки общих картинок;

- `-Right` – признак выгрузки прав.

- `-Extension` – загрузка будет выполнена в указанное расширение.

Если команда пакетного режима запуска прошла успешно, возвращает код возврата 0, в противном случае – 1 (101, если в данных имеются ошибки).

```
/DumpConfigToFiles <каталог выгрузки> [-Extension <Имя расширения>] [-AllExtensions]
```

Выполнить выгрузку конфигурации в файлы. Допустимо указывать следующие ключи:

- `-Extension` – выполняется выгрузка указанного расширения.

- `-AllExtensions` – выгружаются все расширения, при этом основная конфигурация не выгружается. Каждое расширение выгружается в каталог со своим именем.

```
/LoadConfigFromFiles <каталог загрузки> [-Extension <Имя расширения>] [-AllExtensions]
```

Выполнить загрузку конфигурации из файлов. Загрузка расширения в основную конфигурацию (и наоборот) не поддерживается. Допустимо указывать следующие ключи:

- `-Extension` – выполняется загрузка указанного расширения. Если расширения нет – оно создается.

- `-AllExtensions` – выполняется загрузка расширений из файлов. Каждый подкаталог в указанном каталоге считается расширением.

7.3.1.3. Мобильное приложение

```
/MAUpdatePublication
```

Обновить публикацию мобильного приложения, если она была ранее создана, в противном случае будет выдана ошибка. Возможно предварительное обновление конфигурации базы данных.

```
/MAWriteFile <имя xml файла>
```

Сохранить конфигурацию в xml-файл, который можно использовать для сборки мобильного приложения. Возможно предварительное обновление конфигурации базы данных.

7.3.1.4. Проверки конфигурации и расширений

```
/CheckModules [-ThinClient] [-WebClient] [-Server] [-ExternalConnection] [-ThickClientOrdinaryApplication] [-MobileAppClient] [-MobileAppServer] [-ExtendedModulesCheck] [-Extension <Имя расширения>] [-AllExtensions]
```

Выполнить проверка программных модулей. Должен быть указан один или несколько ключей режимов проверки. Если ключи не указаны, проверка выполнена не будет. Допустимо использование следующих ключей:

- `-ThinClient` – проверка в режиме работы тонкого клиента;

- `-WebClient` – проверка в режиме работы веб-клиента;

- `-Server` – проверка в режиме работы сервера «1С:Предприятия»;

- `-ExternalConnection` – проверка в режиме работы внешнего соединения;

- `-ThickClientOrdinaryApplication` – проверка в режиме работы клиентского приложения;

- `-MobileAppClient` – проверка в режиме работы клиента мобильного приложения;

- `-MobileAppServer` – проверка в режиме работы сервера мобильного приложения;

- `-ExtendedModulesCheck` – проверка обращений к методам и свойствам объектов «через точку» (для ограниченного набора типов); проверка правильности строковых литералов – параметров некоторых функций, таких как `ПолучитьФорму()`.

- `-Extension` – выполнить заданные проверки для указанного расширения.

- `-AllExtensions` – выполнить заданные проверки для всех расширений.

```
/IBCheckAndRepair [-ReIndex] [-LogIntegrity | -LogAndRefsIntegrity] [-RecalcTotals] [-IBCompression] [-Rebuild] [-TestOnly | [-BadRefCreate | -BadRefClear | -BadRefNone] [-BadDataCreate | -BadDataDelete] [-UseStartPoint] [-TimeLimit:hh:mm]]
```

Выполнить операцию тестирования и исправления информационной базы. Допустимо использование следующих ключей:

- `-ReIndex` – реиндексация таблиц;

- `-LogIntegrity` – проверка логической целостности;

- `-LogAndRefsIntegrity` – проверка логической и ссылочной целостности;

- `-RecalcTotals` – пересчет итогов;

- `-IBCompression` – сжатие таблиц;

- `-Rebuild` – реструктуризация таблиц информационной базы;

- **-TestOnly** – только тестирование при наличии ссылок на несуществующие объекты:
 - **-BadRefCreate** – создавать объекты;
 - **-BadRefClear** – очищать объекты;
 - **-BadRefNone** – не изменять при частичной потере объектов:
 - **-BadDataCreate** – создавать объекты;
 - **-BadDataDelete** – удалять объекты;
- **-UseStartPoint** – использовать сохраненную точку возврата для продолжения тестирования с того места, на котором оно было прервано в предыдущем сеансе;
- **-TimeLimit:hhh:mm** – ограничение максимального времени сеанса тестирования:
 - **hhh** – количество часов (0..999);
 - **mm** – количество минут (0..59).

Одновременное использование параметров внутри подгруппы параметров не допускается.

```
/CheckConfig [-ConfigLogIntegrity] [-IncorrectReferences] [-ThinClient] [-WebClient] [-Server] [-ExternalConnection] [-ExternalConnectionServer] [-MobileAppClient] [-MobileAppServer] [-ThickClientManagedApplication] [-ThickClientServerManagedApplication] [-ThickClientOrdinaryApplication] [-ThickClientServerOrdinaryApplication] [-DistributiveModules] [-UnreferenceProcedures] [-HandlersExistence] [-EmptyHandlers] [-ExtendedModulesCheck] [-CheckUseModality] [-UnsupportedFunctional] [-Extension <Имя расширения>] [-AllExtensions]
```

Выполнить централизованную проверку конфигурации. Допустимо использование следующих ключей:

- **-ConfigLogIntegrity** – проверка логической целостности конфигурации. Стандартная проверка, обычно выполняемая перед обновлением базы данных;
- **-IncorrectReferences** – поиск некорректных ссылок. Поиск ссылок на удаленные объекты. Выполняется по всей конфигурации, включая права, формы, макеты, интерфейсы и т. д. Также осуществляется поиск логически неправильных ссылок;
- **-ThinClient** – синтаксический контроль модулей для режима эмуляции среды управляемого приложения (тонкий клиент), выполняемого в файловом режиме;
- **-WebClient** – синтаксический контроль модулей в режиме эмуляции среды веб-клиента;
- **-Server** – синтаксический контроль модулей в режиме эмуляции среды сервера «1С:Предприятия»;
- **-ExternalConnection** – синтаксический контроль модулей в режиме эмуляции среды внешнего соединения, выполняемого в файловом режиме;
- **-ExternalConnectionServer** – синтаксический контроль модулей в режиме эмуляции среды внешнего соединения, выполняемого в клиент-серверном режиме;
- **-MobileAppClient** – синтаксический контроль модулей в режиме эмуляции среды мобильного приложения, выполняемого в клиентском режиме запуска;
- **-MobileAppServer** – синтаксический контроль модулей в режиме эмуляции среды мобильного приложения, выполняемого в серверном режиме запуска;
- **-ThickClientManagedApplication** – синтаксический контроль модулей в режиме эмуляции среды управляемого приложения (толстый клиент), выполняемого в файловом режиме;
- **-ThickClientServerManagedApplication** – синтаксический контроль модулей в режиме эмуляции среды управляемого приложения (толстый клиент), выполняемого в клиент-серверном режиме;
- **-ThickClientOrdinaryApplication** – синтаксический контроль модулей в режиме эмуляции среды обычного приложения (толстый клиент), выполняемого в файловом режиме;
- **-ThickClientServerOrdinaryApplication** – синтаксический контроль модулей в режиме эмуляции среды обычного приложения (толстый клиент), выполняемого в клиент-серверном режиме;
- **-ExternalConnection** – синтаксический контроль модулей в режиме эмуляции среды внешнего соединения, выполняемого в файловом режиме;
- **-ExternalConnectionServer** – синтаксический контроль модулей в режиме эмуляции среды внешнего соединения, выполняемого в клиент-серверном режиме;
- **-DistributiveModules** – поставка модулей без исходных текстов. В случае если в настройках поставки конфигурации для некоторых модулей указана поставка без исходных текстов, проверяется возможность генерации образов этих модулей;
- **-UnreferenceProcedures** – поиск неиспользуемых процедур и функций. Поиск локальных (не экспортных) процедур и функций, на которые отсутствуют ссылки. В том числе осуществляется поиск неиспользуемых обработчиков событий;
- **-HandlersExistence** – проверка существования назначенных обработчиков. Проверка существования обработчиков событий интерфейсов, форм и элементов управления;
- **-EmptyHandlers** – поиск пустых обработчиков. Поиск назначенных обработчиков событий, в которых не выполняется никаких действий. Существование таких обработчиков может привести к снижению производительности системы;
- **-ExtendedModulesCheck** – проверка обращений к методам и свойствам объектов «через точку» (для ограниченного набора типов); проверка правильности строковых литералов – параметров некоторых функций, таких как `ПолучитьФорму()`;

- `-CheckUseModality` – режим поиска использования в модулях методов, связанных с модальностью. Ключ используется только вместе с ключом `-ExtendedModulesCheck`.

- `-UnsupportedFunctional` – выполняется поиск функциональности, которая не может быть выполнена на мобильном приложении. Проверка в этом режиме показывает:

- наличие в конфигурации метаданных, классы которых не реализованы на мобильной платформе;
- наличие в конфигурации планов обмена, у которых установлено свойство [Распределенная информационная база](#);
- использование типов, которые не реализованы на мобильной платформе:
 - в свойствах [Тип](#) реквизитов метаданных, констант, параметров сеанса;
 - в свойстве [Тип параметра команды](#) объекта конфигурации [Команда](#);
 - в свойстве [Тип](#) реквизитов и колонок реквизита формы;
 - наличие форм с типом формы [Обычная](#);
 - наличие в форме элементов управления, которые не реализованы на мобильной платформе. Проверка не выполняется для форм, у которых свойство [Назначения использования](#) не предполагает использование на мобильном устройстве;
 - сложный состав рабочего стола (использование более чем одной формы).
- `-Extension` – выполнить заданные проверки для указанного расширения.
- `-AllExtensions` – выполнить заданные проверки для всех расширений.

7.3.1.5. Поддержка конфигурации

```
/UpdateCfg <имя cf или cfu файла>
```

Выполнить обновление конфигурации, находящейся на поддержке.

7.3.1.6. Журнал регистрации

```
/ReduceEventLogSize <Date> [-saveAs <имя файла>] [-KeepSplitting]
```

Выполнить сокращение журнала регистрации. Допустимо использование следующих ключей:

- `Date` – новая граница журнала регистраций в формате [ГГГГ-ММ-ДД](#);
- `-saveAs <имя файла>` – параметр для сохранения копии выгружаемых записей;
- `-KeepSplitting` – требуется сохранить разделение на файлы по периодам.

7.3.1.7. Удаление данных

```
/EraseData [/Z[<разделители>]]
```

Выполнить удаление данных информационной базы. С помощью параметра `/z` задается область, в которой будут удалены данные. Удаление данных доступно пользователю с правом [Администрирование](#).

Удаление данных информационной базы осуществляется из:

- таблиц, определяемых структурой метаданных;
- хранилищ настроек;
- истории;
- административных параметров информационной базы;
- списка пользователей;
- часового пояса (если в конфигурации не определено независимых разделителей, то становится неопределенным, если определен хотя бы один независимый разделитель, то удаляется).

Если в сеансе не используется ни один разделитель, то следующие настройки информационной базы заполняются значениями по умолчанию:

- Время ожидания блокировки данных;
- Минимальная длина паролей пользователей;
- Проверка сложности паролей пользователей;
- Разрешение полнотекстового поиска.

7.3.1.8. Предопределенные данные

```
/SetPredefinedDataUpdate [-Auto] [-UpdateAutomatically] [-DoNotUpdateAutomatically]
```

Параметр предназначен для указания режимов обновления предопределенных данных. Допустимо использование следующих ключей:

- `-Auto` – фактическое значение вычисляется автоматически (значение по умолчанию). Для главного узла информационной базы – значение

будет равно `-UpdateAutomatically`, для периферийного узла информационной базы будет равно `-DoNotUpdateAutomatically`.

- `-UpdateAutomatically` – при реструктуризации информационной базы будет выполняться автоматическое создание предопределенных элементов и обновление существующих значений.
- `-DoNotUpdateAutomatically` – при реструктуризации информационной базы не будет выполняться автоматическое создание новых предопределенных элементов и обновление их значений.

7.3.1.9. Распределенная информационная база

`/ResetMasterNode`

Отменить назначение главного узла распределенной информационной базы. Действие параметра аналогично вызову метода `УстановитьГлавныйУзел()` со значением параметра `Неопределено`.

7.3.1.10. Прочие параметры

`/Visible`

Делает исполнение пакетной команды видимым пользователю. На время работы конфигулятора открывается окно заставки.

`/RunEnterprise`

Запустить «1С:Предприятие» после исполнения пакетной команды. После ключа может быть указана дополнительная командная строка. При запуске «1С:Предприятия», переданные в ней параметры будут использованы вместо параметров текущей сессии. Дополнительная командная строка обязательно должна быть в кавычках, вложенные в нее кавычки должны быть удвоены.

`/ConvertFiles <имя файла|путь>`

Выполнить пакетную конвертацию файлов 1С:Предприятия 8.x. `<имя файла|путь>` – имя файла или каталога.

Если задан каталог, осуществляется конвертация всех доступных документов в указанном каталоге и вложенных каталогах. Для успешной конвертации файлы должны быть доступны для записи. Если указанный в качестве параметра файл недоступен для записи, выдается сообщение об ошибке.

В случае режима работы с каталогом, недоступные для записи файлы пропускаются без выдачи сообщений об ошибке.

Для работы данного механизма должен быть запущен конфигулятор и открыта конфигурация, в среде которой будет производиться конвертация. Имя информационной базы и параметры авторизации можно указать через стандартные параметры командной строки. В случае отсутствия таких параметров, будут выданы соответствующие запросы, как и в других аналогичных механизмах командной строки, работающих в режиме Конфигуратор.

`/DumpResult <имя файла>`

Записать результат работы конфигулятора в файл. Результат – число (0 – в случае успеха).

7.3.2. Параметры создания файла поставки и обновления

`/CreateTemplateListFile <имя файла> [-TemplatesSourcePath]`

Создать файла шаблонов конфигураций. Допустимо указывать следующие каталоги и использовать следующие ключи:

- `<имя файла>` – имя файла списка шаблонов конфигураций. Если не указано, создается в заданном каталоге с именем по умолчанию, если указано только имя создается с указанным именем в заданном каталоге. При указании полного пути используется данный путь;
- `-TemplatesSourcePath` – путь для поиска файлов шаблонов конфигураций. Если не задан, берется путь, установленный в системе в диалоге настройке при запуске.

`/CreateDistributive <каталог создания комплекта поставки> -File <имя файла описания комплекта поставки>[-Option <вариант поставки>] [-MakeSetup] [-MakeFiles] [-digisign <имя файла с параметрами лицензирования >]`

Выполнить создание комплектов поставки и файлов комплектов поставки по готовому описанию комплекта поставки.

Из параметров `-MakeSetup` и `-MakeFiles` может быть использован только один. При отсутствии этих параметров используется `-MakeSetup` (т.е. создается комплект поставки). Допустимо указывать следующие каталоги и использовать следующие ключи:

- `<каталог создания комплекта поставки>` – указание каталога создания комплекта поставки или файлов комплекта поставки;
- `-File <файл описания комплекта поставки>` – указание файла описания комплекта поставки;
- `-Option <вариант поставки>` – создать вариант поставки из описания комплекта поставки. По умолчанию используется вариант поставки **Полный**;
- `-MakeSetup` – создавать комплект поставки;
- `-MakeFiles` – создавать файлы комплекта поставки;
- `-digisign <имя файла с параметрами лицензирования>` – указание параметров лицензирования рабочего места пользователя.

`/CreateDistributionFiles [-cfile <имя cf-файла>] [-cfufile <имя cfu-файла>] [-f <имя cf-файла>|-v <версия дистрибутива>+][-digisign <имя файла с параметрами лицензирования>]`

Выполнить создание файлов поставки и обновления. Допустимо указывать следующие каталоги и использовать следующие ключи:

- `-cfile <имя cf-файла>` – указание создать дистрибутив;

- `-cfufile <имя cfu-файла>` – указание создать обновление;
- `-f <имя cf-файла>` – дистрибутив, включаемый в обновление, задан именем;
- `-v <версия дистрибутива>` – дистрибутив, включаемый в обновление, задан версией;
- `-digisign <имя файла с параметрами лицензирования>` – указание параметров лицензирования рабочего места пользователя.

Группа параметров `-f <имя cf-файла>|-v <версия дистрибутива>` повторяется столько раз, сколько файлов дистрибутивов включается в обновление.

7.3.3. Параметры работы с хранилищем конфигурации

`/ConfigurationRepositoryF <каталог хранилища>`

Параметр предназначен для указания пути к хранилищу конфигурации.

`/ConfigurationRepositoryN <имя>`

Параметр предназначен для указания имени пользователя хранилища конфигурации.

`/ConfigurationRepositoryP <пароль>`

Параметр предназначен для указания пароля пользователя хранилища конфигурации.

`/ConfigurationRepositoryDumpCfg <имя cf файла> [-v <номер версии хранилища>]`

Сохранить конфигурацию из хранилища в файл. Допустимо использовать следующие ключи:

- `-v <номер версии хранилища>` – номер версии, если номер версии не указан, или равен -1, будет сохранена последняя версия.

`/ConfigurationRepositoryUpdateCfg [-v <номер версии хранилища>] [-revised]`

Обновить конфигурацию из хранилища. Допустимо использовать следующие ключи:

- `-v <номер версии хранилища>` – номер версии, если номер версии не указан, или равен -1, будет сохранена последняя версия, если конфигурация не подключена к хранилищу, то параметр игнорируется;
- `-revised` – получать захваченные объекты, если потребуется. Если конфигурация не подключена к хранилищу, то параметр игнорируется;
- `-force` – если при пакетном обновлении конфигурации из хранилища должны быть получены новые объекты конфигурации или удалиться существующие, указание этого параметра свидетельствует о подтверждении пользователем описанных выше операций. Если параметр не указан – действия выполнены не будут.

`/ConfigurationRepositoryUnbindCfg [-force]`

Выполнить отключение конфигурации от хранилища конфигурации (у пользователя должны быть административные права в данной информационной базе). Если пользователь аутентифицируется в хранилище (интерактивно или через параметры командной строки), то отключение конфигурации от хранилища также отражается в самом хранилище конфигурации (информация о подключении удаляется), если же пользователь не аутентифицировался в хранилище, то производится только локальное отключение конфигурации от хранилища.

В случае если в конфигурации имеются захваченные объекты, которые были изменены относительно хранилища, то будет выдано соответствующее сообщение и отключения не выполнятся.

`-force` – ключ предназначен для пропуска диалога аутентификации (если не указаны параметры пользователя хранилища), а также для игнорирования наличия захваченных и измененных объектов.

`/ConfigurationRepositoryReport <имя файла> [-NBegin <номер версии>] [-NEnd <номер версии>] [-GroupByObject] [-GroupByComment]`

Выполнить построение отчета по истории хранилища. Если параметры группировки не указаны и режим совместимости конфигурации равен значению **Не использовать**, то отчет формируется с группировкой по версиям. В режимах совместимости **Версия 8.1** и **Версия 8.2.13** отчет формируется с группировкой по объектам. Если конфигурация базы данных отличается от редактируемой по свойству совместимости, при обработке командной строки учитывается значение режима совместимости конфигурации базы данных. Допустимо указывать следующие имена файлов и использовать следующие ключи:

- `<имя файла>` – имя файла, в который выводится отчет;
- `-NBegin` – номер сохраненной версии, от которой начинается строиться отчет;
- `-NEnd` – номер сохраненной версии, по которой строится отчет;
- `-GroupByObject` – признак формирования отчета по версиям с группировкой по объектам;
- `-GroupByComment` – признак формирования отчета по версиям с группировкой по комментарию.

`/ConfigurationRepositoryClearCache`

Выполнить очистку локальной базы данных хранилища конфигурации.

`/ConfigurationRepositoryCreate [-AllowConfigurationChanges -ChangesAllowedRule <Правило поддержки> -ChangesNotRecommendedRule <Правило поддержки>] [-NoBind]`

Создать хранилище конфигурации. Допустимо использовать следующие ключи:

- `-AllowConfigurationChanges` – если конфигурация находится на поддержке без возможности изменения, будет включена возможность изменения.

• `-ChangesAllowedRule <Правило поддержки>` – устанавливает правило поддержки для объектов, для которых изменения разрешены поставщиком. Может быть установлено одно из следующих правил:

- `ObjectNotEditable` – объект поставщика не редактируется,
- `ObjectIsEditableSupportEnabled` – объект поставщика редактируется с сохранением поддержки,
- `ObjectNotSupported` – объект поставщика снят с поддержки.

• `-ChangesNotRecommendedRule` – устанавливает правило поддержки для объектов, для которых изменения не рекомендуются поставщиком. Может быть установлено одно из следующих правил:

- `ObjectNotEditable` – Объект поставщика не редактируется,
- `ObjectIsEditableSupportEnabled` – Объект поставщика редактируется с сохранением поддержки,
- `ObjectNotSupported` – Объект поставщика снят с поддержки.

• `-NoBind` – к созданному хранилищу подключение выполнено не будет.

```
/ConfigurationRepositoryAddUser -User <Имя> -Pwd <Пароль> -Rights <Права> [-RestoreDeletedUser]
```

Создать пользователя хранилища конфигурации. Пользователь, от имени которого выполняется подключение к хранилищу, должен обладать административными правами. Если пользователь с указанным именем существует, то пользователь добавлен не будет. Допустимо использовать следующие ключи:

- `-User` – имя создаваемого пользователя.
- `-Pwd` – пароль создаваемого пользователя.
- `-Rights` – права пользователя. Возможные значения:
 - `ReadOnly` – право на просмотр,
 - `LockObjects` – право на захват объектов,
 - `ManageConfigurationVersions` – право на изменение состава версий,
 - `Administration` – право на административные функции.
- `-RestoreDeletedUser` – если обнаружен удаленный пользователь с таким же именем, он будет восстановлен.

```
/ConfigurationRepositoryCopyUsers -Path <путь> -User <Имя> -Pwd <Пароль> [-RestoreDeletedUser]
```

Выполнить копирования пользователей из другого хранилища конфигурации. Копирование удаленных пользователей не выполняется. Если пользователь с указанным именем существует, то пользователь не будет добавлен. Допустимо использовать следующие ключи:

- `-Path` – путь к хранилищу, из которого выполняется копирование пользователей.
- `-User` – имя пользователя хранилища, из которого выполняется копирование пользователей.
- `-Pwd` – пароль пользователя хранилища, из которого выполняется копирование пользователей.
- `-RestoreDeletedUser` – если обнаружен удаленный пользователь с таким же именем, он будет восстановлен.

```
/ConfigurationRepositoryClearLocalCache
```

Выполнить очистку локального кэша версий конфигурации.

```
/ConfigurationRepositoryBindCfg [-forceBindAlreadyBindedUser] [-forceReplaceCfg]
```

Выполняет подключение ранее неподключенной информационной базы к хранилищу конфигурации. Допустимо использовать следующие ключи:

- `-forceBindAlreadyBindedUser` – выполняет подключение даже в том случае, если для данного пользователя уже есть конфигурация, связанная с данным хранилищем.
- `-forceReplaceCfg` – если конфигурация непустая, данный ключ подтверждает замену конфигурации на конфигурацию из хранилища.

7.4. Регистрация «1С:Предприятие» в качестве OLE-Automation-сервера

```
/RegServer
```

Выполнить регистрацию объектов `V83.Application` и `V83C.Application`.

```
/UnregServer
```

Отменить регистрацию объектов `V83.Application` и `V83C.Application`.

7.5. Строка соединения с информационной базой

Строка соединения – строка, задающая параметры информационной базы, каждый из которых представляет собой фрагмент вида `<Имя параметра>=<Значение>`, где:

- `Имя параметра` – имя параметра;

Фрагменты отделяются друг от друга символами `;`. Если значение содержит пробельные символы, то оно должно быть заключено в двойные кавычки (`"`). Набор параметров определяется вариантом создаваемой информационной базы – файловый или клиент-серверный. Также существует общий набор параметров, пригодный для любого варианта информационной базы.

Строка соединения указывается в форме списка информационных баз под списком, может быть задана в параметре командной строки выбора режима `CREATEINFOBASE`, как параметр методов `СоздатьНачальныйОбраз()`.

7.5.1. Общий набор параметров

Usr

Указывает имя пользователя.

Pwd

Указывает пароль пользователя.

LicDstr

Управляет получением клиентских лицензий через сервер 1С:Предприятия. Значение параметра:

- **Y** – получать клиентскую лицензию через сервер «1С:Предприятия». Если клиентское приложение не получило программную лицензию или аппаратную лицензию из локального ключа HASP или из сетевого ключа HASP, то производится попытка получения клиентской лицензии через сервер 1С:Предприятия.
- **N** – не получать клиентскую лицензию через сервер 1С:Предприятия. Значение по умолчанию.

Z n

Установка значений разделителей прикладного решения.

Подробное описание приведено в книге «1С:Предприятие 8.3. Руководство разработчика».

prmod

Указывает на необходимость выполнить запуск в привилегированном режиме (значение параметра равно 1). Разрешен аутентифицированному пользователю, имеющему административные права. Журнал регистрации фиксирует установку или отказ в возможности установки режима привилегированного сеанса.

7.5.2. Параметры для файлового варианта информационной базы

File

Имя каталога, в котором размещается файл информационной базы;

Locale

Язык (страна), который будет использован при открытии или создании информационной базы. Допустимые значения такие же как у параметра `<Форматная строка>` метода `Формат()`. Параметр `Locale` задавать не обязательно. Если не задан, то будут использованы региональные установки текущей информационной базы.

7.5.3. Параметры для клиент-серверного варианта информационной базы

Srvr

Имя сервера 1С:Предприятия.

Для обеспечения бесперебойной работы клиентских приложений возможно указание нескольких адресов кластера. Для этого:

- Значением параметра `Srvr` может быть список адресов кластера через запятую.
- В диалоге добавления информационной базы в клиентском приложении значением свойства `Кластер серверов 1С:Предприятия` может быть список адресов кластера через запятую.

Ref

Имя информационной базы на сервере «1С:Предприятия».

DBMS

Тип используемого сервера баз данных:

- `MSSQLServer` – Microsoft SQL Server;
- `PostgreSQL` – PostgreSQL;
- `IBMDB2` – IBM DB2;
- `OracleDatabase` – Oracle Database.

DBSrvr

Имя сервера баз данных.

DB

Имя базы данных в сервере баз данных.

DBUID

Имя пользователя сервера баз данных.

DBPwd

Пароль пользователя сервера баз данных. Если пароль для пользователя сервера баз данных не задан, то данный параметр можно не указывать.

SQLYOfFs

Смещение дат, используемое для хранения дат в Microsoft SQL Server. Может принимать значения 0 или 2000. Данный параметр задавать не обязательно. Если не задан, принимается значение 0.

Locale

Язык (страна), аналогично файловому варианту.

CrSQLDB

Создать базу данных в случае ее отсутствия. Значение параметра:

- **Y** – создавать базу данных в случае отсутствия.
- **N** – не создавать. Значение по умолчанию.

SchJobDn

В созданной информационной базе запретить выполнение регламентных созданий. Значение параметра:

- **Y** – разрешить;
- **N** – запретить. Значение по умолчанию.

SUsr

Имя администратора кластера, в котором должен быть создан начальный образ. Параметр необходимо задавать, если в кластере определены администраторы и для них аутентификация операционной системы не установлена или не подходит;

SPwd

Пароль администратора кластера.

7.6. Параметры командной строки веб-клиента

O=<скорость соединения>

Определяет скорость соединения:

- **Normal** – обычная,
- **Low** – низкая скорость соединения.

C=<строка текста>

Передача параметра в прикладное решение.

N=<имя>

Имя пользователя. Должно быть указано так же, как в списке пользователей, создаваемом в Конфигураторе.

P=<пароль>

Пароль пользователя, имя которого указано в параметре N. Если у пользователя нет пароля, этот параметр можно опустить.

WA<режим>

Определяет режим использования аутентификации операционной системы при запуске «1С:Предприятие». Если параметр **/WA** не указывается, то подразумевается, что используется параметр командной строки **/WA+**.

<режим> может принимать следующие значения:

- **-** – запрет применения аутентификации операционной системы при старте «1С:Предприятия».
- **+** – установка обязательного применения аутентификации операционной системы при старте «1С:Предприятие».

OIDA<режим>

Применение сквозной аутентификации пользователя между разными информационными базами и/или внешними ресурсами для тонкого и веб-клиентов. Если при запуске клиента параметр **/OIDA** не задан, или задан параметр **/OIDA+**, то производится попытка аутентификации через OpenID-провайдера, адрес которого задан в файле [deafault.vrd](#) публикации этой информационной базы.

Если OpenID-провайдер требует интерактивной аутентификации (происходит первое обращение или истекло время жизни признака

аутентифицированности), клиент отображает диалог для ввода имени и пароля пользователя.

Аутентификация происходит по списку пользователей информационной базы OpenID-провайдера.

Аутентифицируемый пользователь информационной базы, использующих OpenID-аутентификацию, должен соответствовать по имени пользователю информационной базы OpenID-провайдера.

<режим> может принимать следующие значения:

- + – использовать OpenID-аутентификацию (по умолчанию).
- - – не использовать OpenID-аутентификацию.

Authoff

Выполняет операцию OpenID logout (завершение сеанса работы пользователя). Завершение сеанса работы выполняется вне зависимости от используемого в дальнейшем метода аутентификации.

L=<код языка>

Указывается код языка интерфейса платформы. Поддерживаемые языки интерфейса (<код языка>) см. [здесь](#).

VL=<код локализации сеанса>

Указывается код локализации сеанса, используемый при форматировании данных типа **Число** и **Дата**, а также в методах **ЧислоПрописью()** и **ПредставлениеПериода()**.

DisableStartupMessages

Поддавляет стартовое сообщение **Конфигурация базы данных не соответствует сохраненной конфигурации. Продолжить?**

DisplayAllFunctions

Включает команду главного меню **Все функции**

DisplayPerformance

Включает отображение количества вызовов сервера и объем данных, отправляемых на сервер и принимаемых с сервера.

DebuggerURL <URL отладчика>

Указание отладчика, к которому приложение сразу после запуска должно подключиться. Указывается URL отладчика (протокол, компьютер и номер порта), на котором в отладчике можно создавать удаленные объекты.

TestClientID<Идентификатор>

Запуска веб-клиента в режиме клиента тестирования. Указанный идентификатор позволяет менеджеру тестирования различать несколько веб-клиентов. Если значение идентификатора не указано или запущено несколько клиентов с одним и тем же значением, то выбирается произвольный.

UsePrivilegedMode

Запуск веб-клиента в привилегированном режиме. Разрешен аутентифицированному пользователю, имеющему административные права. Журнал регистрации фиксирует установку или отказ в возможности установки режима привилегированного сеанса.

Z=<Общий реквизит 1>, <Общий реквизит 2>, ..., <Общий реквизит N>

Установка разделителей при запуске клиентского приложения.

Подробное описание приведено в книге «1С:Предприятие 8.3. Руководство разработчика».

itdi

Запуск в режиме интерфейса с использованием закладок.

isdi

Запуск в режиме интерфейса с использованием отдельных окон.

iTaxi

Запуск в режиме интерфейса **Такси**.

SYSTEMWEBCLIENTSTAT

Включение механизма, позволяющего накапливать статистику использования веб-клиента. Механизм предназначен для использования специалистами фирмы «1С».

Приложение 8. Используемые компоненты и материалы

В программном продукте были использованы следующие компоненты:

- Словари, используемые в полнотекстовом поиске, основаны на словарных базах и словарях тезауруса русского, украинского и английского языков, предоставленных компанией «Информатик».
- Перевод интерфейсов выполнен:
 - Румынский – «Contabilizare-Prof S.R.L.», «SkySoft, S.R.L.».
 - Латышский – ООО «АНДИ М».
 - Литовский – АОЗТ «АВАКОМПА».
 - Болгарский – «DAVID Holding Inc.».
 - Казахский – ООО «Зерде».
 - Грузинский – ООО «Интегрированные Бизнес Решения» (IBS).
 - Вьетнамский – «1 VS» JSC.
 - Турецкий – 1T irketi.

- zlib general purpose compression library

version 1.2.3. July 18, 2005

Copyright © 1995–2005 Jean-loup Gailly and Mark Adler

- Portions of this software are based in part on the work of the Independent JPEG Group
- PNG reference library.

libpng version 1.2.19 – August 18, 2007

Copyright © 1998–2007 Glenn Randers-Pehrson

Copyright © 1996–1997 Andreas Dilger

Copyright © 1995–1996 Guy Eric Schalnat, Group 42, Inc.

- TIFF Software Distribution.

Copyright © 1988–1997 Sam Leffler

Copyright © 1991–1997 Silicon Graphics, Inc.

- Various 1C products provide read/write capability and/or other LZW capability covered by Unisys-owned U.S. patent 4,558,302. Licensing information can be obtained by contacting Unisys at the following address:

Unisys Corporation

Welch Licensing Dept. – MSC1SW19

Township Line and Union Meeting Roads

P.O. Box 500

Fax: (215) 986-3090

- PROJ 4 release 4.4

Copyright © 2000, Frank Warmerdam

- Copyright © 1990–2004 Info-ZIP. All rights reserved
- Dr. Gladman's AES library: (A Password Based File Encryption Example with AES and HMAC-SHA1).

Copyright © 2002, Dr Brian Gladman, Worcester, UK. All rights reserved.

- This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).
- This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

- FreeType version 2.1.9

Copyright © 1996–2002, 2006 David Turner, Robert Wilhelm and Werner Lembers.

- ICU 4.6.

Copyright © 1995-2011 International Business Machines Corporation and others.

- ImageMagick version 6.3.2.

Copyright © 1999–2007 ImageMagick Studio LLC.

- libgsf version 1.10.1.

Copyright © 1994, 1995, 1996, 1999, 2000, 2001, 2002, 2004, 2005 Free Software Foundation, Inc.

- STLport-5.1

Copyright © 1994 Hewlett-Packard Company

Copyright © 1996–1999 Silicon Graphics Computer Systems, Inc.

Copyright © 1997 Moscow Center for SPARC Technology

Copyright © 1999–2003 Boris Fomitchev.

- libxml2 2.9.1

Copyright © 1998–2003 Daniel Veillard. All Rights Reserved

- libxslt 1.1.28

Licence for libxslt except libexslt Copyright © 2001–2002 Daniel Veillard. All Rights Reserved.

Licence for libexslt Copyright © 2001–2002 Thomas Broyer, Charlie Bozeman and Daniel Veillard. All Rights Reserved.

- curl version 7.32.0 (12 August 2013)

Copyright © 1996–2013, Daniel Stenberg, [<daniel@haxx.se>](mailto:daniel@haxx.se).

- UW IMAP toolkit

version imap-2007f (22 July 2011).

Copyright (c) 1988-2008 University of Washington.

- wxWidgets 2.9.2 Copyright (c) 1998-2005 Julian Smart, Robert Roebing et al

- Google Closure Library

Copyright 2006 The Closure Library Authors. All Rights Reserved.

- Closure Stylesheets

Copyright 2008 Google Inc.

- Closure Compiler

Copyright 2014 The Closure Compiler Authors.

- Google Closure Templates

Copyright 2009 Google Inc.

- SQLite database engine

Copyright © 2004–2013 D. Richard Hipp and team.

- Mimetic 0.9.7 Copyright (c) 2009 Stefano Barbato

The MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the «Software»), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

- libEtPan! – a mail library

Copyright (C) 2001, 2005 – DINH Viet Hoa

All rights reserved.

Copyright (C) 2007 g10 Code GmbH

All rights reserved.

Copyright (C) 2006 Andrej Kacian <andrej@kacian.sk>

All rights reserved.

Copyright (c) 1983, 1990, 1993

The Regents of the University of California. All rights reserved.

1С:Предприятие 8.3.6

Руководство администратора

Клиент-серверный вариант

83.005.09 от 12-02-2015

Клиент-серверный вариант. Руководство администратора (02)

Введение (06)

Глава 1. Требования к аппаратуре и программному обеспечению (07)

Глава 2. Клиент-серверный вариант работы (44)

Глава 3. Установка компонентов системы (14)

Глава 4. Запуск компонентов системы (07)

Глава 5. Администрирование (38)

Глава 6. Удаление системы (01)

Приложение 1. Формат строки описания лицензии (02)

Клиент-серверный вариант. Руководство администратора

ПРАВО ТИРАЖИРОВАНИЯ
ПРОГРАММНЫХ СРЕДСТВ И ДОКУМЕНТАЦИИ
ПРИНАДЛЕЖИТ ФИРМЕ «1С»

Приобретая систему «1С:Предприятие», вы тем самым даете согласие не допускать копирования программ и документации без письменного разрешения фирмы «1С».

© ООО «1С», 1996 – 2015

Фирма «1С», Москва, 123056, а/я 64

Отдел продаж: Селезневская ул., 21,

телефон: +7 (495) 737-92-57,

факс: +7 (495) 681-44-07,

e-mail: 1c@1c.ru

URL: <http://www.1c.ru>, <http://www.v8.1c.ru>, <http://users.v8.1c.ru>

Группа разработки программ – А. Алексеев, Р. Алейников, А. Безбородов, Д. Бескоровайнов, А. Бушнев, П. Василец, А. Виноградов, Я. Вирковский, А. Волков, И. Гольштейн, Е. Горностаев, И. Гусаров, Г. Дамье, А. Даровских, О. Дерут, Н. Евграфов, Б. Евтифеев, Д. Зарецкий, Д. Ивашов, И. Коваленко, С. Копиенко, Н. Корсаков, С. Кравченко, В. Кудрявцев, П. Кукушкин, А. Кунченко, А. Лакутин, М. Лейбович, Г. Леонтьев, А. Лехан, А. Макеев, А. Малышенок, А. Медведев, Е. Митрошкин, С. Мурзин, С. Нуралиев, Л. Онучин, М. Отставнов, Д. Павленко, А. Плякин, П. Романов, А. Рукин, Д. Русанов, М. Саблин, Е. Силин, Д. Службин, А. Смирнов, В. Соколов, П. Солодкий, А. Соляник, В. Сосновский, Д. Сысоенков, Д. Тишков, А. Топорков, В. Тунегов, А. Трубкин, А. Тюшкин, В. Филиппов, А. Хасанов, А. Цилябин, В. Ческис, В. Черемисинов, П. Чиков, А. Чичерин, А. Шевченко, А. Щербинин, В. Шульга.

Документация – В. Байдаков, В. Дранищев, Е. Королькова, А. Краюшкин, И. Кузнецов, М. Лавров, А. Моничев, А. Плякин, М. Радченко.

Группа консультационной поддержки – О. Акулова, С. Алексеева, О. Багрова, О. Баклушина, Э. Гарифуллина, В. Давыдова, О. Дмитренко, Л. Ермакова, М. Ершова, Ю. Жестков, О. Завальская, Н. Заявлиная, М. Звонилов, М. Иванова, Г. Коробка, Ю. Лаврова, С. Лепешкина, С. Мазурин, С. Марков, Ю. Мисан, А. Павликов, И. Панин, О. Пехтерева, С. Постнова, А. Прокуровский, Е. Романова, Г. Степаненко, Н. Степанов, Т. Токарева, Е. Широкова.

Группа тестирования – С. Баталин, А. Беляков, Б. Зиатдинов, А. Лапин, Е. Медведев, С. Потапкин.

Корректурa – Е. Семененко.

Верстка – О. Шестакова.

Наименование книги:	1С:Предприятие 8.3. Клиент-серверный вариант. Руководство администратора.
Номер издания:	83.005.09
Дата выхода:	12 февраля 2015 года

ЛИНИЯ КОНСУЛЬТАЦИЙ

Для пользователей системы «1С:Предприятие 8» услуги линии консультаций предоставляются по линии информационно-технологического сопровождения (ИТС).

Получение консультаций возможно только после регистрации программного продукта (для чего необходимо заполнить регистрационную анкету и выслать ее в фирму «1С») и оформления подписки на ИТС.

В стоимость комплекта поставки программного продукта включено обслуживание по линии ИТС в течение периода, указанного в вашей регистрационной анкете. Купон на бесплатное обслуживание по линии ИТС, а также конверт с маркой и адресом входят в состав приобретенного вами комплекта программ.

По истечении периода бесплатного обслуживания получение консультаций возможно только по договору на ИТС. Для заключения договора обращайтесь к партнерам фирмы «1С».

Телефон и адрес электронной почты для консультаций указаны на диске ИТС.

Линия консультаций работает с 9:30 до 17:30 по московскому времени, кроме суббот, воскресений и праздничных дней (по распорядку московских предприятий).

Обращаясь к линии консультаций, следует находиться рядом со своим компьютером, иметь под рукой настоящее руководство и свою половину регистрационной анкеты. Желательно заранее уточнить типы используемого компьютера и принтера.

Набрав телефон линии консультаций, вы должны услышать ответ дежурного сотрудника. После этого сообщите наименование вашей организации, номер вашего экземпляра системы (он обозначен на CD-ROM комплекта поставки и на вашей половине регистрационной анкеты) и, возможно, другую регистрационную информацию по запросу сотрудника линии консультаций. Названная вами информация будет проверена по данным, указанным в отосланной в фирму «1С» половине регистрационной анкеты.

Отвечая на возникшие у вас вопросы, дежурный сотрудник может воспроизвести возникшую ситуацию на своем компьютере. Он даст вам рекомендации сразу же или после обсуждения с разработчиками. Вам не нужно просить к телефону конкретных специалистов: мы отвечаем за работу всего персонала. Работа линии консультаций регистрируется, поэтому при повторных обращениях по сложным проблемам вы можете сослаться на дату и время предыдущего разговора.

МЫ ВСЕГДА РАДЫ ВАМ ПОМОЧЬ!

Введение

Настоящая книга является руководством по администрированию клиент-серверного варианта системы «1С:Предприятие».

Структура Руководства

[Глава 1](#) содержит требования к аппаратуре и программному обеспечению для установки и работы сервера «1С:Предприятия».

[Глава 2](#) посвящена особенностям клиент-серверного варианта работы.

[Глава 3](#) посвящена описанию процесса установки сервера «1С:Предприятия».

[Глава 4](#) описывает запуск сервера «1С:Предприятия».

[Глава 5](#) посвящена особенностям администрирования клиент-серверного варианта системы «1С:Предприятие».

[Глава 6](#) описывает удаление системы.

ВНИМАНИЕ! Описание общих аспектов администрирования системы «1С:Предприятие» приводится в книге «1С:Предприятие 8.3. Руководство администратора», входящей в комплект базовых поставок системы «1С:Предприятие».

Что вы должны знать

Характер изложения материала данного Руководства предполагает, что вы знакомы с операционной системой компьютера, на котором работает система «1С:Предприятие» (ОС семейства Microsoft Windows и ОС семейства Linux, подробнее перечень поддерживаемых ОС приведен на сайте «1С:Предприятия»: <http://v8.1c.ru/requirements/>), и владеете базовыми навыками работы в ней.

Кроме того, необходимо обладать навыками администрирования используемой операционной системы.

При выполнении некоторых процедур администрирования могут потребоваться административные права доступа, а также дистрибутив используемой операционной системы.

Книги документации

В состав документации входят книги по технологической платформе «1С:Предприятие»:

- «1С:Предприятие 8.3. Руководство пользователя». Книга содержит описание общих приемов работы с программными продуктами, созданными на базе платформы «1С:Предприятие», и может не поставляться отдельно. Данная книга не поставляется в составе продукта «1С:Предприятие 8.3. Версия для обучения программированию».
- «1С:Предприятие 8.3. Руководство разработчика». Книга необходима для изменения и настройки конфигурации под особенности учета конкретной организации, а также для разработки новых конфигураций.
- «1С:Предприятие 8.3. Руководство администратора». Книга описывает администрирование системы «1С:Предприятие», включая информацию об особенностях построения клиент-серверных систем. Данная книга не поставляется в составе продукта «1С:Предприятие 8.3. Версия для обучения программированию».
- «1С:Предприятие 8.3. Клиент-серверный вариант. Руководство администратора». Книга описывает особенности работы системы «1С:Предприятие» с информационными базами в варианте клиент-сервер в части установки и эксплуатации. Данная книга не поставляется в составе продукта «1С:Предприятие 8.3. Версия для обучения программированию».

- Синтаксис встроенного языка и языка запросов представлен в книге «1С:Предприятие 8.3. Руководство разработчика». Описание объектной модели полностью включено в поставку в электронном виде (в разделах справки конфигуратора и синтакс-помощнике). Описание объектной модели также содержится в книге «1С:Предприятие 8.3. Описание встроенного языка», которая распространяется отдельно.

ВНИМАНИЕ! Комплект поставки конкретного продукта может включать лишь некоторые из перечисленных книг документации.

Методические материалы и дополнительные возможности

Фирма «1С» осуществляет методическую поддержку освоения и внедрения системы программ «1С:Предприятие». Методическая поддержка включает в себя разнообразные формы предоставления информации, необходимой для грамотной и эффективной разработки и использования прикладных решений.

Сопроводительные файлы комплекта поставки «1С:Предприятия»

В процессе установки платформы «1С:Предприятие» выполняется копирование на жесткий диск ряда сопроводительных файлов, содержащих описание изменений, реализованных в данной версии платформы, и инструкции по переходу с предыдущих версий.

Все сопроводительные файлы располагаются в каталоге установочных файлов конкретной версии системы «1С:Предприятие», в каталоге `\docs\ru`. Если при установке системы использовался каталог, предложенный по умолчанию, то эти файлы будут располагаться в каталоге `C:\Program Files\1cv8\НомерВерсии\docs\ru`. Здесь `НомерВерсии` означает номер установленной версии. Так, для версии 8.3.3.100 каталог будет иметь следующий вид: `C:\Program Files\1cv8\8.3.3.100\docs\ru`.

- [V8Update.htm](#) – в этом файле содержатся отличия текущей версии платформы от предыдущих версий и особенности перехода на новую версию.

ИТС – информационно-технологическое сопровождение

Фирма «1С» осуществляет платную методическую поддержку пользователей в рамках информационно-технологического сопровождения (ИТС) программ системы «1С:Предприятие».

Ежемесячные выпуски ИТС содержат большое количество постоянно обновляемой информации, позволяющей более эффективно использовать продукты системы «1С:Предприятие». Отметим наиболее важные составляющие ИТС:

- обновления технологической платформы «1С:Предприятие» и прикладных решений;
- методические материалы по технологической платформе «1С:Предприятие»;
- методические материалы по типовым прикладным решениям фирмы «1С»;
- конфигурация «Конвертация данных» для настройки правил обмена между информационными базами «1С:Предприятия», имеющими различную конфигурацию;
- система стандартов и методик разработки конфигураций для платформы «1С:Предприятие» (предназначена для ознакомления партнеров и пользователей фирмы «1С» с техническими и проектными решениями, используемыми при разработке типовых конфигураций на платформе «1С:Предприятие»);
- советы линии консультаций, помогающие пользователям «1С:Предприятия» получить ответы на наиболее часто встречающиеся вопросы и избежать типичных ошибок;
- информация по обучению работе с платформой «1С:Предприятие» и прикладными решениями фирмы «1С»;
- демонстрационные ролики программных продуктов, позволяющие получить первое представление о возможностях прикладных решений;

- справочники по заполнению деклараций по налогам (налог на прибыль, НДС, налог на имущество, ЕСН, взносы в ПФР);
- обширная подборка бухгалтерской периодики, включая текущие выпуски журналов и архивы;
- база аналитических обзоров законодательства и арбитражной практики;
- рекомендации по составлению квартальной и годовой отчетности в «1С:Бухгалтерии 8»;
- справочник типовых хозяйственных операций хозрасчетного предприятия;
- справочник по оформлению расчетов с персоналом по оплате труда и правовым аспектам трудовых отношений;
- справочная правовая система «Гарант» – полный набор нормативных документов законодательства Российской Федерации, в том числе по бухгалтерскому учету, налогам и предпринимательству.

Начиная с 2005 года, стандартный сервис по поддержке пользователей – подписчиков ИТС включает доступ к сайту поддержки пользователей системы «1С:Предприятие».

Более подробно о проекте ИТС можно прочитать на сайте фирмы «1С»:

<http://www.1c.ru/rus/support/its/its.htm>.

Оставить заявку на демонстрацию дисков и приобретение подписки на ИТС можно на сайте фирмы «1С»:

<http://www.1c.ru/rus/support/its/zajavka.jsp>.

Оформить подписку на ИТС можно у партнеров фирмы «1С». Список партнеров, имеющих опыт массового обслуживания пользователей в рамках проектов ИТС, опубликован на сайте фирмы «1С»:

<http://www.1c.ru/rus/partners/service.jsp>.

Информация по «1С:Предприятию 8»

Адрес сайта: <http://v8.1c.ru/AllInfo>.

Данный сайт представляет собой набор ссылок на часто используемую информацию для следующих категорий пользователей:

- пользователи прикладных решений,
- разработчики прикладных решений,
- партнеры фирмы «1С».

Сайт системы программ «1С:Предприятие 8»

Адрес сайта: <http://v8.1c.ru>.

Сайт содержит информацию по технологической платформе системы «1С:Предприятие 8» и по типовым прикладным решениям, выпущенным фирмой «1С» на ее основе.

Пользовательский сайт

Адрес сайта: <http://users.v8.1c.ru>.

На сайте поддержки пользователей системы «1С:Предприятие» представлена информация о номерах версий платформы и конфигураций, дате их выхода, выпусках ИТС, на которых опубликовано обновление. По каждой версии представлена следующая информация:

- для технологической платформы:
 - отличия данной версии от предыдущих и особенности перехода;

- ошибки, исправленные при выпуске данной версии;
- дистрибутив обновления;
- файл [readme.htm](#);
- для прикладных решений:
 - новое в релизе;
 - полный список изменений;
 - список основных изменений;
 - порядок обновления;
 - дистрибутив обновления;
 - номер версии платформы, необходимой для использования релиза конфигурации.

Также на пользовательском сайте публикуются рекомендации по администрированию системы «1С:Предприятие».

Основное преимущество, которое дает пользователям «1С:Предприятия» поддержка на данном сайте, – это возможность обновления технологической платформы и прикладных решений через Интернет до получения дисков ИТС.

Также на сайте публикуются дополнительные компоненты, используемые системой «1С:Предприятие» (например, СУБД PostgreSQL), а также тестовые версии платформы и прикладных решений.

ВНИМАНИЕ! К сайту поддержки пользователей имеют право доступа зарегистрированные пользователи программных продуктов системы «1С:Предприятие». При этом пользователи продуктов, для которых предусмотрено обслуживание по линии информационно-технологического сопровождения (ИТС), должны иметь действующую подписку на ИТС.

Зарегистрироваться на сайте поддержки пользователей можно самостоятельно либо обратиться к партнеру фирмы «1С».

Для самостоятельной регистрации необходимо с главной страницы сайта перейти по ссылке [Самостоятельная регистрация пользователей по PIN-коду](#) и выполнить указанные на сайте действия.

В процессе регистрации потребуется PIN-код, входящий в комплект поставки программного продукта.

Если в комплекте поставки отсутствует PIN-код для регистрации на сайте, следует обратиться к партнеру фирмы «1С».

Для регистрации программного продукта необходимо заполнить регистрационную анкету на программный продукт (она является частью регистрационной карточки) и отправить ее в фирму «1С» по почте или факсу.

Для оформления подписки на ИТС можно обратиться к любому из сервис-партнеров фирмы «1С». Список партнеров, имеющих опыт массового обслуживания пользователей в рамках проектов ИТС, опубликован на сайте фирмы «1С»: <http://www.1c.ru/rus/partners/service.jsp>.

Принятые обозначения

Для лучшего понимания излагаемого материала в настоящем Руководстве приняты некоторые общие приемы выделения отдельных элементов текста. Соглашение о таких приемах приведено ниже.

Обозначения клавиш. Клавиши, такие как [Enter](#), [Esc](#), [Del](#) и подобные, будут обозначаться, как показано выше, без кавычек.

Для ссылок на клавиши управления курсором (клавиши со стрелками) будет использоваться фраза [клавиши управления курсором](#), когда необходимо сослаться сразу на все эти клавиши. Если необходимо упомянуть эти клавиши по отдельности, будут использоваться выражения [Стрелка вверх](#), [Стрелка вниз](#), [Стрелка вправо](#) и [Стрелка влево](#).

Комбинации клавиш. Когда для выполнения какой-либо команды необходимо нажать комбинацию из двух клавиш, она дается в виде [Ctrl + F3](#).

Обозначения кнопок. Наименования кнопок в форме будут даваться их названиями без кавычек, например, [ОК](#), [Отмена](#), [Удалить](#) и так далее.

Ключевые слова встроенного языка. Ключевые слова встроенного языка системы «1С:Предприятие» выделяются шрифтом и пишутся так, как в программных модулях: [РабочаяДата](#). В тексте также будут встречаться ссылки на описания разделов или элементов встроенного языка (свойства, методы и т. д.). С данными описаниями можно ознакомиться в справке (ветвь [Встроенный язык](#)).

Описание действия с помощью меню. Для описания выбора пункта меню используется следующая конструкция: [Меню – Подменю – Подменю – ... – Пункт](#). Например: «Для выбора масштаба изображения используется пункт [Таблица – Вид – Масштаб](#), что эквивалентно тексту: «Для выбора масштаба изображения используется пункт [Масштаб](#) подменю [Вид](#) меню [Таблица](#) главного меню программы». Если выбор осуществляется не из главного меню программы, то это указывается дополнительно.

Режимы работы системы «1С:Предприятие». Система «1С:Предприятие» работает в двух режимах: настройки и проверки конфигурации (далее в Руководстве – режим Конфигуратор или конфигурирование), когда в Руководстве описывается работа по созданию или изменению конфигурации) и исполнения конфигурации (далее в Руководстве – режим 1С:Предприятие).

В данном Руководстве пользователем будет называться специалист, выполняющий разработку или сопровождение конфигурации.

Выражение %USERPROFILE% означает переменную окружения ОС Windows, которая содержит путь к каталогу профиля текущего пользователя. В случае стандартной установки (и для пользователя [Ivanov](#)) этот путь выглядит следующим образом:

[Копировать в буфер обмена](#)

C:\Documents and Settings\Ivanov

Для ОС Windows Vista и выше этот путь выглядит следующим образом:

[Копировать в буфер обмена](#)

C:\Users\Ivanov

Выражение %APPDATA% означает переменную окружения ОС Windows, которая содержит путь к каталогу (в профиле пользователя), где приложения хранят свои данные. В случае стандартной установки (и для пользователя [Ivanov](#)) этот путь выглядит следующим образом:

[Копировать в буфер обмена](#)

C:\Documents and Settings\Ivanov\Application Data

Для ОС Windows Vista и выше этот путь выглядит следующим образом:

[Копировать в буфер обмена](#)

C:\Users\Ivanov\AppData\Roaming

Выражение %LOCALAPPDATA% означает переменную окружения ОС Windows Vista и старше, содержащую путь к каталогу (в профиле пользователя), в котором находятся данные приложения, специфичные для пользователя. В случае стандартной установки (и для пользователя [Ivanov](#)) этот путь выглядит следующим образом:

[Копировать в буфер обмена](#)

C:\Users\Ivanov\AppData\Local

Исполняемые файлы, разработанные одновременно для ОС Windows и для ОС Linux, обозначаются одинаково, вне зависимости от используемой ОС, и без указания расширения (если таковое

присутствует). Так, исполняемый файл `ragent.exe` будет упоминаться как `ragent`. При этом для ОС **Windows** необходимо добавить расширение `.exe` (регистр файл не имеет значения), а для ОС **Linux** ничего добавлять не надо (регистр файлов имеет значение).

Глава 1. Требования к аппаратуре и программному обеспечению

Текущий список поддерживаемых дистрибутивов серверных операционных систем и систем управления базами данных опубликован на сайте <http://www.v8.1c.ru/requirements/>.

ПРИМЕЧАНИЕ. Разрядность сервера «1С:Предприятия» не связана с разрядностью СУБД. Совместно могут использоваться серверы (сервер «1С:Предприятия» и СУБД) разной разрядности.

1.1. 32-разрядный сервер

Требования к рабочим серверам, входящим в состав кластера серверов «1С:Предприятия»:

- Операционные системы Windows XP Service Pack 2, Windows Server 2003, Windows Vista, Windows 7, Windows Server 2008 или один из поддерживаемых дистрибутивов Linux.
- Процессор не ниже Pentium/Xeon 2,4 ГГц. Желательно использование многопроцессорных или многоядерных машин, так как наличие нескольких процессоров/ядер благотворно сказывается на пропускной способности кластера серверов «1С:Предприятия», особенно в случае интенсивной работы нескольких пользователей.
- Оперативная память не менее 2 Гбайт. И хотя рабочие процессы кластера серверов «1С:Предприятия» могут исполняться в достаточно небольших объемах памяти, при пиковых нагрузках их потребности могут быть весьма значительными.
- Требуется наличие USB-порта для подключения ключа аппаратной защиты кластера серверов «1С:Предприятия».

Требования к серверу баз данных главным образом определяются требованиями Microsoft SQL Server, PostgreSQL, IBM DB2, Oracle Database. В качестве сервера баз данных может использоваться любой компьютер, на котором может работать Microsoft SQL Server, PostgreSQL, IBM DB2, Oracle Database (см. [здесь](#)).

Требования к серверу баз данных:

- Сервер баз данных:
 - Microsoft SQL Server 2000 Service Pack 2 (рекомендуется Service Pack 4).
 - Microsoft SQL Server 2005 Service Pack 3.
 - Microsoft SQL Server 2008 Service Pack 1.
 - Microsoft SQL Server 2008 R2.
 - Microsoft SQL Server 2012.
 - Microsoft SQL Server 2014.
 - PostgreSQL Windows/Linux:
 - версия 8.1.5,
 - версия 8.2.4,
 - версия 8.3.8,
 - версия 8.4.3,
 - версия 9.0.3,

- версия 9.1.9,
- версия 9.2.4.
- IBM DB2 Windows/Linux 32:
 - версия 9.1.301.537,
 - версия 9.5.200.353,
 - версия 9.7.100.177,
 - версия 10.1.0.872.
- Oracle Database:
 - версия 10g R2 – 10.2.0.4 Windows/Linux.

Требует установки патчей, список которых указан на странице <http://www.v8.1c.ru/requirements/>.

- версия 11g R1 – 11.1.0.7.0 Windows/Linux.

Требует установки патчей, список которых указан на странице <http://www.v8.1c.ru/requirements/>.

- версия 11g R2 – 11.2.0.3.0 Windows , 11.2.0.2.0 Linux.

Требует установки патчей, список которых указан на странице <http://www.v8.1c.ru/requirements/>.

- Операционная система: в соответствии с требованиями Microsoft SQL Server, PostgreSQL, IBM DB2, Oracle Database.
- Компьютер сервера баз данных: технические характеристики компьютера и операционная система должны соответствовать требованиям используемой версии сервера баз данных Microsoft SQL Server, PostgreSQL, IBM DB2, Oracle Database.

1.2. 64-разрядный сервер

Требования к рабочим серверам, входящим в состав кластера серверов «1С:Предприятия»:

- Операционные системы Windows XP Service Pack 2, Windows Server 2003, Windows Vista, Windows 7, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012 для x64 или один из поддерживаемых дистрибутивов Linux для x86-64.
- Процессор с архитектурой x86-64 (Intel с поддержкой Intel 64, AMD с поддержкой AMD64). Желательно использование многопроцессорных или многоядерных машин, так как наличие нескольких процессоров/ядер благотворно сказывается на пропускной способности кластера серверов «1С:Предприятия», особенно в случае интенсивной работы нескольких пользователей.
- Оперативная память не менее 2 Гбайт (рекомендуется 4 Гбайт и выше). И хотя рабочие процессы кластера серверов «1С:Предприятия» могут исполняться в достаточно небольших объемах памяти, в пиковых ситуациях их потребности могут быть весьма значительными.
- Требуется наличие USB-порта для подключения ключа аппаратной защиты кластера серверов «1С:Предприятия».
- Устройство чтения компакт-дисков.

Требования к серверу баз данных главным образом определяются требованиями Microsoft SQL Server, PostgreSQL, IBM DB2, Oracle Database. В качестве сервера баз данных может использоваться любой

компьютер, на котором может работать Microsoft SQL Server, PostgreSQL, IBM DB2, Oracle Database (см. [здесь](#)).

Требования к серверу баз данных:

- сервер баз данных:
 - Microsoft SQL Server 2000 (64-bit) Service Pack 2 (рекомендуется Service Pack 4).
 - Microsoft SQL Server 2005 (64-bit) Service Pack 3.
 - Microsoft SQL Server 2008 (64-bit) Service Pack 1.
 - Microsoft SQL Server 2008 R2.
 - Microsoft SQL Server 2012.
 - Microsoft SQL Server 2014.
 - PostgreSQL Linux:
 - версия 8.1.5,
 - версия 8.2.4,
 - версия 8.3.8,
 - версия 8.4.3,
 - версия 9.0.3,
 - версия 9.1.9,
 - версия 9.2.4.
 - IBM DB2 Windows/Linux 64:
 - версия 9.1.301.537,
 - версия 9.5.200.353,
 - версия 9.7.100.177,
 - версия 10.1.0.872.
 - Oracle Database:
 - версия 10g R2 – 10.2.0.4 Windows/Linux.
Требует установки патчей, список которых указан на странице <http://www.v8.1c.ru/requirements/>.
 - версия 11g R1 – 11.1.0.7.0 Windows/Linux.
Требует установки патчей, список которых указан на странице <http://www.v8.1c.ru/requirements/>.
 - версия 11g R2 – 11.2.0.3.0 Windows, 11.2.0.2.0 Linux.
Требует установки патчей, список которых указан на странице <http://www.v8.1c.ru/requirements/>.

- Операционная система: в соответствии с требованиями Microsoft SQL Server, PostgreSQL, IBM DB2, Oracle Database.
- Компьютер сервера баз данных: технические характеристики компьютера и операционная система должны соответствовать требованиям используемой версии сервера баз данных Microsoft SQL Server, PostgreSQL, IBM DB2, Oracle Database.

1.3. Общие требования для клиент-серверного варианта работы

В случае если кластер серверов «1С:Предприятия» и сервер баз данных установлены на разных компьютерах, на производительность системы может существенно влиять пропускная способность сетевого соединения между компьютерами кластера серверов «1С:Предприятия» и компьютером сервера баз данных. Рекомендуется использовать сетевые карты с пропускной способностью 100 Мбит и выше.

ВАЖНО! Для нормальной работы «1С:Предприятия» в варианте клиент-сервер необходимо на клиентских компьютерах отключить использование энергосберегающих режимов [Sleep](#), [Standby](#) и [Hibernate](#).

Для использования некоторых возможностей сервера, работающего под управлением ОС Linux, может потребоваться наличие следующих библиотек:

- [ImageMagick](#):
 - Имя загружаемой библиотеки: [libMagickWand](#), [libWand](#) или [libMagickWand-6.Q16](#).
 - Используемая версия:
 - версии [6.2.8](#) и выше, если не планируется работать с географической схемой;
 - версии [6.3.2](#) и выше, если планируется работать с географической схемой.
 - Назначение:
 - Для работы сервера «1С:Предприятия» в управляемом режиме;
 - При использовании на сервере объектов [Диаграмма](#), [ГрафическаяСхема](#), [ТабличныйДокумент](#);
 - При использовании на сервере метода [ПолучитьКартинку\(\)](#) объектов [Диаграмма](#), [ДиаграммаГанта](#), [Дендрограмма](#), [СводнаяДиаграмма](#).
- [FreeType](#):
 - Имя загружаемой библиотеки: [libfreetype](#).
 - Используемая версия: [2.1.9](#) и выше.
 - Назначение:
 - Для работы сервера «1С:Предприятия» в управляемом режиме;
 - При использовании на сервере объектов [Диаграмма](#), [ГрафическаяСхема](#), [ТабличныйДокумент](#);
 - при использовании метода [ПолучитьКартинку\(\)](#) объектов [Диаграмма](#), [ДиаграммаГанта](#), [Дендрограмма](#), [СводнаяДиаграмма](#);
 - При выполнении операции сохранения в PDF.
- [Libgsf](#):
 - Имя загружаемой библиотеки: [libgsf-1](#).

- Используемая версия: 1.10.1 и выше.
- Назначение: экспорт/импорт документов в формате XLS.
- **Glib**;
- Имя загружаемой библиотеки: [libglib-2.0](#).
- Используемая версия: 2.12.4 и выше.
- Назначение: экспорт/импорт документов в формате XLS.
- **unixOdbc**:
- Имя загружаемой библиотеки: [libodbc](#).
- Используемая версия: 2.2.11 и выше.
- Назначение: работа с внешними источниками данных.
- **Kerberos**:
- Имя загружаемой библиотеки: [libkrb5](#).
- Используемая версия: 1.4.2 и выше.
- Назначение: Аутентификация средствами ОС.
- **GSS-API Kerberos**:
- Имя загружаемой библиотеки: [libgssapi_krb5](#).
- Используемая версия: 1.4.2 и выше.
- Назначение: Аутентификация средствами ОС.
- **Microsoft Core Fonts**.

«1С:Предприятие» загружает библиотеку, указывая имя в виде [имя загружаемой библиотеки.so.X.Y](#), где:

- [имя загружаемой библиотеки](#) – значение из вышеприведенного списка;
- [so](#) – признак файл с библиотекой;
- [X.Y](#) – цифры суффикса присутствующей в системе библиотеки.

Используются только библиотеки, зарегистрированные в системном кеше динамического линковщика времени выполнения (данную информацию можно получить, запустив команду [ldconfig -p](#)). Если в системе имеется несколько версий одной библиотеки, будет загружена самая старшая версия.

Подробную информацию о поддерживаемых СУБД (разрядность серверов, используемые ОС, номера версий) можно получить по адресу <http://v8.1c.ru/requirements/>.

1.4. Использование энергосберегающих режимов работы компьютера

Если на компьютере запущено «1С:Предприятие», то использование энергосберегающих режимов будет возможно только при соблюдении всех нижеперечисленных условий:

- ключ защиты установлен локально,

- используется файл-серверный вариант,
- файл базы данных расположен на локальном диске.

В противном случае переход в энергосберегающий режим будет запрещен независимо от вызвавшей его причины.

1.5. Поддерживаемые СУБД

Подробную информацию о текущем списке поддерживаемых СУБД (разрядность серверов, используемые ОС, номера версий) можно получить по адресу: <http://v8.1c.ru/requirements/>.

Система «1С:Предприятие» поддерживает следующие СУБД:

- **Microsoft SQL Server.**

Документация на SQL-сервер:

- Версия Microsoft SQL Server 2000 (на английском языке):
<http://msdn.microsoft.com/ru-ru/library/aa257103>.
- Версия Microsoft SQL Server 2005:
<http://msdn.microsoft.com/ru-ru/library/bb418498.aspx>.
- Версия Microsoft SQL Server 2008:
<http://msdn.microsoft.com/ru-ru/library/bb418491.aspx>.
- Версия Microsoft SQL Server 2008 R2:
<http://msdn.microsoft.com/ru-ru/library/bb418470.aspx>.
- Версия Microsoft SQL Server 2012:
<http://msdn.microsoft.com/ru-ru/library/bb418433.aspx>.
- Версия Microsoft SQL Server 2014:
<http://msdn.microsoft.com/ru-ru/library/dd631854.aspx>.

- **PostgreSQL.**

Поддерживаемый дистрибутив можно получить по адресу: <http://users.v8.1c.ru/AddComp.aspx> (необходим доступ к разделу технической поддержки).

Документация на SQL-сервер:

- Версия PostgreSQL 8.1 (на английском языке):
<http://www.postgresql.org/docs/8.1/static/index.html>;
- Версия PostgreSQL 8.2 (на английском языке):
<http://www.postgresql.org/docs/8.2/static/index.html>;
- Версия PostgreSQL 8.3 (на английском языке):
<http://www.postgresql.org/docs/8.3/static/index.html>;
- Версия PostgreSQL 8.4 (на английском языке):
<http://www.postgresql.org/docs/8.4/static/index.html>.
- Версия PostgreSQL 9.0 (на английском языке):
<http://www.postgresql.org/docs/9.0/static/index.html>.
- Версия PostgreSQL 9.1 (на английском языке):
<http://www.postgresql.org/docs/9.1/static/index.html>.
- Версия PostgreSQL 9.2 (на английском языке):

<http://www.postgresql.org/docs/9.2/static/index.html>.

- **IBM DB2.**

Поддерживаемый дистрибутив можно получить по адресу: <http://users.v8.1c.ru/AddComp.aspx> (необходим доступ к разделу технической поддержки).

Документация на SQL-сервер:

- Версия IBM DB2 v9.1 (на английском языке):
<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp>;
- Версия IBM DB2 v9.5:
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp>;
- Версия IBM DB2 v9.7:
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp>.
- Версия IBM DB2 v10.1 (на английском языке):
<http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/index.jsp>.

- **Oracle Database.**

Документация на SQL-сервер:

- Версия Oracle Database 10g Release 2:
 - <http://www.oracle.com/pls/db102/homepage> – просмотр (на английском языке);
 - http://download.oracle.com/docs/cds/B19306_01.zip – скачать (на английском языке).
- Версия Oracle Database 11g Release 1:
 - <http://www.oracle.com/pls/db111/homepage> – просмотр (на английском языке);
 - http://download.oracle.com/docs/cds/B28359_01.zip – скачать (на английском языке).
- Версия Oracle Database 11g Release 2:
 - <http://www.oracle.com/pls/db112/homepage> – просмотр (на английском языке);
 - http://download.oracle.com/docs/cds/E11882_01.zip – скачать (на английском языке).

1.6. Прочие требования

Пользователь, от чьего имени выполняется сервер, должен иметь:

- право **Список содержимого папки** на каталог временных файлов в ОС Windows;
- право **Чтение** на каталог временных файлов в ОС Linux.

Глава 2. Клиент-серверный вариант работы

1.1. Общее устройство системы

«1С:Предприятие» позволяет работать с информационными базами в варианте клиент-сервер. В случае «1С:Предприятия» под вариантом клиент-сервер понимается архитектура, подразумевающая наличие следующих программных уровней:

- один из видов клиентского приложения «1С:Предприятия» (обычный клиент, тонкий клиент или веб-клиент);
- веб-сервер (только для веб-клиента и для тонкого клиента, подключенного через веб-сервер);
- кластер серверов «1С:Предприятия»;
- сервер баз данных.

На [рис. 1](#) показана схема взаимодействия элементов системы.

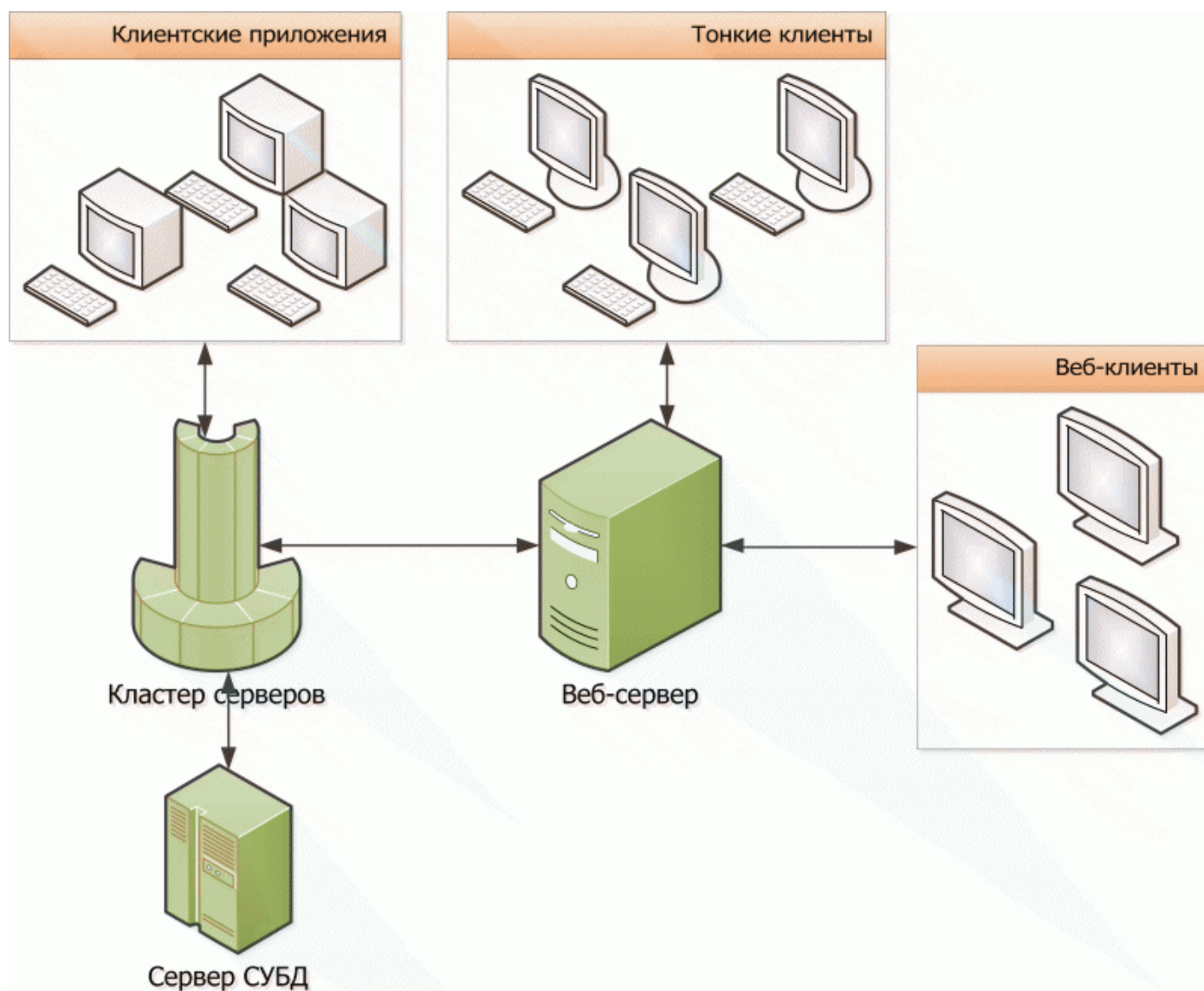


Рис. 1. Схема взаимодействия элементов системы

Клиентские приложения, тонкие клиенты и веб-клиенты – это и есть «1С:Предприятие» (в различных режимах запуска), с которым работает конечный пользователь. Для работы веб-клиента требуется только веб-браузер.

Кластер серверов «1С:Предприятия» является логическим понятием и представляет собой совокупность рабочих серверов, функционирующих на одном или нескольких компьютерах, и списка информационных баз, которые размещены в этом кластере.

Кластер серверов «1С:Предприятия» образует промежуточный программный слой между клиентским приложением и сервером баз данных. Клиентские приложения не имеют непосредственного доступа к серверу баз данных. Для доступа к информационной базе клиентское приложение взаимодействует с кластером серверов «1С:Предприятия».

Архитектура системы ориентирована на максимальный перенос выполнения всей функциональности на кластер серверов и максимальное «облегчение» клиента. В кластере серверов выполняется вся работа прикладных объектов, выполняется подготовка к отображению форм (чтение объектов из информационной базы и заполнение данных формы, расположение элементов, запись данных формы после изменения) и командного интерфейса, формируются отчеты. На клиенте выполняется только отображение информации, подготовленной в кластере серверов, выполняется взаимодействие с пользователем и вызовы серверных методов для выполнения необходимых действий.

Кроме того, на серверах, входящих в кластер серверов «1С:Предприятия», хранятся файлы, содержащие журналы регистрации информационных баз, зарегистрированных на данном сервере «1С:Предприятия», и другие служебные файлы. Все эти данные не являются жизненно необходимыми для работы с информационными базами, и их потеря не приведет к неработоспособности информационных баз. Также на серверах, входящих в кластер, выполняются фоновые задания.

Установка кластера серверов «1С:Предприятия» выполняется программой установки «1С:Предприятия». Настройка кластера серверов выполняется с помощью утилиты администрирования кластера серверов, входящей в комплект поставки.

Ключ аппаратной защиты кластера серверов «1С:Предприятия» не является сетевым, поэтому он должен быть подключен к каждому компьютеру, на котором функционируют рабочие процессы кластера.

Веб-сервер необходим для работы веб-клиента и одного из вариантов работы тонкого клиента. Фактически в случае работы через веб-сервер с кластером серверов взаимодействует именно веб-сервер. А уже веб-сервер взаимодействует с тонким и веб-клиентами.

ПРИМЕЧАНИЕ. Если явно не оговорено обратное, далее под термином «клиентское приложение» понимается обычный клиент, тонкий клиент или веб-сервер.

Сервер баз данных. Хранение жизненно важных данных информационных баз «1С:Предприятия» в варианте клиент-сервер обеспечивается сервером баз данных. В качестве сервера баз данных в «1С:Предприятии» могут использоваться различные СУБД (см. [здесь](#)). При этом каждая информационная база целиком сохраняется в отдельной базе данных используемой СУБД.

2.1. Устройство кластера серверов

2.1.1. Общие понятия

Основной единицей кластера серверов выступает **рабочий сервер**. Рабочий сервер – это компьютер, на котором выполняется **агент сервера (ragent)**. Агент сервера «представляет» рабочий сервер в кластере серверов. Как правило, на одном компьютере располагается один рабочий сервер, однако в некоторых случаях (например, для целей отладки) возможна работа на одном физическом компьютере нескольких рабочих серверов. Рабочие серверы, расположенные на одном компьютере, должны иметь разные номера сетевых портов, идентифицирующих рабочий сервер, и работать с разными каталогами данных кластера.

СОВЕТ. Не рекомендуется размещать несколько рабочих серверов на одном физическом компьютере для систем, находящихся в промышленной эксплуатации.

Кластер серверов представляет собой один или несколько рабочих серверов. При этом информация о том, сколько рабочих серверов выполняется на одном физическом компьютере, не важна для описания и работы кластера. Для того чтобы «знать», для каких кластеров серверов данный рабочий сервер является центральным, агент сервера ведет **список кластеров**. Список кластеров представляет собой файл [1cv8wsrv.lst](#), в котором хранится информация о тех кластерах серверов, в состав которых входит данный рабочий сервер, а также список администраторов данного рабочего сервера. Данный файл хранится в каталоге данных кластера (см. [здесь](#)). Фактически, агент сервера не входит в состав рабочего сервера, а лишь обеспечивает его работу (в том числе и «представительство» в кластере серверов).

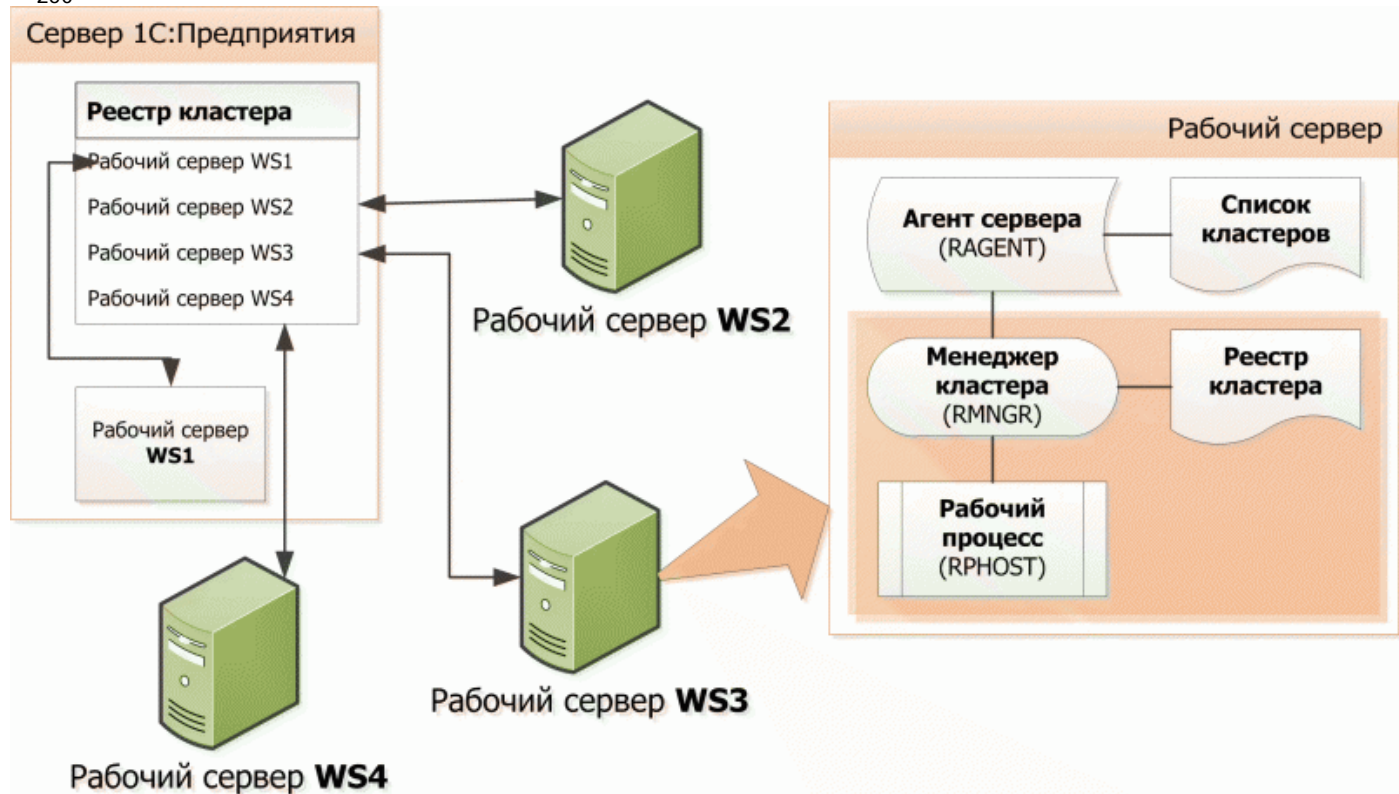


Рис. 2. Общая схема сервера «1С:Предприятия»

В состав любого кластера должен входить минимум один рабочий сервер, у которого установлено свойство [Центральный сервер](#). Максимальное количество центральных серверов не ограничено. Это означает, что для всех рабочих серверов, входящих в состав кластера серверов, можно установить флажок [Центральный сервер](#). Рабочий сервер может быть центральным сервером в одном кластере и обычным (не центральным) – в другом кластере серверов. Также, рабочий сервер с установленным признаком [Центральный сервер](#), может выступать в качестве точки подключения к кластеру серверов, в состав которого он (рабочий сервер) входит.

Рабочая часть рабочего сервера включает в себя **менеджер кластера (rmngr)** и **рабочий процесс (rphost)**. Менеджер кластера обеспечивает функционирование рабочего сервера и взаимодействие с другими рабочими серверами, входящими в состав кластера. Рабочий процесс непосредственно обслуживает клиентские приложения, взаимодействует с сервером баз данных, исполняет код, который в прикладном решении отмечен как выполняемый на сервере. Количество рабочих процессов определяется настройками рабочего сервера, кластера серверов и физическими характеристиками компьютера, на котором работает рабочий сервер.

В состав рабочего сервера может входить минимум один менеджер кластера. Если менеджер кластера работает на центральном сервере, то он называется **главным менеджером кластера**. Максимальное количество менеджеров кластера равно количеству сервисов кластера (см. [здесь](#)). Однако если один рабочий сервер входит в состав нескольких кластеров, то для каждого кластера будет создан минимум один менеджер кластера.

Главный менеджер кластера «ведет» **реестр кластера**. Реестр кластера представляет собой файл `1CV8Clst.lst`, который хранит следующую информацию:

- список информационных баз, зарегистрированных в данном кластере;
- список рабочих серверов, входящих в кластер;
- список рабочих процессов, входящих в кластер;
- список менеджеров кластера;
- список сервисов кластера;
- список администраторов кластера.

Если в состав кластера входит несколько центральных серверов, то реестр кластера ведется каждым из главных менеджеров кластера. Для того чтобы данные каждой копии реестра кластера были актуальными, внутри кластера серверов выполняется постоянная синхронизация реестра кластера между главными менеджерами кластера центральных серверов.

В самом простом случае, рабочий сервер и кластер серверов могут располагаться на одном компьютере, как показано на [рис. 3](#).

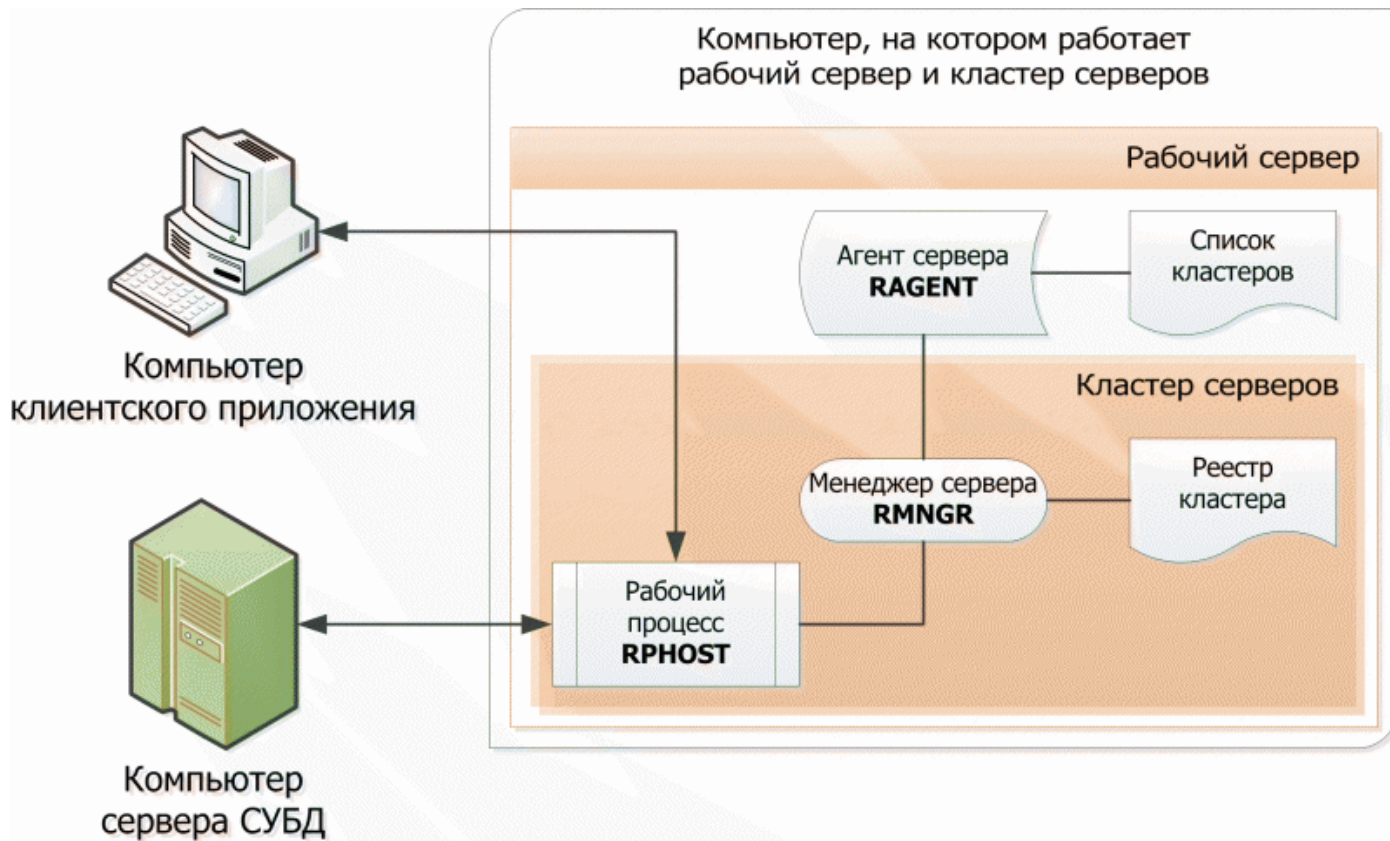


Рис. 3. Простой вариант кластера

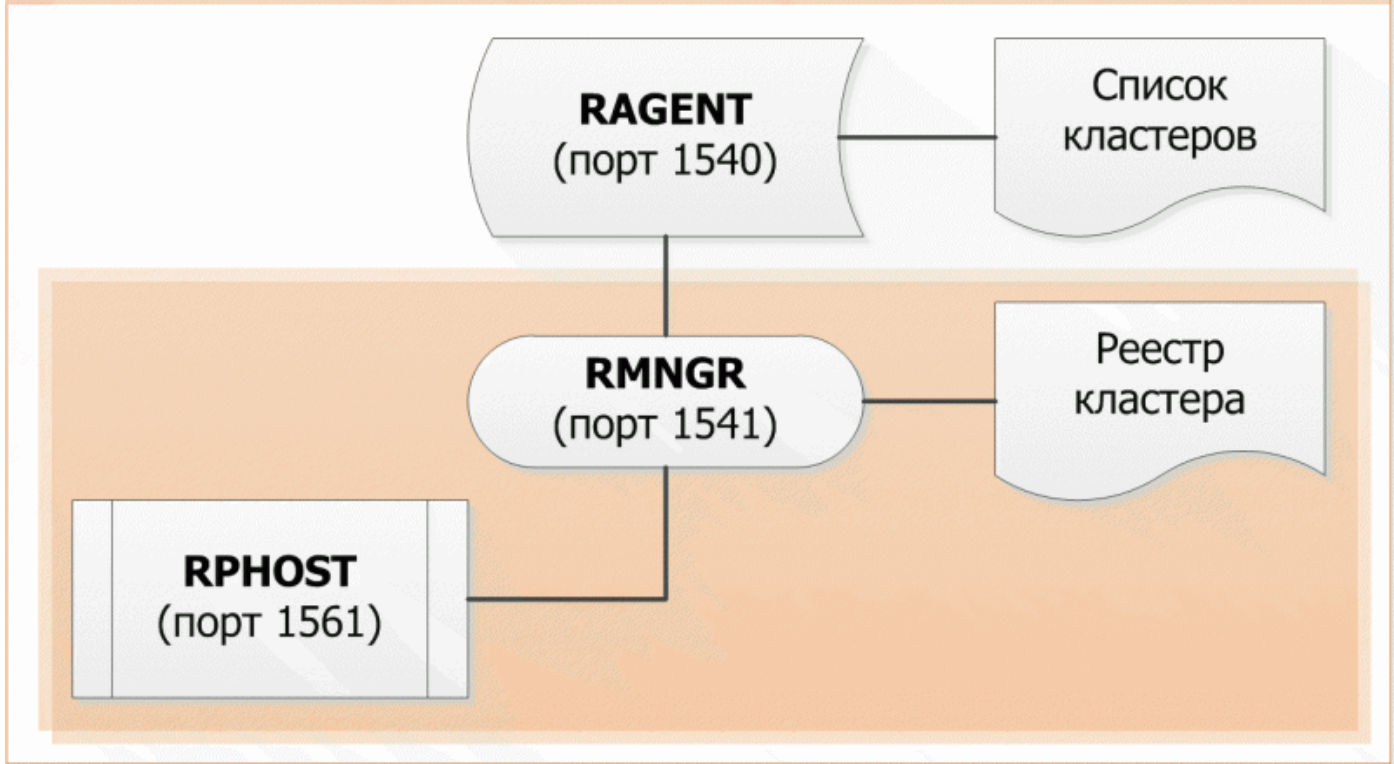
2.1.2. Взаимодействие клиентского приложения с кластером серверов

Взаимодействие различных процессов кластера серверов «1С:Предприятия» между собой и с клиентским приложением осуществляется по протоколу TCP/IP. Таким образом, каждый из процессов кластера адресуется именем рабочего сервера, на котором он запущен, и номером сетевого порта.

По умолчанию при установке кластера серверов используются следующие номера сетевых портов:

- Агент сервера – порт **1540**;
- Менеджер кластера – порт **1541**.
- Для рабочего процесса номер порта выделяется динамически из указанного диапазона, по умолчанию используется диапазон сетевых портов **1560:1591**.

При необходимости можно изменить номера и диапазон используемых портов.

Рабочий сервер (имя **1C_Serv**)**Рис. 4. Простой кластер**

На приведенном рисунке имя центрального сервера – **1C_Serv**. Таким образом, сам центральный сервер адресуется как **1C_Serv:1540**. Кластер, расположенный на этом сервере, адресуется как **1C_Serv:1541**, а рабочий процесс – **1C_Serv:1561**.

Когда клиентское приложение пытается подключиться к информационной базе в варианте клиент-сервер, оно обращается по адресу кластера серверов: **1C_Serv:1541**.

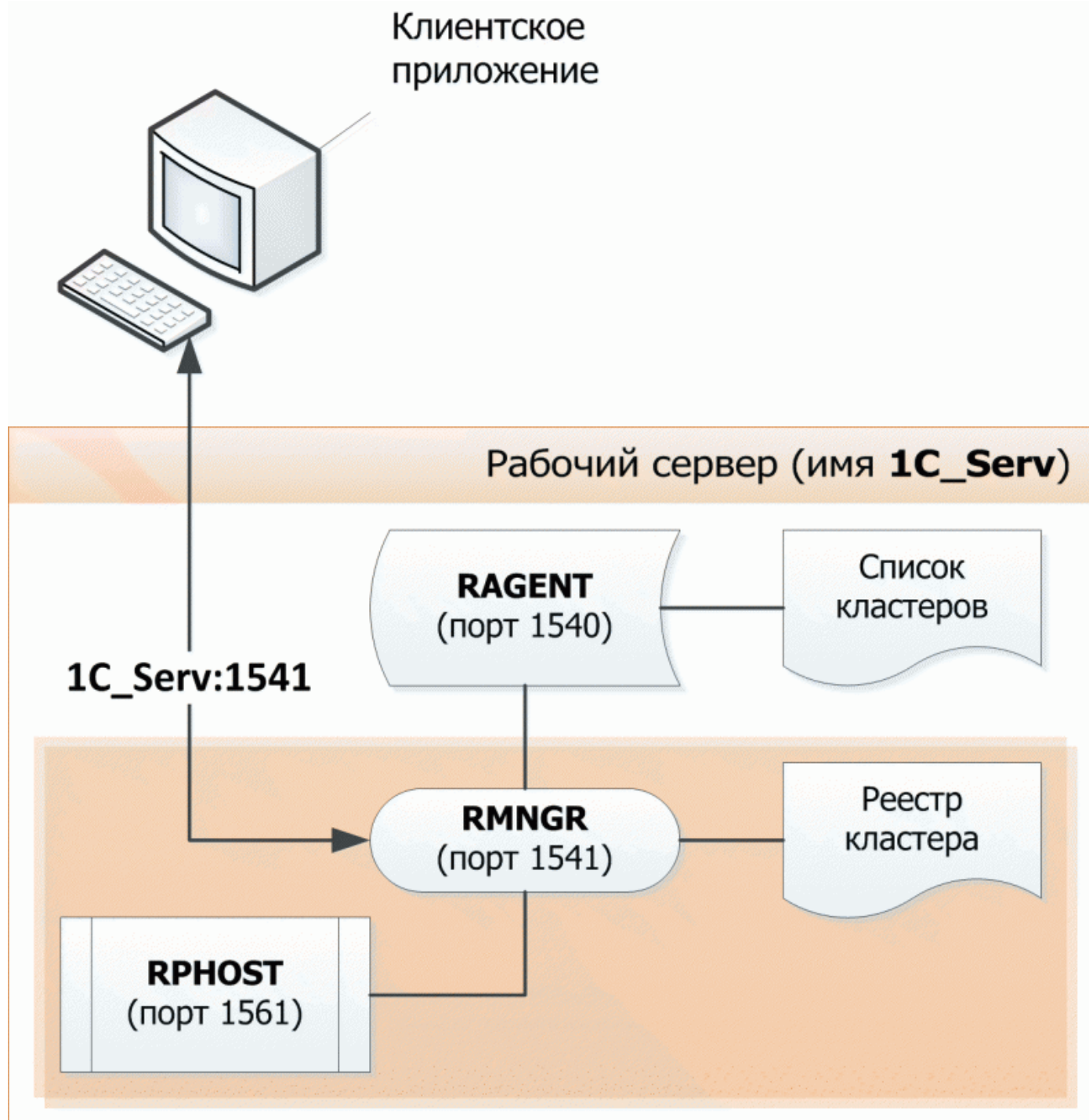


Рис. 5. Обращение к простому кластеру

Менеджер кластера выбирает рабочий процесс, который будет обслуживать клиентское приложение, и сообщает клиентскому приложению его адрес. В данном случае, поскольку рабочий процесс один, это будет **1C_Serv:1561**.

Клиентское приложение

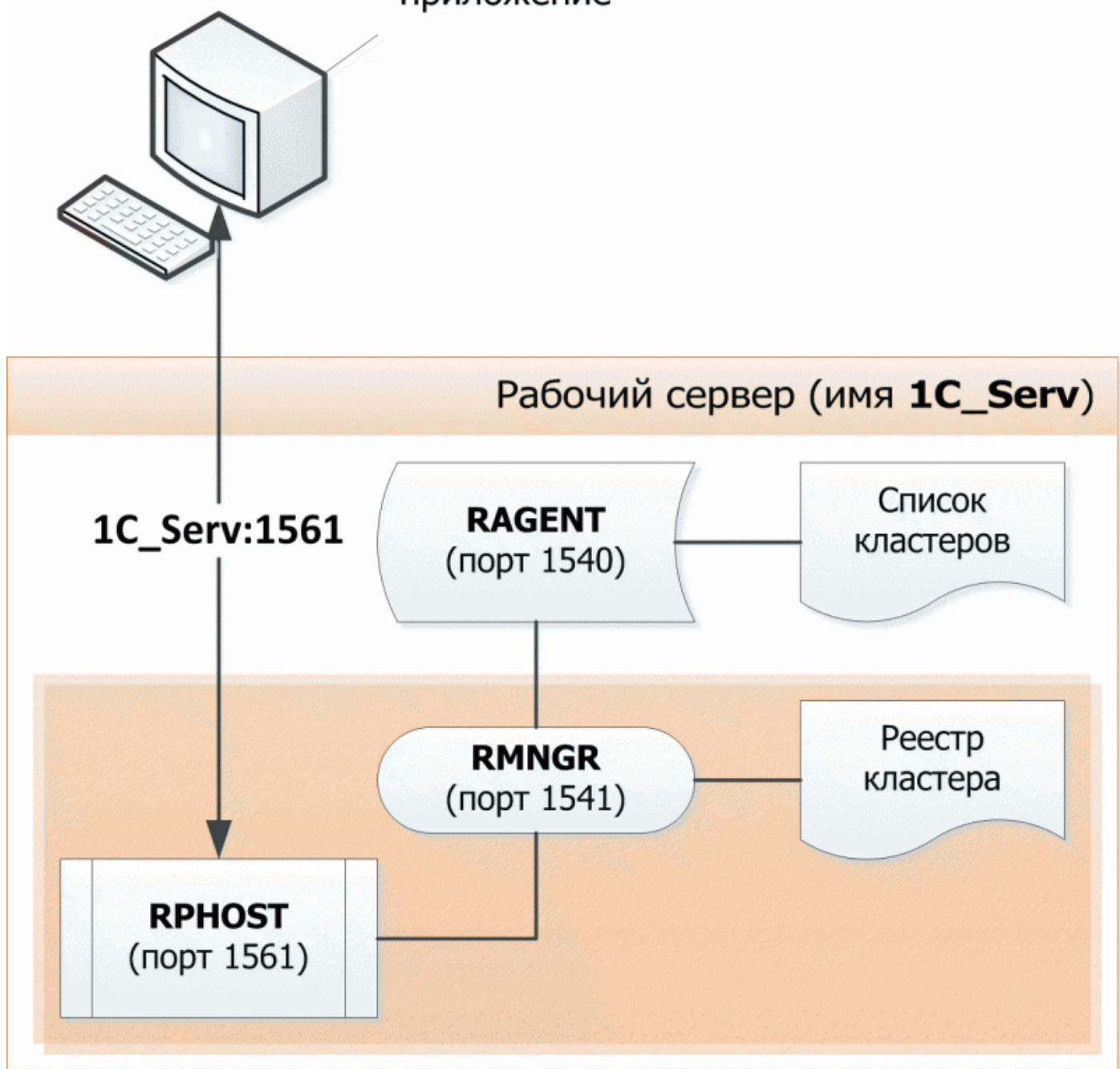


Рис. 6. Установленное соединение

Клиентское приложение связывается с выделенным ему рабочим процессом, который выполняет аутентификацию пользователя информационной базы и осуществляет все дальнейшее взаимодействие клиентского приложения с информационной базой.

ВНИМАНИЕ! Для функционирования системы версии клиентского приложения и сервера должны быть полностью идентичными. Функционирование системы будет невозможно, если, например, сервер имеет версию [8.3.3.100](#), а клиентское приложение – версию [8.3.3.150](#).

2.1.3. Сервисы кластера

Вся функциональность менеджера кластера серверов разделена на несколько независимых сервисов. Каждый сервис обладает некоторыми характеристиками. Далее приведен список сервисов, с кратким описанием и указанием состава характеристик:

Сервис	Описание	Характеристики
Блокировок объектов	Хранит пессимистические (не транзакционные) блокировки объектов.	Память Репликация

		Перенос+ Деление по ИБ
Блокировок кластера	Хранит блокировки информационных баз, сведения об активных процессах и другую динамическую информацию о работе кластера.	Репликация Не переносится
Времени	Поддерживает получение оперативной отметки времени и некоторые вспомогательные функции.	Репликация Перенос+ Деление по ИБ
Журналов регистрации	Поддерживает доступ к журналам регистрации.	Диск Перенос- Деление по ИБ
Заданий	Управляет запуском и отслеживанием времени жизни фоновых и регламентных заданий.	Перенос+
Конфигурации кластера	Хранит все настройки кластера.	Не переносится Репликация
Нумерации	Обеспечивает генерацию уникальных номеров и кодов объектов.	Репликация Перенос+ Деление по ИБ
Полнотекстового поиска	Выполняет полнотекстовый поиск и осуществляет индексирование.	Диск Перенос- Деление по ИБ
Пользовательских настроек	Обеспечивает доступ к файлам, в которых размещаются некоторые пользовательские настройки.	Перенос- Деление по ИБ
Сеансовых данных	Обеспечивает хранение и кеширование сеансовой информации, например, информация форм управляемого приложения. Обеспечивает получение клиентских лицензий.	Диск Репликация Перенос+ Деление по ИБ
Транзакционных блокировок	Содержит транзакционные блокировки управляемого режима.	Память Перенос+ Деление по ИБ
Управления предметами отладки	Управляет подсоединением отладчика к серверным предметам отладки.	Не переносится
Работы с внешними источниками данных через ODBC	Обеспечивает взаимодействие с внешними базами данных с помощью интерфейса ODBC.	Память Перенос+ Деление по ИБ
Сервис лицензирования	Обеспечивает выдачу программных лицензий. Если сервис лицензирования кластера временно недоступен, и через него уже были получены лицензии, то временно выдаются такие же лицензии, которые уже были через него получены ранее, но не дольше, чем в течение 20 секунд.	Перенос+
Сервис фоновой обновления конфигурации	Сервис обслуживает фоновую реструктуризацию базы данных.	Репликация

базы данных	Перемещение данного сервиса на другой менеджер кластера требует остановки всех рабочих процессов. При этом будет остановлено системное фоновое задание. Поэтому после перемещения фоновое обновление будет в приостановленном состоянии.	Перенос+ Деление по ИБ
Сервис внешнего управления сеансами	Сервис управляет возможностью создания сеансов, требующих для своей работы клиентской лицензии.	Перенос+ Деление по ИБ
Сервис работы с внешними источниками данных через XMLA	Обеспечивает взаимодействие с источниками OLAP с помощью интерфейса XMLA.	Перенос- Деление по ИБ
Сервис тестирования	Имитация работы пользователя с кластером «1С:Предприятия»	Перенос-

В таблице использованы следующие термины:

- **Диск** – ресурсоемкий сервис, создает повышенную нагрузку на дисковую подсистему.
- **Память** – ресурсоемкий сервис, создает повышенную нагрузку на процессор и оперативную память.
- **Репликация** – поддерживается репликация между основным и резервными экземплярами сервиса. Репликация возникает в том случае, если уровень отказоустойчивости кластера отличен от 0 (см. [здесь](#)). Для сервисов конфигурации кластера и блокировок кластера, репликация возникает в том случае, если в кластере зарегистрировано несколько рабочих серверов с установленным признаком [Центральный сервер](#).
- **Деление по ИБ** – для каждой информационной базы существует свой экземпляр сервиса.
- Возможность переноса между рабочими серверами:
 - **Перенос+** – сервис может мигрировать между рабочими серверами без потери данных.
 - **Перенос-** – сервис может мигрировать между рабочими серверами с потерей данных.
 - **Не переносится** – сервис не может мигрировать между рабочими серверами и размещается только на компьютере центрального сервера кластера. Для таких сервисов не могут быть созданы требования назначения функциональности (см. [здесь](#)).

Сервисы кластера по типам сервисов, информационным базам и сеансам равномерно распределяются по рабочим серверам кластера.

2.1.4. Сеансы и соединения

2.1.4.1. Общая информация

Сеанс определяет активного пользователя информационной базы и поток управления этого пользователя. Активным пользователем может являться:

- экземпляр клиентского приложения «1С:Предприятия»;
- экземпляр веб-приложения, в котором исполняется веб-клиент;
- экземпляр внешнего соединения (полученный из объекта `V83.COMConnector`);
- один экземпляр фонового задания;
- одно обращение к Web-сервису.

Сеансы могут быть **активными** и **спящими**. Одно из назначений спящего сеанса – сохранение работоспособности клиентского приложения после перехода клиентского компьютера в различные режимы энергосбережения. Сеанс переходит в спящее состояние в двух случаях:

- При нештатном разрыве соединения, назначенного сеансу (для толстого клиента, внешнего соединения, тонкого клиента при прямом соединении с сервером). При физическом отключении сети сервер обнаруживает разрыв соединения с клиентским приложением в течение 2-3 минуты.
- По истечении интервала времени, в течение которого клиентское приложение, использующее сеанс, не проявляется активности (для веб-клиента и тонкого клиента при подключении через веб-сервер). Если компьютер клиента не находится в режиме энергосбережения, и клиентское приложение бездействует (не выполняет никаких действий пользователя), то оно периодически вызывает сервер «1С:Предприятия» с интервалом 5-10 минут для поддержания активности сеанса. Поэтому не рекомендуется устанавливать время засыпания сеанса меньше 10 минут.

Любая активность клиентского приложения, использующего сеанс, приводит к пробуждению сеанса. Спящий сеанс завершается в следующих случаях:

- По истечении интервала времени, который определяет время жизни спящего сеанса.
- Если блокировки, установленные спящим сеансом, конфликтуют с блокировками, которые пытаются установить активные сеансы.

Сеанс, который переходит в спящее состояние, освобождает занятую им клиентскую лицензию. При активизации спящего сеанса он пытается получить клиентскую лицензию (если до перехода в спящий режим сеанс занимал клиентскую лицензию). Невозможность получения лицензии приводит к невозможности восстановления и завершению сеанса.

Время перевода пассивного сеанса в спящее состояние и время, по истечении которого спящий сеанс будет завершен, задается в диалоге настройки параметров информационной базы, которые редактируются с помощью конфигулятора. Подробное описание диалога редактирования параметров информационной базы приводится в книге «1С:Предприятие 8.3. Руководство администратора».

Все данные, хранимые кластером, которые относятся к одному активному пользователю и актуальны только на время работы этого пользователя, являются **данными сеанса**. К данным сеанса относятся:

- информационная база,
- номер сеанса,
- аутентифицированный пользователь информационной базы,
- язык интерфейса,
- значения параметров сеанса,
- временные хранилища,
- статистика работы сеанса,
- информация форм управляемого приложения,
- некоторые внутренние данные платформы.

Данные сеансов сохраняет менеджер кластера. Для этого предусмотрен **сервис сеансовых данных** (см. [здесь](#)). При перезапуске кластера серверов данные сеансов сохраняются. Если активный пользователь не выполнил ни одного обращения к кластеру за определенный интервал времени и сеанс не назначен соединению, то сеанс переводится в спящее состояние. Интервал времени неактивности настраивается в параметрах информационной базы (в конфигуляторе). Значение по умолчанию равно 1200 секунд. Для поддержания сеанса тонкий клиент и веб-клиент обеспечивают обращение к кластеру не реже 1 раза в 10 минут. Для ускорения доступа данные сеансов кешируются в рабочих процессах и в толстых клиентах. В списке активных пользователей показывается список активных сеансов.

Изменения данных сеанса, выполненные за время одного вызова сервера, хранятся в рабочем процессе и передаются менеджеру кластера только при возврате управления клиенту, как штатного, так и в результате программного исключения.

Изменения данных сеанса не сохраняются в менеджере кластера, если:

- в процессе вызова сервера аварийно завершился рабочий процесс;
- при возврате управления клиенту произошла ошибка передачи данных.

Соединение является средством доступа сеансов к кластеру серверов «1С:Предприятия», содержит ограниченное множество данных соединения, не отождествляется с активным пользователем. Также соединения используются для взаимодействия процессов кластера.

Для обращения клиента к кластеру сеанс назначается соединению. Все время, пока клиент не выполняет обращений к кластеру, сеанс может быть не назначен никакому соединению.

Разные варианты использования «1С:Предприятия» по-разному работают с сеансами и соединениями.

- Конфигуратор и толстый клиент:
 - **при старте:** устанавливает соединение, начинает сеанс и назначает его соединению;
 - **при завершении:** отменяет назначение сеанса соединению, заканчивает сеанс и разрывает соединение.

- Одно обращение к Web-сервису и одно выполнение фонового или регламентного задания:
 - **в начале обращения:** выбирает соединение из пула, создает сеанс и назначает его соединению;
 - **в конце обращения:** отменяет назначение сеанса соединению, заканчивает сеанс и возвращает соединение в пул.
- Тонкий клиент и веб-клиент начинают сеанс при старте и заканчивают сеанс при завершении:
 - **в начале обращения** к кластеру выбирается соединение из пула, и ему назначается сеанс данного клиента.
 - **в конце обращения** к кластеру назначение сеанса соединению отменяется, и соединение возвращается в пул.

Информация о сеансах отражается:

- в журнале регистрации,
- консоли кластера,
- средствах программного администрирования,
- технологическом журнале,
- глобальном контексте.

Администратор кластера может получить список существующих сеансов как для всего кластера, так и в разрезе информационных баз. Для этого утилита администрирования кластера и средства программного администрирования имеют соответствующие возможности.

Администратор кластера может закончить сеанс принудительно при помощи утилиты администрирования кластера и средств программного администрирования. При этом работа активного пользователя завершится аварийно. Если удаляется сеанс, назначенный соединению, то происходит разрыв этого соединения.

Администратор кластера с помощью утилиты администрирования и средств программного администрирования может установить блокировку установки сеансов, которая не позволяет начинать новые сеансы, но при этом не препятствует работе существующих сеансов.

2.1.4.2. Виды соединений

Можно выделить два вида соединений:

- соединения с информационной базой,
- служебные соединения с рабочими процессами кластера.

2.1.4.2.1. Соединения с информационной базой

Соединения с информационной базой имеют следующие отличительные особенности:

- соединение выполняется с конкретной информационной базой кластера;
- в таком соединении может выполняться код на встроенном языке;
- соединение может переустанавливаться с течением времени;
- соединение может быть разорвано принудительно командой консоли кластера или средствами встроенного языка;
- наличие соединений с информационной базой у рабочего процесса кластера препятствует остановке и запуску этого рабочего процесса.

Возможны следующие виды соединений с информационной базой:

- Толстый клиент,
- Тонкий клиент,
- Конфигуратор,
- Модуль расширения веб-сервера,
- COM-соединение,

- **Фоновое задание.**

Толстый клиент

Представляет собой соединение толстого клиента с информационной базой. Это соединение предназначено для модификации данных информационной базы и выполнения другой функциональности, предоставляемой конфигурацией информационной базы.

Соединение **Толстый клиент** создается в результате интерактивного запуска толстого клиента в режиме 1С:Предприятие или в результате подключения к информационной базе с использованием технологии Automation Client/Server, например:

[Копировать в буфер обмена](#)

```
// Создать Automation сервер 1С:Предприятия
AutomationСервер = Новый СОМОбъект("V83.Application");
// Установить соединение с информационной базой
// TestBase в кластере 1541 центрального сервера TestSrv
AutomationСервер.Connect("Srvr="TestSrv";Ref="TestBase");
```

Тонкий клиент

Представляет собой соединение тонкого клиента с информационной базой. Это соединение предназначено для модификации данных информационной базы и выполнения другой функциональности, предоставляемой конфигурацией информационной базы.

Соединение **Тонкий клиент** создается в результате интерактивного запуска тонкого клиента или в результате подключения к информационной базе с использованием технологии Automation Client/Server, например:

[Копировать в буфер обмена](#)

```
// Создать Automation сервер 1С:Предприятия
AutomationСервер = Новый СОМОбъект("V83C.Application");
// Установить соединение с информационной базой
// TestBase в кластере 1541 центрального сервера TestSrv
AutomationСервер.Connect("Srvr="TestSrv";Ref="TestBase");
```

Конфигуратор

Представляет собой соединение конфигуратора с информационной базой. Это соединение предназначено для создания и модификации конфигурации информационной базы и для выполнения административных и регламентных действий.

Модуль расширения веб-сервера

Представляет собой соединение веб-сервера с рабочим процессом сервера. Это соединение предназначено для работы веб-клиента, Web-сервисов, а также тонкого клиента (по протоколу HTTP).

Соединение создается в момент обращения к Web-сервису или при обращении веб-клиента или тонкого клиента (по протоколу HTTP) к серверу «1С:Предприятия». Соединение существует до перезапуска веб-сервера или до тех пор, пока соединение находится в пуле соединений модулей расширений веб-сервера (пока не закончится время жизни соединения в пуле, или пока данное соединение не будет вытеснено из пула другими соединениями).

Подробнее о Web-сервисах см. книгу «1С:Предприятие 8.3. Руководство разработчика».

СОМ-соединение

Представляет собой соединение процесса, использующего функциональность внешнего соединения «1С:Предприятия», с информационной базой. Это соединение предназначено для модификации данных информационной базы и выполнения другой функциональности, предоставляемой конфигурацией информационной базы.

СОМ-соединение создается в результате подключения к информационной базе с использованием технологии СОМ, например:

[Копировать в буфер обмена](#)

```
// Создать Automation сервер 1С:Предприятия
СОМСоединитель = Новый СОМОбъект("V83.СОМConnector");
// Установить соединение с информационной базой
// TestBase в кластере 1541 центрального сервера TestSrv
СоединениеСИнформационнойБазой = СОМСоединитель.Connect("Srvr="TestSrv";Ref="TestBase");
```

Фоновое задание

Представляет собой соединение рабочего процесса кластера с информационной базой. Это соединение предназначено для выполнения кода процедуры фонового задания.

Соединение фонового задания создается в случае:

- Получения платформой списка регламентных заданий, зарегистрированных в информационной базе;
- Запуска платформой зарегистрированного в информационной базе регламентного задания;
- Запуска фоновое задания на выполнение из встроенного языка;
- Фонового исполнения отчета,
- Фонового поиска по подстроке,
- Фонового запроса списка,
- Запуска фоновой реструктуризации.

В частности, запуск фоновое задания может выполняться разработчиком, средствами встроенного языка, например:

[Копировать в буфер обмена](#)

```
// Выполнить фоновое задание, описанное в процедуре
// ОбновлениеИндексаПолнотекстовогоПоиска
// общего модуля РегламентныеПроцедуры
ФоновоеЗадание = ФоновыеЗадания.Выполнить ( "РегламентныеПроцедуры.ОбновлениеИндексаПолнотекстовогоПоиска" );
```

Соединение фоновое задания существует до тех пор, пока существует контекст исполняемой процедуры фоновое задания. После того как процедура выполнена или отчет сформирован, соединение фоновое задания закрывается.

Подробнее о фоновых заданиях и фоновом исполнении отчетов см. книгу «1С:Предприятие 8.3. Руководство разработчика».

2.1.4.2.2. Служебные соединения

Служебные соединения имеют следующие отличительные особенности:

- соединение выполняется с рабочим процессом и не ассоциируется с конкретной информационной базой;
- в служебных соединениях код на встроенном языке не выполняется;
- соединение не может быть разорвано принудительно, оно создается и завершается системой;
- наличие служебных соединений не препятствует остановке и запуску рабочих процессов кластера серверов.

Возможны следующие виды служебных соединений:

- [Планировщик заданий](#),
- [Отладчик](#),
- [Консоль кластера](#),
- [Сервер администрирования](#),
- [СОМ-администратор](#),
- [Системное фоновое задание](#).

Планировщик заданий

Представляет собой соединение планировщика заданий с рабочим процессом. Это соединение предназначено для управления работой фоновых заданий, в том числе для запуска регламентных заданий по расписанию. Данное соединение также используется для других случаев обращения менеджера кластера ([rmngr](#)) к рабочему процессу ([rphost](#)), например, при получении списка сеансов.

Соединение планировщика заданий создается при первом запуске фоновое задания. Оно может порождать соединение фоновое задания с информационной базой в том же рабочем процессе кластера серверов. После завершения соединения фоновое задания соединение планировщика заданий не завершается, а существует до тех пор, пока рабочий процесс кластера не будет выключен или удален.

Подробнее о фоновых заданиях см. книгу «1С:Предприятие 8.3. Руководство разработчика».

Отладчик

Представляет собой соединение отладчика с рабочим процессом кластера, находящимся в режиме отладки. Это соединение предназначено для управления ходом отладки и поиском предметов отладки, имеющихся в настоящий

момент.

Соединение отладчика создается при подключении предмета отладки или при поиске предметов отладки. Существует до тех пор, пока предмет отладки не будет отключен или не завершит свою работу.

Подробнее об отладчике см. книгу «1С:Предприятие 8.3. Руководство разработчика».

Консоль кластера

Представляет собой соединение консоли кластера серверов ([mmc](#), см. [здесь](#)) с рабочим процессом. Это соединение предназначено для администрирования информационных баз кластера серверов.

Соединение консоли кластера создается в момент обращения к данным рабочего процесса (например, при получении параметров информационной базы, при получении подробного списка соединений информационной базы и пр.).

Сервер администрирования

Представляет собой соединение сервера удаленного администрирования кластера серверов с рабочим процессом. Это соединение предназначено для администрирования информационных баз кластера серверов.

Соединение удаленного администрирования создается в момент обращения к данным рабочего процесса (например, при получении параметров информационной базы, при получении подробного списка соединений информационной базы и пр.).

COM-администратор

Представляет собой соединение с рабочим процессом сервера с использованием технологии COM. Это соединение предназначено для администрирования информационных баз кластера серверов.

Соединение COM-администратора создается при подключении к выбранному рабочему процессу с использованием технологии COM, например:

[Копировать в буфер обмена](#)

```
// Создать COMСоединитель 1С:Предприятия
COMСоединитель = Новый СОМОбъект("V83.COMConnector");
// Установить соединение с рабочим процессом 1562
// в кластере 1541 центрального сервера TestSrv
СоединениеСРабочимПроцессом = COMСоединитель.ConnectWorkingProcess("tcp://TestSrv:1562");
```

Системное фоновое задание

Представляет собой соединение рабочего процесса кластера с информационной базой. Это соединение предназначено для выполнения фонового обновления конфигурации базы данных.

Соединение фонового задания существует до тех пор, пока выполняется фоновая реструктуризация.

Подробнее о фоновом обновлении конфигурации базы данных см. книгу «1С:Предприятие 8.3. Руководство разработчика».

2.1.4.3. Виды сеансов

Возможны следующие виды сеансов:

- [Толстый клиент](#),
- [Тонкий клиент](#),
- [Веб-клиент](#),
- [Конфигуратор](#),
- [COM-соединение](#),
- [WS-соединение](#),
- [Фоновое задание](#),
- [Консоль кластера](#),
- [Сервер администрирования](#),
- [COM-администратор](#).

Описание сеансов в общем соответствует описанию соответствующего соединения. Описание сеанса веб-клиента приведено ниже.

Веб-клиент

Является представлением в кластере серверов экземпляра веб-клиента. Этот сеанс предназначен для модификации данных информационной базы и выполнения другой функциональности, предоставляемой конфигурацией информационной базы. Обращение веб-клиента к серверу выполняется через соединение [Модуль расширения веб-сервера](#).

Сеанс [Веб-клиент](#) создается в результате интерактивного запуска веб-клиента и существует до тех пор, пока не будет завершен сеанс интерактивной работы с информационной базой (закрывается последнее окно веб-браузера).

2.1.4.4. Внешнее управление сеансами

При эксплуатации клиент-серверных систем возникает задача управления возможностью создания сеансов с информационными базами:

- Ограничить количество одновременно работающих пользователей с одной информационной базой.
- Обеспечить некоторый гарантированный запас лицензий при работе с информационной базой, например, при наличии 100 лицензий обеспечить гарантированную возможность доступа к информационной базе двух пользователей с фиксированными именами.
- Другие аналогичные задачи.

Для реализации этих задач предоставляется механизм **внешнего управления сеансами**. Для работы механизма необходимо наличие специализированного Web-сервиса, который занимается разрешением или запрещением создания сеанса.

Предполагается следующая схема работы механизма:

- При попытке начать сеанс, кластер серверов информирует Web-сервис о необходимости создания сеанса и передает в Web-сервис набор параметров, позволяющих определить все характеристики создаваемого сеанса (имя информационной базы, имя пользователя, кластер серверов и т. д.).
- Web-сервис принимает решение о разрешении или запрещении создания сеанса и возвращает управление кластеру серверов. При необходимости, Web-сервис должен вести учет созданных сеансов в необходимых разрезах.
- При завершении сеанса, кластер серверов также информирует Web-сервис о том, что сеанс завершается.
- Если сеанс переводится в состояние [Спящий](#) или выводится из этого состояния, то кластер серверов также информирует Web-сервис о каждом из этих событий. Данная возможность доступна только для версии 2 сервиса внешнего управления сеансами.

Таким образом, система внешнего управления сеансами точно знает, какое количество сеансов сейчас создано в «подшефных» информационных базах и может принимать необходимые решения.

Описание Web-сервиса, который используется в механизме внешнего управления сеансами, и пример реализации см. [здесь](#).

2.1.5. Отказоустойчивый кластер

Отказоустойчивый кластер обеспечивает бесперебойную работу пользователей в следующих случаях:

- перезапуск рабочих процессов и менеджеров кластера (как плановый, так и аварийный);
- выход из строя сервера, входящего в состав кластера.

Механизмы, которые обеспечивают бесперебойную работу, описаны далее в этом разделе.

2.1.5.1. Транзакционность сеансовых данных

В процессе работы системы сеанс сохраняется при различных сбоях, и возможно восстановление работы после переустановки соединения. Однако существуют ситуации, при которых сеанс завершается системой и предлагается выполнить перезапуск клиентского приложения:

- в процессе вызова сервера, после фиксации первой за этот вызов транзакции, аварийно завершился рабочий процесс;
- при возврате управления клиенту произошла ошибка передачи данных.

2.1.5.2. Повтор последнего вызова

Если в процессе вызова сервера произошла ошибка передачи данных, то это может означать, что:

- произошел обрыв канала связи,
- произошло аварийное завершение рабочего процесса сервера.

Если клиентское приложение вызвало сервер вне транзакции и ошибка передачи данных получена клиентом до фиксации первой за вызов сервера транзакции, то клиент автоматически выполняет процедуру установки соединения с сервером и повторяет тот же самый вызов. После этого его работа продолжается.

Если ошибка передачи данных получена клиентом после фиксации первой за вызов сервера транзакции, то сеанс этого клиента завершается и для продолжения работы требуется перезапуск клиентского приложения.

2.1.5.3. Повтор интерактивного действия

Если клиентское приложение вызвало сервер в транзакции (в обычном режиме работы, а не в режиме управляемого приложения), то ошибка передачи данных считается восстановимой ошибкой и приводит к аварийному завершению интерактивного действия, но не к аварийному завершению клиентского приложения.

Важно, что целостность данных приложения в этом случае не гарантируется платформой, если интерактивное действие состоит более чем из одной транзакции или меняет состояние данных приложения, кроме данных, имеющих логику кеширования.

2.1.5.4. Уровень отказоустойчивости

Уровень отказоустойчивости определяет максимальное количество рабочих серверов, входящих в состав кластера, одновременный выход из строя которых не приведет к аварийному завершению сеансов подключенных пользователей. Надо понимать, что «выход из строя» подразумевает ситуации, подобные следующим: отключение питания компьютера, разрыв сетевого кабеля, проблемы с операционной системой, не позволяющие запустить процесс и т. д.

Таким образом, если в кластер серверов входит только один рабочий сервер, то максимальный уровень отказоустойчивости будет 0, т. к. выход из строя единственного рабочего сервера приведет к аварийному завершению всех подключенных пользователей. Если в кластер входит 4 рабочих сервера, то уровень отказоустойчивости может изменяться от 0 до 3. При этом 0 означает, что фатальным считается выход из строя любого рабочего сервера, а значение 3 означает, что кластер сохранит работоспособность даже в том случае, если выдут из строя 3 из 4 рабочих серверов.

Следует понимать, что увеличение уровня отказоустойчивости выполняется ценой некоторого падения производительности кластера, т. к. кластер будет тратить некоторые ресурсы на синхронизацию данных между рабочими серверами.

Уровень отказоустойчивости связан с количеством центральных серверов в кластере. Количество центральных серверов определяет возможность создания новых соединений. Если, например, в кластер входит два центральных сервера при общем количестве 3 рабочих сервера, то пользователи смогут подключаться к информационным базам в случае аварийного завершения одного центрального сервера. При этом остается два работающих рабочих сервера: один центральный и один рабочий. Если в кластере будет только один центральный сервер, то аварийное завершение этого сервера приведет к тому, что кластер станет недоступен пользователям, несмотря на то, что в нем сохранили работоспособность еще 2 рабочих сервера.

Если в кластере присутствует 3 рабочих сервера (из них один центральный) и установлен уровень отказоустойчивости равный 1, то могут наблюдаться различные ситуации. Рассмотрим их.

Выход из строя одного рабочего сервера

Аварийно завершается работа обычного рабочего сервера. Это соответствует уровню отказоустойчивости, кластер продолжает обслуживать пользователей. При этом подключение новых пользователей также возможно, т. к. сохраняет работоспособность центральный сервер.

Кластер серверов



Рабочий сервер Рабочий сервер

Центральный сервер

Начальная ситуация:
все работает

Кластер серверов



Рабочий сервер Рабочий сервер

Центральный сервер

Произошла авария
рабочего сервера

Кластер серверов



Рабочий сервер Рабочий сервер

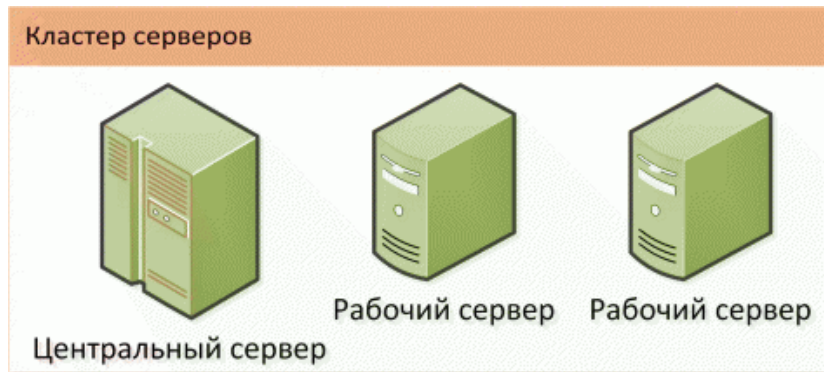
Центральный сервер

Результат аварии:
1. пользователи работают
2. кластер работоспособен

Рис. 7. Аварийное завершение одного рабочего сервера

Выход из строя центрального сервера

Аварийно завершается работа центрального сервера. Это соответствует уровню отказоустойчивости, но кластер прекращает работу, т. к. завершил работу центральный сервер кластера.



Начальная ситуация:
все работает



Произошла авария
центрального сервера



Результат аварии:
1. пользователи отключены
2. кластер не работоспособен

Рис. 8. Аварийное завершение центрального сервера

Выход из строя двух рабочих серверов

Аварийно завершают работу два рабочих сервера, при этом центральный сервер сохраняет работоспособность. При этом превышен уровень отказоустойчивости, как следствие, будут завершены сеансы пользователей, которые обслуживались завершившимися рабочими серверами. Пользователи, которых обслуживает центральный сервер – сохраняют работоспособность. При этом будет возможно подключение новых пользователей.



Рис. 9. Аварийное завершение двух рабочих серверов

Таким образом, можно вывести следующую формулу, связывающую количество центральных серверов в кластере и уровень отказоустойчивости: $\text{Количество центральных серверов} = \text{Уровень отказоустойчивости} + 1$. При этом надо понимать, что буквальное следование этой формуле приведет к некоторому снижению производительности кластера, т. к. на синхронизацию реестра кластера будет тратиться некоторая часть мощности системы. При определении количества центральных серверов и уровня отказоустойчивости следует соблюдать баланс между отказоустойчивостью и приемлемым уровнем производительности кластера серверов, учитывая при этом характеристики оборудования компьютеров, входящих в состав кластера.

2.1.6. Масштабируемость кластера серверов

Масштабируемость кластера серверов обеспечивается несколькими способами:

- Нарращиванием вычислительных мощностей компьютера, на котором развернут единственный рабочий сервер кластера.
- Возможностью включения в состав кластера серверов одного или нескольких новых рабочих серверов.

Все необходимые действия по обеспечению масштабируемости кластер серверов выполняет автоматически. Администратор кластер может влиять на действия кластера серверов с помощью изменения свойств рабочего сервера.

В список рабочих серверов кластера можно добавлять новые сервера и менять свойства существующих (см. [здесь](#)). Измененные значения свойств действуют только на новые соединения и сеансы. Удаление рабочего сервера следует проводить особым образом, чтобы не допустить аварийного отключения пользователей, которых обслуживает удаляемый сервер. Подробнее порядок удаления рабочего сервера см. [здесь](#). Невозможно удаление последнего рабочего сервера в кластере с установленным признаком **Центральный сервер**. При создании кластера по умолчанию, рабочий сервер того компьютера, на котором создается кластер серверов, автоматически включается в список рабочих серверов и для этого рабочего сервера устанавливается флажок **Центральный сервер**.

Во время работы кластер серверов автоматически распределяет нагрузку между рабочими серверами таким образом, чтобы обеспечить минимальное время обслуживания клиентских приложений. Сервисы кластера (см. [здесь](#)) равномерно распределяются по рабочим серверам по типам сервисов, информационным базам и сеансам.

Во время установки соединения с информационной базой, кластер серверов выбирает рабочие сервера с максимальной доступной производительностью (на момент выбора рабочего сервера). Существующие соединения могут быть

перемещены на другой рабочий сервер. Более подробное описание этого механизма см. [здесь](#).

Описание других свойств, управляющих работой рабочего сервера, см. [здесь](#).

2.1.7. Балансировка нагрузки в кластере

2.1.7.1. Доступная производительность рабочего процесса

Каждый рабочий процесс имеет свойство **Доступная производительность**. Оно определяет, насколько быстро данный рабочий процесс способен выполнить эталонный вызов сервера по сравнению с другими рабочими процессами. Эталонный вызов включает в себя следующие операции:

- Операция с памятью: выделение массива, заполнение массива, освобождение массива.
- Операция с файлами: создание, запись, удаление.
- Выполняется определение степени загрузки процессоров компьютера, на котором работает рабочий процесс и количество потоков, ожидающих исполнения. Это значение корректирует время выполнения эталонного вызова в сторону увеличения. Если пользователь, от имени которого работает сервер, не входит в группу **Пользователи журналов производительности (Performance Log Users)**, то определение степени загрузки процессора не выполняется.

Значение свойства **Доступная производительность** вычисляется делением числа 10 000 на среднее (за 5 минут) время выполнения эталонного вызова текущим рабочим процессом. Эталонный вызов выполняется каждые 2 секунды в том случае, если в кластере присутствует несколько рабочих серверов. Если кластер серверов состоит из одного рабочего сервера – все рабочие процессы считаются равноправными.

Клиенты распределяются между рабочими процессами так, чтобы сделать доступную производительность всех рабочих процессов примерно одинаковой. Существенным считается отличие доступной производительности более чем на 25%.

При изменении соотношения между доступной производительностью рабочих процессов клиенты динамически в течение не более 10 минут перераспределяются между рабочими процессами.

При выключении рабочего процесса его клиенты динамически перераспределяются между оставшимися включенными рабочими процессами.

2.1.7.2. Установка нового соединения

2.1.7.2.1. Прямое соединение с сервером

При установке нового соединения с сервером «1С:Предприятия», системе можно указать, каким образом выбирать рабочий процесс (свойство кластера серверов **Режим распределения нагрузки**):

- Приоритет по производительности,
- Приоритет по доступной памяти.

Режим выбора с приоритетом по производительности

При установке нового соединения с сервером «1С:Предприятия» вначале выбирается рабочий сервер с максимальной производительностью. Если таких серверов несколько – рабочий сервер выбирается произвольным образом из этого списка. Затем на выбранном рабочем сервере выбирается рабочий процесс, который обслуживает максимальное количество соединений с необходимой информационной базой. Если таких рабочих процессов несколько – выбирается процесс, который обслуживает максимальное количество соединений с любыми информационными базами. Если и таких рабочих процессов несколько – рабочий процесс выбирается произвольным образом из этого списка.

Существующее соединение с сервером «1С:Предприятия» может быть переустановлено с другим рабочим процессом в одном из следующих случаев:

- текущий рабочий процесс выключен;
- есть рабочий процесс с доступной производительностью не менее чем на 25% большей, чем у данного рабочего процесса.

Переустановка соединения возможна, если выполняются все условия, перечисленные ниже:

- клиентский поток не исполняется на сервере,
- нет открытой транзакции,
- не создано ни одной временной таблицы.

Режим выбора с приоритетом по памяти

При установке нового соединения с сервером «1С:Предприятия» выбирается процесс, который уже обслуживает соединения с нужной информационной базой. Если такой рабочий процесс не обнаружен, то выбирается рабочий процесс, который запущен на сервере с наибольшим количеством свободной оперативной памяти.

Если существует несколько рабочих серверов, которые обслуживают соединения с необходимой информационной базой, то будет выбран сервер с максимальной доступной производительностью. Если на рабочем сервере есть несколько рабочих процессов, обслуживающих соединения с необходимой информационной базой, то будет выбран рабочий процесс с максимальным количеством соединений с этой базой.

Существующее соединение с сервером «1С:Предприятия» может быть переустановлено с другим рабочим процессом в одном из следующих случаев:

- текущий рабочий процесс выключен;
- есть рабочий процесс с доступной производительностью, не менее чем на 25% большей, чем у данного рабочего процесса и обслуживающего ту же информационную базу, что и данный рабочий процесс.

Переустановка соединения возможна, если выполняются все условия, перечисленные ниже:

- клиентский поток не исполняется на сервере,
- нет открытой транзакции,
- не создано ни одной временной таблицы.

2.1.7.2.2. Соединение через расширение веб-сервера

При выполнении обращения к серверу от имени нового сеанса система:

- Выбирает любое соединение из пула соединений, который существует у расширения веб-сервера.
- Если в пуле нет свободных соединений, то производится создание нового соединения в соответствии с параметром кластера [Режим распределения нагрузки](#).

При выполнении обращения от имени существующего сеанса:

- В пуле соединений выполняется поиск соединения с тем же рабочим процессом, через который выполнялось взаимодействие в прошлый вызов. В случае успеха – используется найденное соединение.
- Происходит попытка выбора рабочего процесса, в соответствии с параметром кластера [Режим распределения загрузки](#), при этом приоритет в поиске отдается рабочему процессу, через который выполнялось предыдущее обращение к серверу. Новый рабочий процесс будет выбран в том случае, если он существенно лучше (по производительности или свободной памяти), чем «старый» рабочий процесс. Если к полученному рабочему процессу есть свободные соединения – будет использовано одно из них.
- Иначе происходит создание нового соединения в соответствии с параметром кластера [Режим распределения загрузки](#).

2.1.7.3. Требования назначения функциональности

2.1.7.3.1. Общая информация

Кластер серверов предоставляет некоторый набор функциональных возможностей (называемые **объекты требований**), распределением которых между рабочими серверами внутри кластера можно управлять. Например, можно указать, что все фоновые задания в кластере будут выполняться на выбранном рабочем сервере.

Для того чтобы поместить соединение или сервис кластера на какой-либо рабочий сервер, необходимо для выбранного рабочего сервера создать требование назначения функциональности. Это требование определяет возможность или невозможность конкретного сервера выполнять ту или иную работу. Рассмотрим более подробно, что собой представляет требование назначения функциональности.

Требование назначения функциональности определяет:

- Для какого объекта требования создается требование. В качестве объекта требования могут выступать некоторые сервисы кластера (см. [здесь](#)), клиентские соединения (см. [здесь](#)) и произвольный объект требования. В качестве объекта требования могут выступать следующие сервисы кластера:
 - Блокировок объектов.
 - Времени.
 - Журналов регистрации.

- Заданий.
 - Нумерации.
 - Полнотекстового поиска.
 - Пользовательских настроек.
 - Сеансовых данных.
 - Транзакционных блокировок.
 - Работы с внешними источниками данных через ODBC.
 - Работы с внешними источниками данных через XMLA.
 - Сервис лицензирования.
 - Сервис фоновое обновления конфигурации базы данных.
 - Сервис тестирования.
 - Сервис внешнего управления сеансами.
- Определяет тип требования. Тип требования определяет, каким образом будет выполняться использование рабочего сервера:
 - **Не назначать** – означает, что рабочий сервер, для которого создано данное требование, **не будет** назначен для обслуживания объекта требования, подходящего под условия, заданные в требовании.
 - **Назначать** – означает, что рабочий сервер, для которого создано данное требование, **будет** являться одним из кандидатов на обслуживание данного объекта требования (если рабочих серверов будет несколько).
 - **Авто** – означает, что рабочий сервер **может** быть использован для обслуживания объекта требования в том случае, если нет рабочего сервера с явным указанием необходимости использования.

СОВЕТ. Тип требования **Авто** имеет смысл использовать тогда, когда в списке требований рабочего сервера есть требование с более широким набором условий, и необходимо иметь требование для более узкого набора условий. Например, данный сервер не может обслуживать соединения клиентских приложений для всех информационных баз, кроме одной информационной базы, для которой такое обслуживание разрешено.

- Дополнительные параметры, необходимые кластеру серверов для принятия решения в ряде случаев:
 - Имя информационной базы. Используется для уточнения требования для формирования требований для клиентских соединений и всех сервисов кластера, которые могут выступать в качестве объекта требования, кроме сервиса лицензирования.
 - Дополнительные параметры. Используются для уточнения требований при размещении клиентского соединения или сервиса сеансовых данных. Дополнительный параметр проверяется на совпадение с началом соответствующего параметра объекта требования. Дополнительный параметр может принимать одно из следующих значений:
 - Для указания конкретного фоновое задания: `BackgroundJob.CommonModule.<Имя модуля>.<Имя метода>`;
 - Для указания всех фоновых заданий: `BackgroundJob.CommonModule`;
 - Для указания всех отчетов: `BackgroundJob.Report`. Указание имени отчета не поддерживается;
 - Ввод по строке или поиск в списке: `BackgroundJob.SystemBackgroundJob`;
 - Для указания фоновой реструктуризации: `SystemBackgroundJob`;
 - Для клиентского приложения:
 - `1CV8` – толстый клиент;
 - `1CV8CDirect` – тонкий клиент в случае прямого подключения к серверу «1С:Предприятия»;
 - `Designer` – конфигуратор;

- **COMConnection** – COM-соединение;
- **WebServerExtension** – соединение с информационной базой через веб-сервер: веб-клиент, тонкий клиент в случае подключения через веб-сервер, Web-сервис.

Рассмотрим, как работает кластер серверов при обработке требований.

В случае необходимости выполнить размещение объекта требования, кластер выполняет следующие действия:

- На всех серверах, входящих в состав кластера, выполняется обработка заданных для этих серверов требований назначения функциональности. Обход серверов и требований выполняется в порядке следования этих объектов в консоли кластера.
- В каждом списке требований определяется первое требование, которое удовлетворяет размещаемому объекту: по собственно объекту, информационной базе и дополнительному параметру.
- Затем полученный список рабочих серверов сортируется по признаку типа требования так, что первыми оказываются рабочие сервера с явным указанием использования. Рабочие сервера, для которых подходящее требование содержит явный запрет на использование – исключаются из списка доступных рабочих серверов. При этом назначение выполняется следующим образом:
 - Есть рабочие сервера с явным указанием использования: в этом случае объект требования будет обслужен одним из этих рабочих серверов.
 - Нет рабочих серверов с явным указанием использования: происходит попытка использовать рабочие сервера с автоматическим указанием использования или те рабочие серверы, для которых не указано требований.
- Подробное описание правил выбора рабочего сервера для обслуживания объекта требования см. [здесь](#).
- При размещении клиентского соединения, из списка доступных серверов будет выбран тот, в состав которого входит рабочий процесс с наивысшей доступной производительностью (см. [здесь](#)). Подробное описание правил выбора рабочего процесса в конкретном рабочем сервере см. [здесь](#).

Клиентское приложение, инициировавшее размещение объекта требования, будет завершено аварийно в одном из следующих случаях:

- Если для объекта требования список рабочих серверов оказывается пустым – нет ни одного рабочего сервера, который может обслужить объект. При этом объект требования не будет размещен и будет вызвано исключение.
- Если невозможно выполнить размещение на выбранном рабочем сервере, например, если выбранный сервер вышел из строя, и нет альтернативных рабочих серверов.

2.1.7.3.2. Назначение объектов требований

Более подробно рассмотрим алгоритм назначения рабочего сервера для обслуживания сервиса кластера.

Для сервисов кластера (см. [здесь](#)) объектом требования может выступать:

- Сервис одного типа, если сервис не делится по информационным базам.
- Сервис одного типа для одной информационной базы, если сервис делится по информационным базам.
- Сервис сеансовых данных.
- Сервис лицензирования.

Сервисы распределяются между подходящими рабочими серверами следующим образом:

- Из перечня выбранных для назначения рабочих серверов выбираются те, которые в данный момент работоспособны. Среди оставшихся рабочих серверов выбираются те сервера, у которых свойство **Приоритет** содержит максимальное значение.
- Среди отобранных рабочих серверов сервисы распределяются равномерно.
- Сервисы, поддерживающие репликацию, могут назначаться на несколько рабочих серверов. Количество используемых рабочих серверов равно уровню отказоустойчивости кластера плюс один (см. [здесь](#)). При этом один сервис будет являться активным, а с другими сервисами (резервными) будет поддерживаться репликация служебных данных сервиса. Репликация выполняется асинхронно. Синхронизация выполняется каждую секунду.
- Сервисы, которые не поддерживают перенос данных, назначаются на все рабочие серверы кластера с установленным флажком **Центральный сервер** (см. [здесь](#)).

- Для каждого сеанса, который обслуживает кластер серверов, создается свой экземпляр сервиса сеансовых данных. При отборе рабочих серверов, которые могут обслужить данный экземпляр сервиса, учитываются дополнительные параметры требования. Из списка доступных выбираются сервера с минимальным количеством обслуживаемых сервисов кластера. Количество используемых рабочих серверов равно уровню отказоустойчивости кластера плюс один (см. [здесь](#)).
- В случае необходимости использования сервиса лицензирования следует явно выбрать рабочий сервер, к которому будет выполняться привязка программной лицензии и явно описать в требованиях размещение сервиса на выбранном рабочем сервере.
- Остальные сервисы назначаются в единственном экземпляре.

Переназначение сервисов кластера между рабочими серверами может выполняться в следующих случаях:

- При добавлении рабочего сервера выполняется частичное переназначение сервисов. Данное переназначение выполняется автоматически.
- При удалении рабочего сервера из состава кластера или недоступности рабочего сервера выполняется переназначение только тех объектов требований, которые обслуживал удаляемый рабочий сервер. Данное переназначение выполняется автоматически.
- При удалении или добавлении информационной базы в кластер выполняется частичное переназначение. Данное переназначение выполняется автоматически.
- Переназначение происходит также в том случае, если администратор кластера выполнит операцию полного или частичного применения требований из консоли кластера (см. [здесь](#)).

2.1.7.3.3. Назначение рабочих процессов

При запуске кластера на каждом рабочем сервере запускается по одному рабочему процессу и начинается процесс вычисления доступной производительности для каждого рабочего сервера (см. [здесь](#)).

Установка соединения клиентского приложения с кластером серверов «1С:Предприятия» выполняется по следующим правилам:

- В соответствии с требованиями назначения и ограничениями по использованию оперативной памяти отбирается необходимый рабочий сервер.

Ограничения по использованию оперативной памяти учитываются в том случае, если запрос на установку соединения выполняется к информационной базе, к которой нет установленных соединений на выбранном рабочем сервере. В случае превышения лимита использования оперативной памяти рабочий сервер исключается из списка, если существует другой рабочий сервер, который не превысил лимит. Также исключаются рабочие сервера, которые не могут обработать требуемое соединение исходя из требований назначения.

- Для выбранного сервера определяется список рабочих процессов, которые доступны и могут обслужить запрашиваемое соединение. Рабочий процесс относится к списку доступных рабочих процессов в следующих случаях:
 - Для рабочего процесса не достигнуто максимальное количество обслуживаемых информационных баз (свойство рабочего сервера [Количество ИБ на процесс](#)).
 - Для рабочего процесса не достигнуто максимальное количество обслуживаемых соединений (свойство рабочего сервера [Количество соединений на процесс](#)).
 - Рабочий процесс не находится в состоянии подготовки к автоматическому перезапуску.
- Из выбранных рабочих процессов предпочтение отдается тем рабочим процессам, которые уже обслуживают соединения информационной базы, соединение с которой необходимо обслужить. Если такого рабочего процесса нет – выбирается рабочий процесс с максимальным количеством обслуживаемых соединений.
- Если не удалось выбрать ни один рабочий процесс, то на данном рабочем сервере запускается новый рабочий процесс, который и будет обслуживать запрошенное соединение.

При установке соединения от лица существующего сеанса (если не удалось повторно использовать соединение предыдущего вызова сервера) предпочтение отдается рабочему процессу, который обслуживал предыдущее соединение этого сеанса. При этом возможен выбор другого рабочего процесса, если доступная производительность другого рабочего процесса выше доступной производительности текущего рабочего процесса не менее чем на 25%.

Если на одном рабочем сервере в течении 20 минут существуют 2 рабочих процесса, для которых суммарное количество обслуживаемых соединений и различных информационных баз меньше, чем значения, указанные в свойствах рабочего сервера (свойства [Количество соединений на процесс](#) и [Количество ИБ на процесс](#)), то процесс, который обслуживает меньшее количество соединений, будет помечен как устаревший и будет остановлен после разрыва последнего соединения. Существующим соединениям с «устаревшим» рабочим процессом будет «предложено покинуть» рабочий

сервер при ближайшем серверном вызове через данное соединение. При этом «устаревший» рабочий процесс не участвует в распределении запросов на обслуживание новых объектов требований.

При подсчете количества обслуживаемых соединений учитываются соединения, которые создаются отладчиком для проверки прав доступа на предмет отладки.

2.1.7.4. Примеры управления кластером

При рассмотрении примеров требований назначений функциональности будет использоваться следующий кластер серверов:



Рис. 10. Кластер для примеров требований

Кластер обладает следующими характеристиками:

- Количество рабочих серверов: 3.
- Уровень отказоустойчивости: 1.
- Количество центральных серверов: 2 (SRV1, SRV2).
- Операционные системы на рабочих серверах:
 - Сервер SRV1, ОС Windows.
 - Сервер SRV2, ОС Linux.
 - Сервер SRV3, ОС Windows.

Кластер обслуживает следующие информационные базы:

- DemoDB,
- WorkDB.

ВНИМАНИЕ! Приведенные ниже примеры не являются законченными решениями какой-либо проблемы, а служат только для демонстрации принципов работы механизма размещения объектов требований по рабочим серверам в кластере.

2.1.7.4.1. Размещение всех фоновых заданий на одном рабочем сервере

Если необходимо разместить все фоновые задания на рабочем сервере SRV1, то для этого необходимо для рабочего сервера SRV1 задать следующее требование назначения функциональности:

- Объект требования: Клиентское соединение с ИБ.
- Тип требования: Назначать.

- Имя ИБ: не указывается.
- Значение дополнительного параметра: `BackgroundJob.CommonModule`.

2.1.7.4.2. Размещение сервиса лицензирования на выделенном рабочем сервере

Необходимо многопользовательскую клиентскую лицензию активировать для компьютера, на котором функционирует рабочий сервер [SRV2](#), разместить на этот компьютер сервис лицензирования и не размещать на этот компьютер больше никаких сервисов. Тогда для рабочего сервера [SRV2](#) следует указать следующие требования:

- Требование 1:
 - Объект требования: [Сервис лицензирования](#).
 - Тип требования: [Назначать](#).
 - Имя ИБ: не указывается.
 - Значение дополнительного параметра: не указывается.
- Требование 2:
 - Объект требования: [Любой объект требования](#).
 - Тип требования: [Не назначать](#).
 - Имя ИБ: не указывается.
 - Значение дополнительного параметра: не указывается.

[Требование 1](#) обеспечит функционирование сервиса лицензирования на сервере [SRV2](#), а [Требование 2](#) обеспечит функционирование на сервере [SRV2](#) только сервиса лицензирования (на сервере [SRV2](#) не будут функционировать другие сервисы кластера).

При активации программной лицензии с помощью сервера «1С:Предприятия» следует указывать имя [SRV2](#), в противном случае активированная лицензия не сможет быть использована кластером серверов, т. к. программная лицензия будет активирована для другого компьютера.

2.1.7.4.3. Запрет размещения сервиса работы с внешними источниками данных на одном рабочем сервере

Необходимо разрешить работу сервиса работы с внешними источниками данных на рабочих серверах [SRV1](#) и [SRV3](#), и запретить – на рабочем сервере [SRV2](#). Для этого следует для рабочего сервера [SRV2](#) указать следующее требование:

- Объект требования: [Сервис работы с внешними источниками данных](#).
- Тип требования: [Не назначать](#).
- Имя ИБ: не указывается.
- Значение дополнительного параметра: не указывается.

2.1.7.4.4. Один рабочий процесс обслуживает одну информационную базу

Необходимо настроить кластер серверов таким образом, чтобы каждая информационная база обслуживалась одним рабочим процессом. Для этого необходимо для каждого рабочего сервера установить свойство [Количество ИБ на процесс](#) в значение 1.

В результате на каждом сервере будет создано по два рабочих процесса (всего – 6, по 2 рабочих процесса на каждый из 3 рабочих серверов). При этом обслуживанием одной информационной базы будет заниматься 3 рабочих процесса на 3 рабочих серверах.

2.1.7.4.5. Назначение рабочих серверов для обслуживания выбранных информационных баз

Необходимо настроить кластер серверов таким образом, чтобы информационную базу [DemoDB](#) обслуживал только рабочий сервер [SRV3](#), а информационную базу [WorkDB](#) обслуживали оба рабочих сервера: [SRV1](#) и [SRV2](#). Для этого необходимо настроить следующие правила:

- Для рабочего сервера [SRV3](#):
 - Объект требования: [Любой объект требования](#).

- Тип требования: [Назначать](#).
- Имя ИБ: [DemoDB](#).
- Значение дополнительного параметра: не указывается.
- Для рабочих серверов [SRV1](#) и [SRV2](#):
 - Объект требования: [Любой объект требования](#).
 - Тип требования: [Назначать](#).
 - Имя ИБ: [WorkDB](#).
 - Значение дополнительного параметра: не указывается.

Указанные правила «разнесут» по рабочим серверам все механизмы кластера серверов: соединения, фоновые задания, сервисы сеансовых данных и т. д.

2.1.7.4.6. Назначение конкретных фоновых заданий на конкретный рабочий сервер

Необходимо настроить кластер серверов таким образом, чтобы на рабочем сервере [SRV1](#) выполнялись только отчеты, на рабочем сервере [SRV2](#) выполнялись только регламентные задания [ОбновлениеИндексаПД](#) и [ОбновлениеАгрегатовПродаж](#), а остальные фоновые задания должны выполняться на рабочем сервере [SRV3](#). Для этого необходимо настроить следующие правила:

- Для рабочего сервера [SRV1](#):
 - Требование:
 - Объект требования: [Клиентское соединение с ИБ](#).
 - Тип требования: [Назначать](#).
 - Имя ИБ: не указывается.
 - Значение дополнительного параметра: [BackgroundJob.Report](#).
- Для рабочего сервера [SRV2](#):
 - Требование:
 - Объект требования: [Клиентское соединение с ИБ](#).
 - Тип требования: [Назначать](#).
 - Имя ИБ: не указывается.
 - Значение дополнительного параметра: [BackgroundJob.CommonModule.РаботаСПолнотекстовымПоиском.ОбновлениеИндексаПолнотекстовогоПоиска](#).
 - Требование:
 - Объект требования: [Клиентское соединение с ИБ](#).
 - Тип требования: [Назначать](#).
 - Имя ИБ: не указывается.
 - Значение дополнительного параметра: [BackgroundJob.CommonModule.РегламентныеЗаданияАгрегатов.ОбновлениеАгрегатовПродаж](#).

2.1.8. Профили безопасности

2.1.8.1. Общая информация

Работающее прикладное решение может использовать для своих нужд различные внешние ресурсы: каталоги файловой системы, COM-объекты (на Windows-системах), внешние компоненты, приложения ОС и т. д. Однако, по соображениям безопасности, для различных прикладных решений могут быть доступны не все возможные внешние ресурсы, а только ограниченное их подмножество. Может потребоваться создать собственный каталог временных файлов для каждой области разделенной информационной базы или задать перечень ресурсов сети Интернет, к которым может иметь

доступ прикладное решение.

Чтобы установить такие ограничения, для кластера серверов можно настроить **профили безопасности**. Профиль безопасности – это набор явно заданных разрешений использования теми или иными внешними ресурсами (с указанием перечня таких ресурсов), которые можно назначать информационным базам, зарегистрированным в кластере. Профили безопасности создаются администратором кластера и позволяют настраивать следующие разрешения:

- Возможность использования данного профиля в качестве профиля безопасности безопасного режима – если это разрешение установлено (свойство профиля [Может использоваться как профиль безопасности безопасного режима](#)), то имя данного профиля безопасности можно указывать во встроенном языке при включении безопасного режима и подключении внешних отчетов и обработок;
- Доступные ресурсы файловой системы сервера (см. [здесь](#));
- Доступные COM-объекты (только для серверов, работающих под управлением ОС Windows) (см. [здесь](#));
- Доступные внешние компоненты (см. [здесь](#));
- Доступные внешние модули (внешние отчеты, обработки, расширения конфигурации), а также возможность использования оператора **Выполнить ()** и функции **Вычислить ()** (см. [здесь](#));
- Доступные приложения операционной системы (см. [здесь](#));
- Доступные ресурсы сети Интернет (см. [здесь](#));
- Возможность доступа к привилегированному режиму – если данное разрешение установлено (свойство профиля [Разрешен полный доступ: к привилегированному режиму](#)), то при использовании данного профиля в качестве профиля безопасного режима, разрешено включать привилегированный режим (см. [здесь](#)).

Для того чтобы информационная база использовала для работы созданный профиль безопасности, следует указать его имя в свойствах информационной базы (с помощью средств администрирования кластера серверов). Также в свойствах информационной базы можно указать имя профиля, который будет использоваться при установке в прикладном решении безопасного режима.

Некоторые разрешения позволяют задавать список разрешенных ресурсов. Это разрешения:

- Доступные ресурсы файловой системы сервера;
- Доступные COM-объекты (только для серверов, работающих под управлением ОС Windows);
- Доступные внешние компоненты;
- Доступные внешние модули;
- Доступные приложения операционной системы;
- Доступные ресурсы сети Интернет.

В этом случае работа осуществляется следующим образом:

- Если в профиле безопасности флажок сброшен (например, доступ к ресурсам сети Интернет), то из информационной базы, для которой указан данный профиль безопасности, невозможно использовать ни одного ресурса сети Интернет.
- Если в профиле безопасности этот флажок установлен, то прикладное решение обладает полным доступом к ресурсам сети Интернет.
- Если требуется предоставить ограниченный доступ (по списку), то следует отключить флажок и задать список ресурсов, доступ к которым разрешен.

Далее подробнее рассмотрим работу с конкретными разрешениями.

2.1.8.2. Ресурсы файловой системы сервера

Для доступа к файловым ресурсам сервера применяются **виртуальные каталоги**. Это означает, что в рамках профиля безопасности существует некоторая виртуальная файловая система, в которой создаются каталоги. Каждый виртуальный каталог имеет отражение на реальную файловую систему по определенным правилам. В тот момент, когда прикладному решению необходимо выполнить файловую операцию, в параметре соответствующей функции указывается путь к файлу, расположенному в виртуальной файловой системе. «1С:Предприятие» транслирует виртуальный каталог в реальный и формирует реальный путь к файлу, с которым и выполняется реальная работа. Прикладное решение не может получить информацию о том, в какой физический путь будет отражен виртуальный каталог.

Если в профиле безопасности указаны несколько виртуальных каталогов, то прикладное решение может осуществлять доступ только к этим ресурсам. Попытка доступа к любому другому каталогу (как реальному, так и виртуальному) – невозможна.

Виртуальные каталоги используются при обращении к методам встроенного языка:

- Глобальный контекст:
 - `КаталогПрограммы()`,
 - `КаталогВременныхФайлов()`,
 - `ЗначениеВФайл()`,
 - `ЗначениеИзФайла()`,
 - `КопироватьФайл()`,
 - `ПереместитьФайл()`,
 - `УдалитьФайлы()`,
 - `НайтиФайлы()`,
 - `СоздатьКаталог()`,
 - `РазделитьФайл()`,
 - `ОбъединитьФайлы()`,
 - `ПолучитьИмяВременногоФайла()`.
- Объект `Картинка`:
 - конструктор на основании имени файла,
 - `Записать()`.
- Объект `ДвоичныеДанные`:
 - конструктор на основании имени файла,
 - `Записать()`.
- Объект `ИзвлечениеТекста`:
 - конструктор на основании имени файла,
 - изменение свойства `ИмяФайла`,
 - `Записать()`.
- Объект `ТекстовыйДокумент`:
 - `Записать()`,
 - `Прочитать()`.
- Объект `ТабличныйДокумент`:
 - `Записать()`,
 - `Прочитать()`.
- Объект `ФорматированныйДокумент`:
 - `Записать()`.
- Объект `ГрафическаяСхема`:

- `Записать()`,
- `Прочитать()`.
- **Объект `ГеографическаяСхема`:**
 - `Записать()`,
 - `Прочитать()`.
- **Объект `Файл`:**
 - конструктор на основании имени файла.
- **Объект `XBase`:**
 - конструктор на основании имени файла.
 - `ОткрытьФайл()`,
 - `СоздатьИндексныйФайл()`,
 - `СоздатьФайл()`.
- **Объект `ЧтениеXML`:**
 - `ОткрытьФайл()`.
- **Объект `ЗаписьXML`:**
 - `ОткрытьФайл()`.
- **Объект `КаноническаяЗаписьXML`:**
 - `ОткрытьФайл()`.
- **Объект `ПреобразованиеXSL`:**
 - `ЗагрузитьИзФайла()`,
 - `ПреобразоватьИзФайла()`.
- **Объект `ЧтениеFastInfoSet`:**
 - `ОткрытьФайл()`.
- **Объект `ЗаписьFastInfoSet`:**
 - `ОткрытьФайл()`.
- **Объект `ЗаписьZipФайла`:**
 - конструктор на основании имени файла,
 - `Добавить()`,
 - `Открыть()`.
- **Объект `ЧтениеZipФайла`:**
 - конструктор на основании имени файла,
 - `Извлечь()`,
 - `ИзвлечьВсе()`,
 - `Открыть()`.

- Объект `СертификатКлиентаФайл`:
 - конструктор по умолчанию.
- Объект `СертификатыУдостоверяющихЦентровФайл`:
 - конструктор по умолчанию.
- Объект `ЗаписьHTML`:
 - `ОткрытьФайл()`.
- Объект `ЧтениеHTML`:
 - `ОткрытьФайл()`.
- Объект `ЧтениеТекста`:
 - конструктор на основании имени файла,
 - `Открыть()`.
- Объект `ЗаписьТекста`:
 - конструктор на основании имени файла,
 - `Открыть()`.
- Объект `СертификатКриптографии`:
 - конструктор на основании имени файла.
- Объект `МенеджерКриптографии`:
 - `Подписать()`,
 - `ПроверитьПодпись()`,
 - `Зашифровать()`,
 - `Расшифровать()`,
 - `ПолучитьСертификатыИзПодписи()`.
- Объект `ХешированиеДанных`:
 - `ДобавитьФайл()`.

Виртуальный каталог описывается несколькими параметрами:

- Логический URL – адрес, который будет использоваться прикладным решением. Этот параметр должен выглядеть как начало реального пути в соответствующей файловой системе (для ОС Windows или Linux). Прикладное решение может использовать только такие пути, которые начинаются со значения, указанного в этом свойстве. Является уникальным в рамках одного профиля.

Существуют два предопределенных логических URL, которые используются методами системы:

- `/bin` – каталог загрузочных модулей текущей версии «1С:Предприятия». Этот виртуальный каталог используется методом `КаталогПрограммы()`.
- `/temp` – каталог временных файлов. Этот виртуальный каталог используется методом `КаталогВременныхФайлов()`.
- Физический URL – адрес, указывающий физическое размещение логического URL в файловой системе сервера. Может включать специальные подстановочные символы. В процессе выполнения файловой операции «1С:Предприятие» преобразует логический URL в реальный адрес файловой системы заменой всех подстановочных символов. При этом каждая последовательность символов, не допустимых в URL, заменяется на символ "_".

В адресе можно использовать следующие подстановочные символы:

- %r – ссылочное имя информационной базы;
 - %i – идентификатор информационной базы;
 - %z – строковое представление текущих значений разделителей текущего сеанса в формате, принятом для параметра командной строки /Z.
 - %s – номер сеанса;
 - %c – номер соединения;
 - %p – идентификатор безопасного режима исполнения прикладного кода;
 - %e – каталог загрузочных модулей «1С:Предприятия»;
 - %t – текущий каталог временных файлов операционной системы;
 - %u – каталог данных приложений текущего пользователя;
 - %a – каталог данных приложений всех пользователей;
 - %n – имя текущего пользователя информационной базы;
 - %% – символ %.
- Разрешено чтение данных – флажок определяет, допустимо или нет чтение файлов из данного виртуального каталога.
 - Разрешена запись данных – флажок определяет, допустима или нет запись файлов в данном виртуальном каталоге.

При указании физического URL возможно указание как на собственные каталоги компьютера, на котором установлено «1С:Предприятие» и на сетевые ресурсы. При этом надо учитывать особенности организации файловых систем и работы с сетевыми ресурсами той операционной системы, на которой функционирует сервер.

При завершении сеанса удаляются физические каталоги, в определении которых участвовал символ подстановки %s. При завершении соединения удаляются физические каталоги, в определении которых участвовал символ подстановки %c. Вход в безопасный режим исполнения кода на встроенном языке обладает собственным уникальным идентификатором. При выходе из безопасного режима удаляются физические каталоги, в определении которых участвовал символ подстановки %p.

2.1.8.3. COM-объекты сервера

Профиль безопасности кластера может содержать список классов COM-объектов, которые можно использовать в прикладном решении.

ВНИМАНИЕ! Данная возможность применима только для серверов, работающих под управлением ОС Windows.

Если информационная база ссылается на профиль безопасности, в котором ограничено использование COM-объектов, то возможно использование только тех классов COM-объектов, которые перечислены в списке разрешенных COM-классов этого профиля. Используемый конфигурацией COM-объект соответствует элементу списка разрешенных COM-объектов профиля безопасности, если совпадает значение свойства **Компьютер COM-объекта** и совпадают отличные от пустых значения свойства **Файл (моникер)** и **Идентификатор COM-класса**. При попытке создания экземпляра любого COM-объекта, отсутствующего в списке, будет выдано исключение.

Разрешенный класс COM-объекта описывается следующими параметрами:

- Имя – уникальное имя класса COM-объекта. Является уникальным в рамках одного профиля.
- Файл (моникер) – имя файла-моникера. Используется при вызове метода **ПолучитьCOMОбъект()** с неуказанным значением параметра **ИмяКлассаCOM**. Подробнее про файл-моникер можно прочитать по адресу: [http://msdn.microsoft.com/ru-RU/library/windows/desktop/ms688670\(v=vs.85\).aspx](http://msdn.microsoft.com/ru-RU/library/windows/desktop/ms688670(v=vs.85).aspx) (на английском языке).
- Идентификатор COM-класса – строка, представляющая собой идентификатор класса COM-объекта, в формате реестра ОС Windows, без обрамляющих фигурных скобок. Данное значение используется в конструкторе нового COM-объекта и методе **ПолучитьCOMОбъект()**.

- Компьютер COM-объекта – компьютер, на котором может быть создан COM-объект. Используется конструктором нового COM-объекта. Для указания текущего компьютера следует использовать строку `localhost`, пустая строка означает, что COM-объект может создаваться на любом компьютере.

2.1.8.4. Внешние компоненты

Профиль безопасности кластера может содержать список внешних компонент, которые можно использовать в прикладном решении. Если информационная база ссылается на профиль безопасности, в котором ограничено использование внешних компонент, то возможно использование только тех внешних компонент, которые перечислены в списке разрешенных внешних компонент этого профиля. При попытке выполнения метода `ПодключитьВнешнююКомпоненту()` для компоненты, отсутствующей в списке, будет выдано исключение. Пустой список значит, что данный профиль не позволяет использовать никакие внешние компоненты.

Разрешенная внешняя компонента описывается следующими параметрами:

- Имя – уникальное имя внешней компоненты. Является уникальным в рамках одного профиля.
- Контрольная сумма – контрольная сумма `SHA1` разрешенной внешней компоненты, преобразованная к формату `base64`.

Для построения контрольной суммы можно использовать объект `ХешированиеДанных` и метод глобального контекста `Base64Строка()`.

Если оформляется разрешение для внешней компоненты, разработанной по технологии `Native API`, то для каждого, поддерживаемого компонентой, типа платформы необходимо задавать отдельное разрешение. Таким образом, внешняя компонента, поддерживающая все платформы, будет иметь 4 разрешения: для 32- и 64-разрядной ОС `Windows` и для 32- и 64-разрядной ОС `Linux`.

2.1.8.5. Внешние модули

Профиль безопасности кластера может содержать список внешних модулей, которые можно использовать из кода прикладного решения без включения безопасного режима. Под внешними модулями понимаются: внешние отчеты и обработки, а также расширения конфигурации. Если информационная база ссылается на профиль безопасности, в котором ограничено использование внешних модулей, то возможно использование без включения безопасного режима только тех внешних модулей, которые перечислены в этом списке. При попытке использования внешнего модуля, не содержащегося в этом списке, без включения безопасного режима, будет выдано исключение. Пустой список значит, что данный профиль не позволяет использовать никакие внешние модули без включения безопасного режима. Также с помощью данного элемента профиля управляется возможность использования оператора `Выполнить()` и функции `Вычислить()` в безопасном режиме. Если в профиле безопасности флажок `Разрешен полный доступ: к внешним модулям` сброшен, то в прикладном решении можно использовать `Выполнить()` и `Вычислить()` только в безопасном режиме (явно устанавливая его перед использованием оператора или функции). Если прикладное решение использует `Выполнить()` и `Вычислить()` в технологических целях (в небезопасном режиме), то для него нельзя использовать профиль безопасности, у которого сброшен флажок `Разрешен полный доступ: к внешним модулям`.

Разрешенный внешний отчет или обработка описывается следующими параметрами:

- Имя – уникальное имя внешнего модуля. Является уникальным в рамках одного профиля.
- Контрольная сумма – контрольная сумма `SHA1` разрешенного внешнего отчета или обработки, преобразованная к формату `base64`.

Для построения контрольной суммы можно использовать объект `ХешированиеДанных` и метод глобального контекста `Base64Строка()`. Для расширения конфигурации контрольную сумму можно получить или с помощью программного интерфейса или с помощью Конфигуратора (команда `главное меню – Конфигурация – Расширения конфигурации – Действия – Показать контрольную сумму`).

2.1.8.6. Приложения операционной системы

Профиль безопасности кластера может содержать список приложений, которые можно запускать из прикладного решения. Если информационная база ссылается на профиль безопасности, в котором ограничен запуск приложений, то возможно использование только тех приложений (исполняемых файлов), которые перечислены в списке разрешенных приложений этого профиля. При попытке выполнения метода `ЗапуститьПриложение()` с приложением или параметрами, которым не соответствует ни одной записи в списке, будет выдано исключение. Пустой список значит, что данный профиль не позволяет использовать никакие приложения.

Разрешенное приложение операционной системы описывается следующими параметрами:

- Имя – имя приложения. Уникально в пределах профиля безопасности.
- Шаблон строки запуска – шаблон строки запуска приложения. Состоит из последовательности шаблонных слов, разделенных пробелами. Шаблонное слово представляет собой произвольную последовательность символов и может содержать символы подстановки. Если шаблонное слово может содержать пробелы, то оно должно быть заключено в

кавычки.

Символы подстановки:

- `%` – произвольная последовательность символов;
- `_` – один произвольный символ;
- `*` – имя файла. Если шаблонное слово начинается с символа `*`, то параметр является маршрутом от виртуального каталога. Перед выполнением виртуальный каталог заменяется на реальный каталог.
- `\` – префикс символа. Указание данного символа перед шаблонным словом делает слово не шаблонным.

Проверка соответствия запускаемого приложения списку разрешенных приложений выполняется для имени запускаемого приложения и каждого из параметров, имеющихся в команде запуска. Отсутствие соответствующего шаблонного слова значит, что шаблон не соответствует строке запуска.

Например, шаблон `xcopy */temp/% */userdata/%` позволяет выполнять копирование произвольного файла из виртуального каталога `/temp` в виртуальный каталог `/userdata`.

В строке запуска параметры разделяются одним или несколькими пробелами.

Строка, заключенная в кавычки (`"`), является одним параметром. Одним параметром считается последовательность слов, если первое и последнее слово содержит нечетное количество одиночных символов кавычки. Если есть первое слово с кавычками, но нет второго, то одним параметром считается все до конца строки. Если параметр не заключен в кавычки, то символ кавычки представляется парой символов `\`. Символ кавычки в составе параметра, заключенного в кавычки, может представляться одной из пар символов: `\` или `"`.

2.1.8.7. Ресурсы сети Интернет

Профиль безопасности может содержать список ресурсов интернет, к которым разрешены обращения из серверного кода прикладного решения. Если в профиле безопасности ограничено использование интернет, то использование объекта [ИнтернетСоединение](#) запрещено, а при помощи объектов [ИнтернетПочта](#), [HTTPСоединение](#), [FTPСоединение](#), [WSOпределения](#) разрешено обращение только к тем ресурсам, которые перечислены в списке разрешенных ресурсов интернет этого профиля. При обращении к другим ресурсам будет порождено исключение. Пустой список значит, что запрещено обращение к любым ресурсам интернет.

Разрешенный ресурс сети Интернет описывается следующими параметрами:

- Имя – имя ресурса для его идентификации. Уникально в пределах профиля безопасности.
- Адрес – адрес ресурса без указания протокола.
- Порт – номер порта, через который выполняется взаимодействие с указанным ресурсом.
- Тип ресурса (протокол) – протокол, по которому выполняется взаимодействие с ресурсом. Могут быть указаны следующие протоколы (регистр символов не играет роли):
 - `imap` – сервер электронной почты, работающий по протоколу IMAP.
 - `pop3` – сервер электронной почты, работающий по протоколу POP3.
 - `smtp` – сервер электронной почты, работающий по протоколу SMTP.
 - `http` – веб-сервер.
 - `https` – защищенное соединение с веб-сервером.
 - `ftp` – ftp-сервер.
 - `ftps` – защищенное соединение с ftp-сервером.

2.1.8.8. Привилегированный режим

При установленном разрешении [Привилегированный режим](#) разрешено включать привилегированный режим в том случае, если профиль с установленным разрешением выбран в качестве профиля безопасного режима. При этом возможность получать доступ к различным внешним ресурсам определяется параметрами установленного профиля безопасности.

2.1.9. Безопасность кластера серверов

При работе «1С:Предприятия» в варианте клиент-сервер задача обеспечения безопасности данных сводится к тому, что доступ к данным «1С:Предприятия» должен осуществляться только механизмами «1С:Предприятия». В связи с этим можно выделить несколько областей обеспечения безопасности.

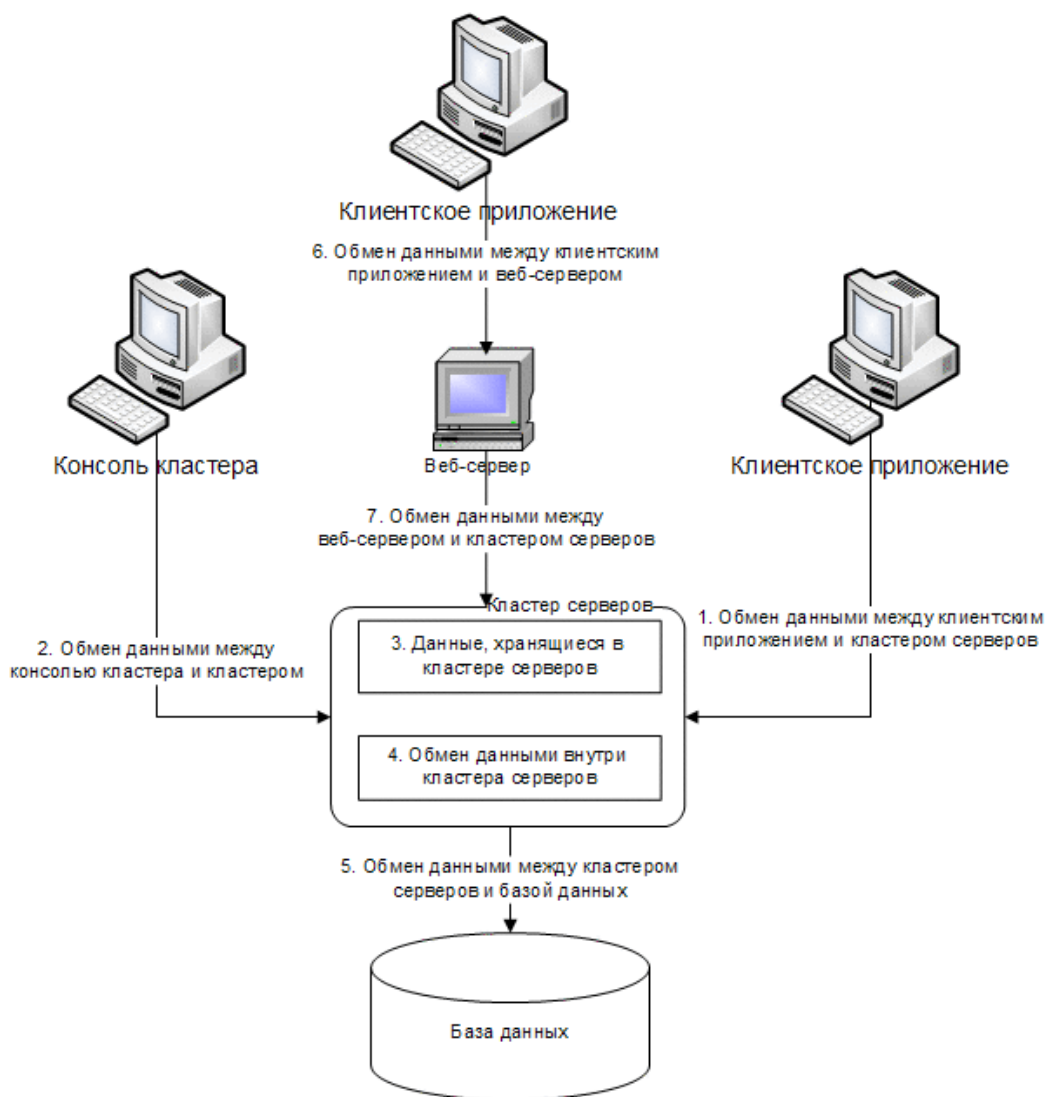


Рис. 11. Общая схема безопасности

Все клиентские приложения и внешние соединения используют данные «1С:Предприятия» только через кластер серверов «1С:Предприятия», выполнив процедуру аутентификации «1С:Предприятия», заключающуюся в том, что перед началом работы с кластером серверов приложение должно сообщить ему имя пользователя системы «1С:Предприятие» и его пароль. Дальнейшая работа с кластером серверов будет возможной, только если пользователь с соответствующим паролем был зарегистрирован в информационной базе.

Данные, с которыми работает кластер серверов «1С:Предприятия», делятся на информационную базу и служебные данные. Таким образом, задача кластера серверов заключается в том, чтобы обеспечить безопасность данных в следующих областях (нумерацию областей см. [рис. 11](#)):

- при обмене между клиентом и кластером серверов (1);
- при обмене между консолью кластера серверов и кластером (2);
- при обмене данными между кластером серверов и веб-сервером (7);
- при хранении служебных данных в кластере серверов (3);
- при обмене данными внутри кластера (4).

Информационная база размещается в базе данных, и ее безопасность, а также безопасность при обмене между кластером серверов и сервером баз данных обеспечивается средствами используемой СУБД (5).

При подключении к информационной базе через веб-сервер безопасность обмена данными между клиентским приложением и веб-сервером обеспечивает используемый веб-сервер (6).

2.1.9.1. Безопасность данных, передаваемых между клиентом и кластером серверов

Безопасность данных, передаваемых между клиентом и кластером серверов, обеспечивается за счет возможности шифрования этих данных. При этом может быть выбран один из трех уровней безопасности:

- **ВЫКЛЮЧЕНО**,
- **ТОЛЬКО СОЕДИНЕНИЕ**,
- **ПОСТОЯННО**.

Уровень **выключено** является самым низким, уровень **постоянно** – самым высоким. При этом используется безопасное соединение по протоколу TCP/IP с шифрованием алгоритмами **RSA** и **Triple DES**.

2.1.9.1.1. Уровень безопасности «постоянно»

Использование уровня безопасности **постоянно** позволяет полностью защитить весь поток данных (как пароли, так и непосредственно данные) между клиентом и кластером серверов.

ВНИМАНИЕ! Следует учитывать, что при этом возможно значительное снижение производительности системы.

В общем виде протокол взаимодействия клиента и кластера серверов представлен на следующем рисунке.

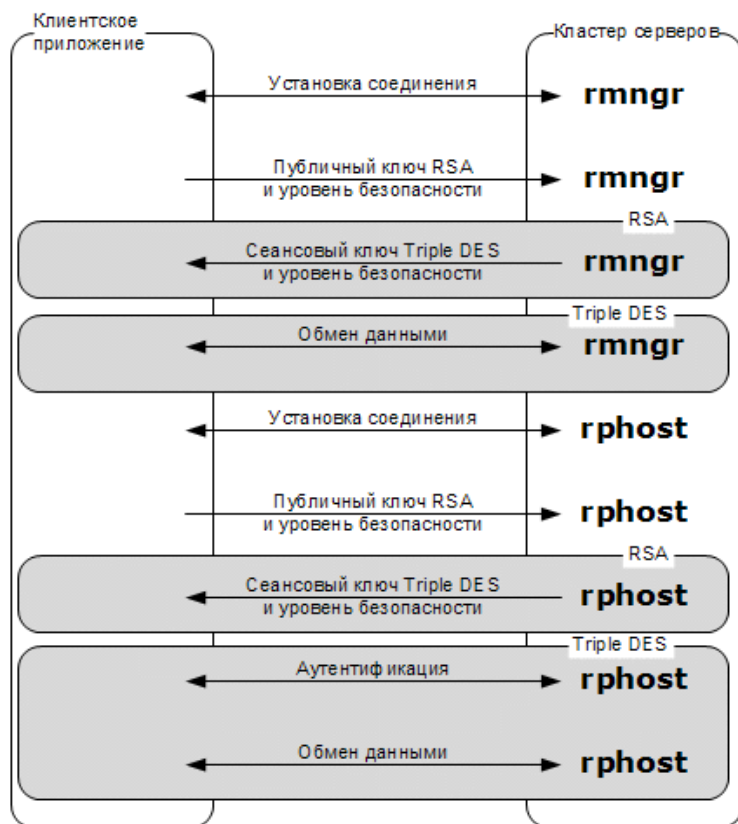


Рис. 12. Уровень безопасности «постоянно»

Протокол взаимодействия одинаков как для менеджера кластера (**rmngr**), так и для рабочего процесса (**rphost**): после установки соединения первый обмен данными выполняется с использованием шифрования по алгоритму **RSA**, дальнейший обмен данными выполняется с использованием шифрования по алгоритму **Triple DES**.

Уровень безопасности задается при создании информационной базы. Эта информация сохраняется как на клиенте (в списке информационных баз), так и в кластере серверов (в реестре сервера). После создания информационной базы, установленный для нее уровень безопасности изменить уже нельзя. Уровень безопасности может быть повышен по инициативе клиентского приложения.

Поэтому после установки соединения клиент генерирует приватный и публичный ключи для шифрования **RSA** и передает в кластер серверов публичный ключ и уровень безопасности, который указан для данной информационной базы на клиенте. Это желаемый уровень безопасности.

Кластер серверов выбирает максимальный уровень безопасности из переданного клиентом и указанного для этой информационной базы в реестре кластера. Это фактический уровень безопасности. Кроме этого, кластер серверов генерирует сеансовый ключ для шифрования **Triple DES** и вместе с фактическим уровнем безопасности передает его клиенту, предварительно зашифровав эти данные с помощью публичного ключа клиента.

Дальнейший обмен данными выполняется в соответствии с фактическим уровнем безопасности, при этом как клиент, так и сервер шифруют передаваемые данные с использованием сеансового ключа по алгоритму **Triple DES**.

2.1.9.1.2. Уровень безопасности «только соединение»

Использование уровня безопасности **только соединение** позволяет частично защитить поток данных (только пароли) между клиентом и кластером серверов. Этот уровень безопасности является компромиссом между безопасностью и производительностью.

С одной стороны, данные информационной базы передаются в открытом виде. Поскольку эти данные составляют основную массу всего потока данных, производительность системы практически не снижается.

С другой стороны, ключевая информация (пароли) передается только в зашифрованном виде. В результате если этот поток данных будет перехвачен, из него можно будет получить лишь незначительную часть данных информационной базы. Т. к. пароли будут недоступны, злоумышленник не сможет аутентифицироваться в информационной базе и получить доступ ко всем данным или выполнить произвольные действия с информационной базой.

В общем виде протокол взаимодействия клиента и кластера серверов представлен на [рис. 13](#).

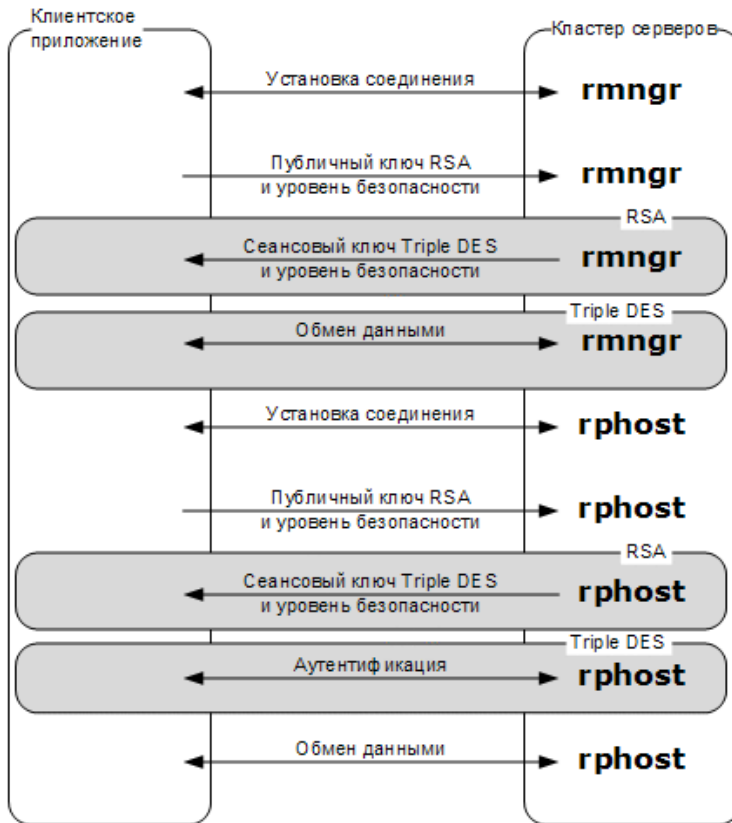


Рис. 13. Уровень безопасности «только соединение»

После установки соединения первый обмен данными выполняется с использованием шифрования по алгоритму **RSA**, затем обмен данными выполняется с использованием шифрования по алгоритму **Triple DES** до окончания процедуры аутентификации. Дальнейший обмен выполняется нешифрованными данными.

2.1.9.1.3. Уровень безопасности «выключено»

Уровень безопасности **выключено** является самым низким и самым производительным. Практически все данные передаются без использования шифрования.

В общем виде протокол взаимодействия клиента и кластера серверов представлен на следующем рисунке.

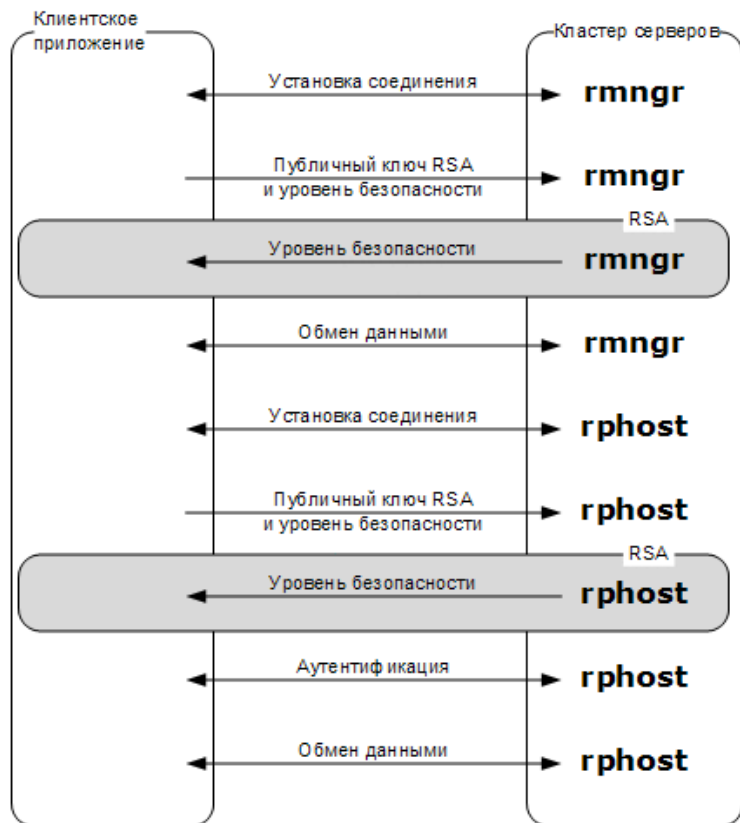


Рис. 14. Уровень безопасности «выключено»

После установки соединения первый обмен данными выполняется с использованием шифрования по алгоритму **RSA**. Дальнейший обмен выполняется нешифрованными данными.

Если клиентское приложение и кластер серверов расположены на одном компьютере и уровень безопасности установлен в значение **выключено**, то шифрование данных между клиентским приложением и кластером серверов вообще не производится.

2.1.9.2. Безопасность данных, передаваемых между консолью кластера серверов и кластером серверов

Безопасность данных, передаваемых между консолью кластера серверов и кластером серверов, обеспечивается также за счет возможности шифрования передаваемых данных. При этом используются те же самые три уровня безопасности: **выключено**, **только соединение** и **постоянно**.

Консоль кластера серверов взаимодействует с агентом сервера (процесс **ragent**). Желаемый уровень безопасности задается при старте агента сервера. Фактический уровень безопасности будет выбран агентом сервера как максимальный из указанного при старте и из уровней безопасности всех кластеров, расположенных на данном центральном сервере. Уровень безопасности кластера задается при его создании (программном или интерактивном).

2.1.9.3. Безопасность данных, хранящихся в кластере серверов

Кластер серверов использует ряд служебных данных, например, список кластеров сервера, реестры кластеров и др. Все служебные данные представляют собой совокупность файлов, которые находятся в двух каталогах:

- каталог данных приложения,
- каталог временных файлов.

Общая идеология работы со служебными данными заключается в том, что доступ к служебным данным кластера серверов должны иметь только менеджер кластера (**rmngr**) и агент сервера (**ragent**). Рабочие процессы (**rphost**) используют служебные данные только через менеджера кластера, поскольку являются потенциально опасными, т. к. в них могут выполняться фрагменты кода конфигураций.

2.1.9.3.1. Безопасность каталога данных приложения

В каталоге данных приложения при установке кластера серверов «1С:Предприятия» создается специальный каталог, предназначенный только для файлов кластера серверов «1С:Предприятия».

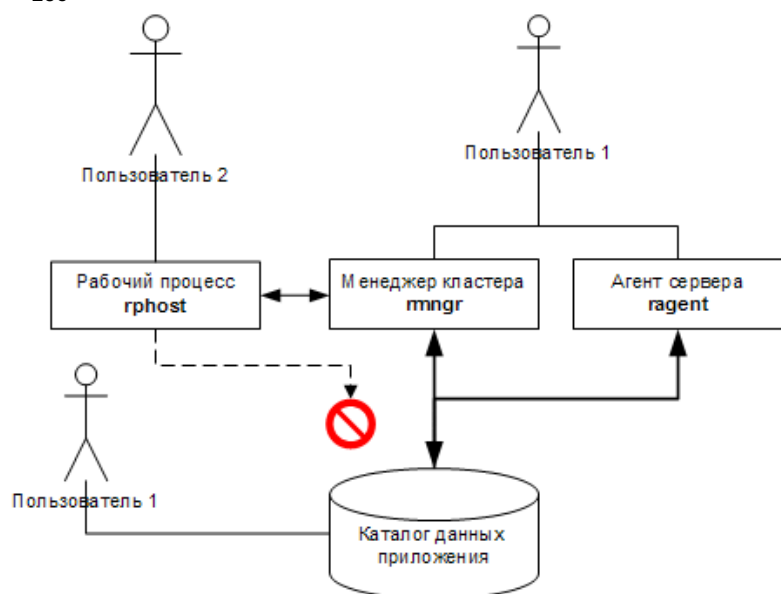


Рис. 15. Каталог данных приложения

Пользователю **USR1CV8**, от имени которого стартует по умолчанию агент сервера, назначаются полные права на этот каталог. Другим пользователям доступ в этот каталог запрещается. Запуск менеджера кластера выполняет агент сервера от имени того же самого пользователя, от которого запущен он сам.

Пользователь **USR1CV8** обладает правами:

- Вход в качестве службы (Log on as a service),
- Вход в качестве пакетного задания (Log on as a batch job).

Пользователь **USR1CV8** должен входить в группы:

- Пользователи журналов производительности (Performance Log Users).

Запуск рабочих процессов также осуществляет агент сервера. По умолчанию рабочий процесс запускается от имени того пользователя, от которого запущен агент сервера. Однако предусмотрена возможность создания дополнительного пользователя операционной системы, от имени которого стартуют только рабочие процессы. Это позволяет предотвратить непосредственный доступ программного кода конфигураций к служебным данным.

Для того чтобы рабочий процесс запускался не от имени того же пользователя, что и агент сервера, в каталоге данных приложений, относящемся к пользователю агента сервера, может быть размещен файл **swpuser.ini** (см. книгу «1С:Предприятие 8.3. Руководство администратора»).

2.1.9.3.2. Безопасность каталога временных файлов

Защита данных из временных файлов выполняется иначе. Поскольку системный каталог временных файлов является общедоступным, то в нем ограничиваются права доступа к каждому временному файлу отдельно.

Для этого при создании временного файла кластером серверов «1С:Предприятия» пользователю **USR1CV8** устанавливаются полные права на создаваемый файл. Другим пользователям доступ к этому файлу запрещается. Таким образом, данные, хранимые во временных файлах, защищаются от несанкционированного доступа.

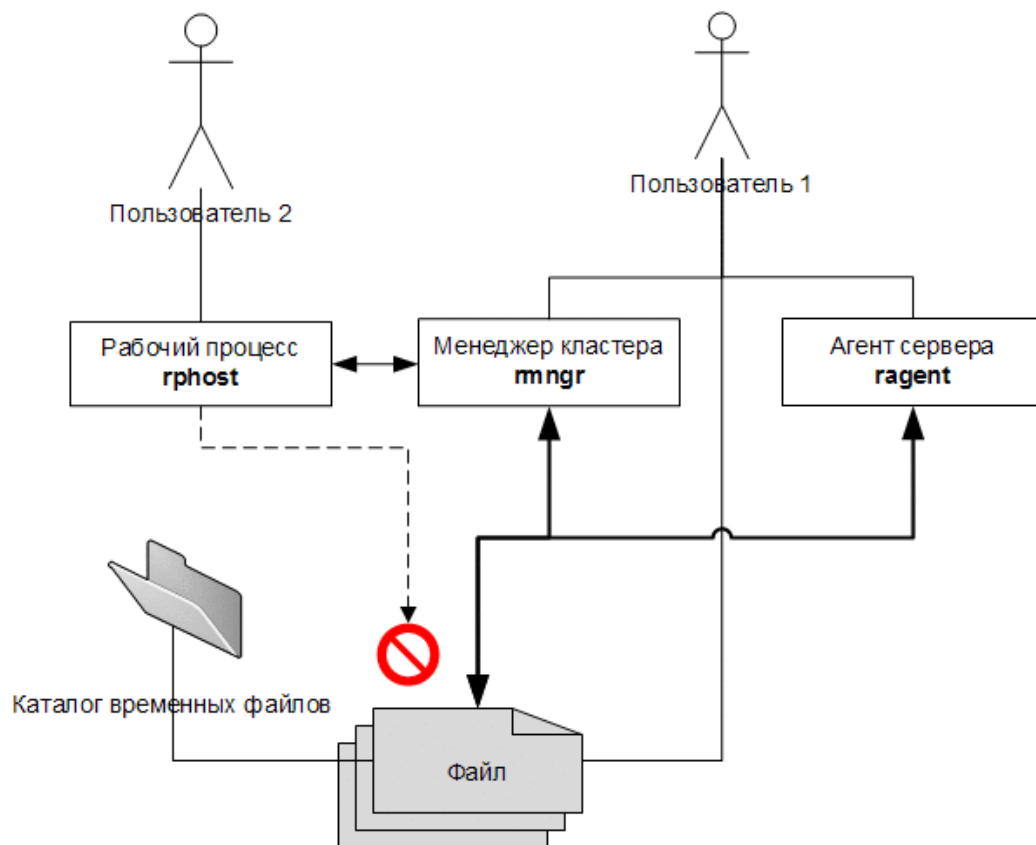


Рис. 16. Ограничение доступа

2.1.9.3.3. Шифрование паролей, хранимых в служебных данных кластера серверов

Пароли администраторов кластера серверов и пароли доступа к информационным базам хранятся в кластере серверов в зашифрованном виде. Для этого используется шифрование алгоритмами [SHA1](#) и [AES128](#):

- [SHA1](#) – с его помощью хранятся пароли, которые проверяет система «1С:Предприятие» (например, пароль администратора кластера, администратора центрального сервера). При этом исходный текст хранимых паролей восстановить нельзя, а можно только проверить совпадение контрольной суммы введенного пароля с хранимой контрольной суммой.
- [AES128](#) – с его помощью хранятся пароли, для которых должен восстанавливаться исходный текст (например, пароли доступа к СУБД).

2.1.9.4. Безопасность данных, передаваемых внутри кластера серверов

Безопасность данных, передаваемых внутри кластера серверов (например, между рабочими процессами и менеджером кластера), обеспечивается за счет возможности шифрования передаваемых данных. При этом используются три уровня безопасности, о которых было сказано выше: [выключено](#), [только соединение](#) и [постоянно](#).

При взаимодействии рабочего процесса с менеджером кластера используется уровень безопасности кластера. При взаимодействии агента сервера с менеджером кластера уровень безопасности выбирается как максимальный из уровня безопасности, с которым запущен агент сервера, и уровня безопасности кластера, обслуживаемого данным менеджером.

2.1.9.5. Безопасность данных, передаваемых между кластером серверов и СУБД

Защита канала между кластером серверов и СУБД осуществляется средствами той СУБД, которая используется. Все поддерживаемые СУБД позволяют настроить свои клиентские компоненты, находящиеся в кластере, так, чтобы трафик между ними и самой СУБД был зашифрован. Все поддерживаемые СУБД могут использовать протокол SSL.

2.1.9.6. Безопасность данных, передаваемых между клиентским приложением и веб-сервером

Для защиты канала между веб-клиентом и веб-сервером или между тонким клиентом и веб-сервером используются криптографические протоколы SSL или TLS.

Поддержку этих протоколов обеспечивает HTTPS-соединение с веб-сервером. Для этого на сервере должен находиться действительный серверный сертификат, который гарантирует клиенту подлинность открытого ключа сервера, используемого для шифрования данных. Также можно использовать клиентские сертификаты, гарантирующие серверу подлинность клиента.

Следует помнить об ограничениях, связанных с операционной системой, под которой работает клиентское приложение:

в клиентском приложении, работающим под ОС Linux, не поддерживается использование клиентских сертификатов из хранилища сертификатов Windows.

2.1.9.7. Безопасность данных, передаваемых между веб-сервером и кластером серверов

Для защиты каналов между кластером серверов и веб-сервером используются алгоритмы шифрования данных, реализуемые системой «1С:Предприятие»: [RSA](#) и [Triple DES](#).

Соединение между кластером и веб-сервером защищается только кластером на основании свойств информационной базы, к которой выполняется подключение (подробнее см. [здесь](#)).

2.1.9.8. Администраторы центрального сервера и администраторы кластера

Администрирование кластера серверов «1С:Предприятия» может выполняться как с использованием аутентификации администраторов, так и без нее.

Если аутентификация не используется, то любой пользователь, подключившийся к центральному серверу кластера, может выполнять все административные действия как с центральным сервером, так и с любым кластером, расположенным на данном сервере. По умолчанию после установки кластера серверов «1С:Предприятия» аутентификация администраторов не используется.

Для ограничения круга пользователей, которые могут выполнять административные действия, создаются списки администраторов центрального сервера (см. [здесь](#)) и списки администраторов каждого из кластеров (см. [здесь](#)), расположенных на данном сервере. Области действий, на которые распространяются права администраторов центрального сервера и администраторов кластеров, не пересекаются.

Аутентификация администратора центрального сервера позволяет пользователю выполнять административные действия с центральным сервером. Но для того, чтобы выполнить какие-либо административные действия с конкретным кластером, требуется аутентификация администратора кластера. В то же время для выполнения административных действий с кластером пользователь должен аутентифицироваться как администратор кластера; дополнительная аутентификация в качестве администратора сервера для этого не требуется.

Механизм аутентификации администраторов центрального сервера/кластера включается системой автоматически, как только в списке администраторов центрального сервера/кластера создается хотя бы один администратор.

При включенном механизме аутентификации пользователь, не аутентифицировавшийся как администратор центрального сервера, может только просмотреть и изменить параметры подключения к центральному серверу в консоли администрирования кластера серверов.

Пользователь, не аутентифицировавшийся в качестве администратора кластера, может только просмотреть свойства кластера. Кроме того, используя средства встроенного языка, такой пользователь может создавать объекты кластера, информационной базы и пр., но не имеет возможности регистрировать их в кластере.

2.1.10. Механизм блокировок информационной базы

Для обеспечения согласованной работы с информационной базой «1С:Предприятие» использует механизм блокировок, который обеспечивает конкурентную работу с тремя составляющими:

- конфигурация,
- информационная база,
- база данных.

Работу этого механизма можно увидеть в утилите администрирования кластера серверов (см. [здесь](#)).

Существуют два вида блокировок: **разделяемая** и **исключительная**. **Разделяемые** блокировки обеспечивают совместную работу нескольких сеансов. **Исключительная** блокировка используется в тех случаях, когда необходимо исключить возможность изменения данных другими сеансами.

С точки зрения администрирования системы важно знать, в каких случаях устанавливается исключительная блокировка (монопольный режим). При этом исключительная блокировка на конфигурацию и информационную базу устанавливается и снимается автоматически самой системой в некоторых случаях (см. ниже). Исключительная блокировка на базу данных может быть установлена или снята как автоматически системой, так и разработчиком при вызове метода встроенного языка `УстановитьМонопольныйРежим()`.

При этом важно понимать, что данные блокировки относятся к механизмам «1С:Предприятия», которые отвечают за доступ к базе данных. Поэтому, например, установка исключительной блокировки базы данных не препятствует доступу к базе данных другим приложениям, не использующим механизмы «1С:Предприятия» для работы с базой данных.

Исключительная блокировка:

- **На конфигурацию** – устанавливается системой при запуске configurator и гарантирует невозможность подключения к одной информационной базе нескольких configuratorов.
- **На информационную базу** – означает, что может существовать только один сеанс любого вида. Блокировка устанавливается:
 - При обновлении конфигурации базы данных;
 - Загрузке информационной базы;
 - Выгрузке информационной базы;
 - Создании начального образа информационной базы;
 - Конвертации информационной базы для новой версии платформы;
 - Тестировании и исправлении.
- **На базу данных** – означает, что может существовать один сеанс вида **Configurator** и только один дополнительный сеанс произвольного вида. Блокировка устанавливается:

- При выполнении метода `УстановитьМонопольныйРежим()` – требуется в тех случаях, когда необходимо выполнить согласованные изменения базы данных, но проводимые изменения не могут быть выполнены в одной транзакции. Например, при массовом удалении объектов в большой информационной базе.
- При групповом проведении документов (только для толстого клиента).

Если в момент попытки перехода в монопольный режим существуют служебные соединения с информационной базой, то такие соединения не препятствуют установке монопольного режима доступа (как к информационной базе, так и к базе данных).

Т. к. сеанс определяет активного пользователя, то наличие сеансов будет мешать переходу в монопольный режим (исключая сеанс консоли кластера). Кроме того, при попытке перехода в монопольный режим происходит неявный разрыв всех соединений с информационной базой, которым не назначен сеанс. Если установлен монопольный режим доступа к информационной базе или базе данных, то новые сеансы не могут быть начаты.

Если существуют веб-серверы, имеющие соединения с данной информационной базой, то при попытке установки монопольного доступа происходит обращение к этим серверам с целью очистки пула соединений.

2.1.11. Статистика загрузки рабочих процессов

Для каждого рабочего процесса кластер серверов вычисляет ряд параметров, которые характеризуют загрузку рабочего процесса. Все эти параметры вычисляются за период времени, равный примерно 10 минутам:

- среднее время реакции кластера серверов;
- среднее время, затраченное кластером серверов;
- среднее время, затраченное СУБД;
- среднее время, затраченное клиентом;
- среднее время, затраченное менеджером блокировок;
- среднее количество клиентских потоков.

Среднее время реакции кластера серверов представляет собой время, которое кластер серверов затратил на обслуживание одного клиентского соединения. Это время складывается из четырех составляющих:

- время, затраченное рабочим процессом;
- время, затраченное СУБД;
- время, затраченное самим клиентом (если управление передавалось обратно на клиента);
- время, затраченное менеджером блокировок.

Обслуживание каждого клиентского соединения рабочий процесс выполняет в отдельном потоке. Таким образом, в некоторый момент времени рабочий процесс может выполнять несколько клиентских потоков, количество которых, как правило, меньше количества соединений (т. к. не все соединения активны постоянно). Среднее количество клиентских потоков как раз и показывает среднее (за сутки) количество потоков, выполняемых рабочим процессом одновременно.

2.1.12. Система мониторинга кластера

При работе кластера серверов необходимо отслеживать работоспособность отдельных элементов кластера и своевременно реагировать на проблемы. Для этого в составе кластера серверов существует подсистема мониторинга кластера. Эта подсистема выполняет регулярный сбор информации о состоянии процессов кластера. Под термином «процесс» в общем случае понимается любой компонент кластера серверов: агент сервера (**ragent**), менеджер сервера (**rmngr**), рабочий процесс (**rphost**). В том случае, если термин «процесс» не может быть однозначно использован в конкретном месте, он будет уточняться. В качестве собираемых (и анализируемых) показателей выступают:

- Проверка соединения с процессом.
- Вычисление доступной производительности рабочего процесса (см. [здесь](#)).
- Проверка объема памяти занимаемой процессом (применимо для менеджера кластера и рабочего процесса).
- Анализ отношения количества ошибок к количеству запросов или, другими словами, среднее, за последние 5 минут, отношение количества событий типа **EXCP** к количеству сообщений типа **CALL** в технологическом журнале.
- Отслеживание рабочих процессов удаленных из реестра кластера.

Некоторые операции (например, проверка объема памяти, занятой процессом) выполняются через соответствующие агенты серверов, тем самым проверяя работоспособность этих агентов.

Проверка производится каждые 10 секунд. Таймаут при проверке соединения составляет 20 секунд. Проверки выполняются последовательно, каждая следующая проверка не начнется до окончания предыдущей проверки. Отрицательные результаты проверки фиксируются в технологическом журнале, с помощью события **ATTN**.

В том случае, если в результате анализа собранных показателей процесс признается проблемным (из анализа исключается показатель доступной производительности), то возможно принудительное завершение проблемного процесса (с формированием дампа аварийного завершения). Принудительное завершение (и формирование дампа аварийного завершения) выполняется в том случае, если в свойства кластера установлен флажок **Принудительно завершать проблемные процессы**. Процессы на дополнительных серверах завершаются с помощью запросов к соответствующим агентам кластера.

Создание дампов аварийного завершения выполняется по общим правилам создания дампов в платформе «1С:Предприятие»:

- Для ОС Windows используются настройки из файла **logcfg.xml**;
- Для ОС Linux используются настройки ОС, а начало записи дампа аварийного завершения выполняется путем отправки процессу сигнала **SIGSEGV**.

Подробнее о настройке формирования дампов аварийного завершения можно прочитать в книге «1С:Предприятие 8.3. Руководство администратора».

Опрос процессов кластера производит только агент центрального сервера. При опросе процессов на дополнительных рабочих серверах используются агенты дополнительных серверов. В случае ошибки соединения с агентом сервера, в технологическом журнале агента центрального сервера появляются сообщения об ошибках.

В процессе работы кластера серверов регулярно выполняются серверные вызовы (отражаемые в технологическом журнале событием **CALL**). Во время исполнения этих вызовов могут происходить различные ошибки (в технологическом журнале отражаются событием **EXCP**). Факт фиксации в технологическом журнале события **EXCP** не означает, что произошла реальная проблема в работе сервера. Например, при выборе рабочего порта из доступного диапазона, каждый занятый порт фиксируется в сообщении **EXCP**, но сама по себе ситуация не является проблемной, т. к. будет выполняться поиск следующего порта и так далее. В общем случае, отношение количества ошибок к количеству серверных вызовов является стабильным значением. Но может возникнуть ситуация, когда из-за внутренних проблем рабочий сервер начинает генерировать количество ошибок, которое превышает среднее количество для данной конфигурации сервера. Для обнаружения таких ситуаций существует свойство **Допустимое отклонение количества ошибок сервера**. Данное свойство указывает, какое отклонение в количестве ошибок (в %) от среднего показателя по рабочим процессам данного кластера серверов является подозрительным и может свидетельствовать о наличии проблем. Если рабочий процесс считает проблемным, то он завершается (с формированием дампа аварийного завершения) и записью события **ATTN** в технологический журнал. Завершение проблемного рабочего процесса выполняется только в том случае, если в свойствах кластера установлен флажок **Принудительно завершать проблемные процессы**. Если значение свойства **Допустимое отклонение количества ошибок сервера** равно 0, то проверка отклонения количества ошибок не выполняется.

Вне зависимости от установленного значения свойства **Допустимое отклонение количества ошибок сервера**, рабочий процесс, который совершает не более 1 ошибки на 100 запросов, считает функционирующим нормально и не признается проблемным.

Рассмотрим пример работы свойства **Допустимое отклонение количества ошибок сервера**. Допустим, на 100 запросов фиксируется, в среднем, 2 ошибки за последние 5 минут. Если свойство **Допустимое отклонение количества ошибок**

сервера установлено в значение 50, то проблемным будет считаться тот рабочий процесс, для которого будет фиксироваться более 3 ошибок на 100 запросов.

Для оценки текущего количества ошибок и вызовов сервера, следует включить запись в технологический журнал событий **EXCP** и **CALL**, затем проработать в системе более 5 минут и любым доступным способом получить количество событий каждого типа для каждого рабочего процесса.

При отслеживании процессов удаленных из реестра кластера, процесс признан проблемным, если он не завершился в течение 20 минут после удаления из реестра кластера.

Описанную систему мониторинга кластера серверов может помочь при решении следующих проблем:

- «Зависание» процесса сервера в памяти процессов компьютера как в процессе работы, так и во время попытки завершения процесса.
- Превышение каким-либо рабочим сервером среднего количества ошибок и, как следствие, нарушение нормального функционирование рабочего процесса.
- Неактуальная информация о памяти, занимаемой рабочим процессом, которая отображается в консоли кластера.
- Повышенное потребление памяти менеджером кластера.

2.1.13. Расположение служебных файлов менеджера кластера

Если при установке системы «1С:Предприятие» был выбран вариант запуска сервера «1С:Предприятия» как сервиса, то первый запуск агента сервера будет выполнен еще в процессе установки системы. При этом сервис будет запущен от имени пользователя, выбранного в диалоге установки системы, но служебные файлы кластера серверов будут расположены в каталоге <каталог установки системы 1С:Предприятие>\srvinfo (в параметрах сервиса будет в явном виде указан ключ запуска **-d**).

Если при установке системы «1С:Предприятие» был выбран вариант запуска сервера как приложения, то запуск сервера в процессе установки системы не выполняется; агента сервера необходимо запустить самостоятельно, после того как установка системы будет закончена. При этом если ключ запуска **-d** указан не будет, служебные файлы кластера серверов будут расположены в каталоге по умолчанию: %USERPROFILE%\Local Settings\Application Data\1C\1Cv8 (%LOCALAPPDATA%\1C\1Cv8 для ОС Windows Vista и старше).

ВНИМАНИЕ! Если однажды на данном центральном сервере уже был создан кластер, то при смене варианта запуска агента сервера (сервис, приложение) или при смене пользователя, от имени которого работает агент сервера, всегда следует заботиться о правильном указании пути к каталогу служебных файлов кластера серверов. Если в процессе запуска агент сервера не обнаружит список кластеров, он создаст новый кластер на данном сервере.

В операционной системе Linux служебные файлы кластера серверов будут расположены в папке /home/usr1cv8/.1cv8/1C/1cv8 (или сокращенный вариант записи – ~/.1cv8/1C/1cv8).

2.2. Особенности работы в ОС Linux

При работе кластера серверов «1С:Предприятия» на компьютерах под управлением операционной системы Linux существуют следующие ограничения:

- Рабочий процесс кластера серверов «1С:Предприятия» не может взаимодействовать с СУБД Microsoft SQL Server.
- Рабочий процесс кластера серверов «1С:Предприятия» не может взаимодействовать с COM-объектами. Допускается компиляция серверных модулей, осуществляющих работу с COM-объектами, однако при попытке выполнения такого модуля будет выдано сообщение об ошибке.
- Аутентификация выполняется на основе Kerberos (Windows версия поддерживает протокол NTLM, который способен, в частности, обеспечить аутентификацию без PDC) и/или имени и пароля пользователя.
- Недоступна функциональность объекта ИнтернетСоединение.

Для использования некоторых возможностей на сервере «1С:Предприятия», запущенного под управлением ОС Linux, необходимо наличие некоторых библиотек. Перечень необходимых библиотек и возможности, при использовании которых эти библиотеки необходимы, см. [здесь](#).

Глава 3. Установка компонентов системы

3.1. Установка сервера «1С:Предприятия»

Система «1С:Предприятие» представляет собой совокупность программных модулей, предназначенных для разработки и использования решений (конфигураций) по ведению учета и автоматизации хозяйственной деятельности предприятий, и конфигурации или совокупности конфигураций.

Программные модули системы «1С:Предприятие» универсальны и способны работать с любой конфигурацией (в рамках имеющегося Лицензионного соглашения).

Одновременно с установкой «1С:Предприятия» также производится установка драйвера защиты от несанкционированного использования.

Программа установки позволяет устанавливать на один компьютер несколько версий «1С:Предприятия», выбирать устанавливаемые компоненты системы, выбирать вариант установки сервера «1С:Предприятия».

3.1.1. Общие сведения о процессе установки

Установка системы «1С:Предприятие» существенно отличается для операционных систем семейства Microsoft Windows (далее ОС Windows) и для операционных систем семейства Linux (далее ОС Linux).

Для ОС Windows установка выполняется с помощью специальной программы установки.

Для ОС Linux такая программа отсутствует, поэтому в соответствующих разделах будут описаны конкретные действия для выполнения той или иной операции. Если на ОС Linux выполняется установка сервера «1С:Предприятия», то после установки сервер не будет запущен.

Перед тем как приступить к установке, следует убедиться, что компьютер не заражен вирусами, а жесткий диск не содержит ошибок и имеется достаточно свободного места для выполнения установки.

ПРИМЕЧАНИЕ. Для установки могут понадобиться дистрибутивные носители операционной системы, установленной на компьютере, а также **права** локального или сетевого **администратора**.

3.1.2. Программа установки для ОС Windows

3.1.2.1. Доступные программы установки

Доступны следующие программы установки:

- **1С:Предприятие 8** – позволяет выполнить установку любого компонента системы, кроме 64-разрядного сервера «1С:Предприятия».
- **1С:Предприятие 8 Тонкий клиент** – позволяет выполнить установку только компонентов, необходимых для доступа к серверу «1С:Предприятия», и собственно тонкий клиент.
- **1С:Предприятие 8 (x86-64)** – позволяет выполнить установку 64-разрядного сервера «1С:Предприятия».

Принципы работы с каждой программой установки одинаковы, поэтому будут представлены общие сведения для программы установки **1С Предприятие 8**.

3.1.2.2. Общие сведения о программе установки

Установку выполняет специальный помощник, переход по страницам которого осуществляется с помощью кнопки **Далее >>**. Для старта помощника следует запустить программу **setup.exe** из каталога выбранного дистрибутива. На каждой странице нужно указать какую-либо необходимую информацию, которая в дальнейшем будет использована для установки системы «1С:Предприятие».

Далее будет приведено краткое описание каждого шага помощника.

ПРИМЕЧАНИЕ. Если запустить программу **setup.exe** с использованием ключа **/S**, то установка будет выполнена в «бесшумном» режиме, по настройкам из файла **1CEStart.cfg** (см. книгу «1С:Предприятие 8.3. Руководство администратора»), а если файл не существует, будет выполнена установка по умолчанию.

3.1.2.2.1. Приветствие

С этого окна начинается работа помощника установки системы «1С:Предприятие».

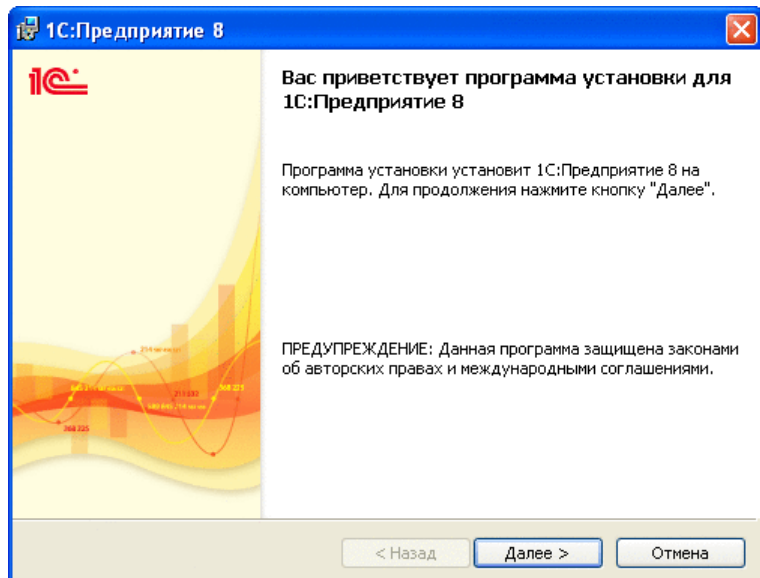


Рис. 17. Приветствие

3.1.2.2.2. Выбор компонентов

На этой странице требуется выбрать те компоненты, которые будут устанавливаться. Перечень компонентов зависит от того, что необходимо установить. Некоторые типовые сценарии установки будут рассмотрены ниже (см. [здесь](#)).

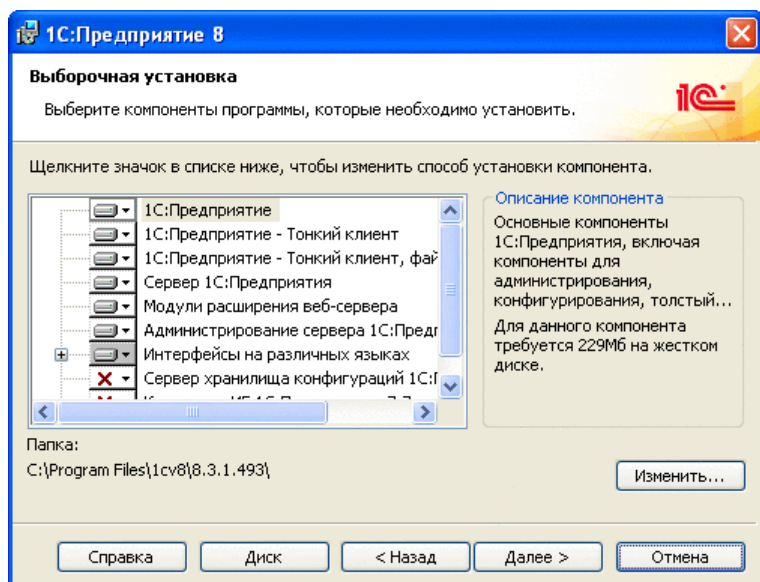


Рис. 18. Выбор компонентов

Если необходимо установить какой-либо компонент, нужно отметить его. Если какой-либо компонент устанавливать нет необходимости, следует запретить его. Для выбора компонента необходимо нажать мышью пиктограмму слева от имени компонента (или клавишу [Пробел](#)). В появившемся меню следует выбрать необходимый пункт (см. [рис. 19](#)).

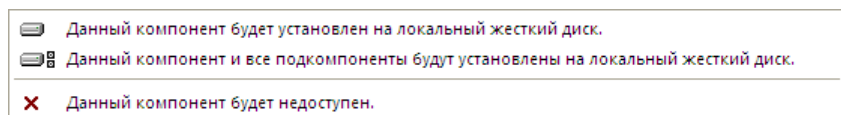


Рис. 19. Меню установки компонента

Устанавливаемые и запрещенные компоненты выглядят на экране так, как показано на [рис. 20](#).

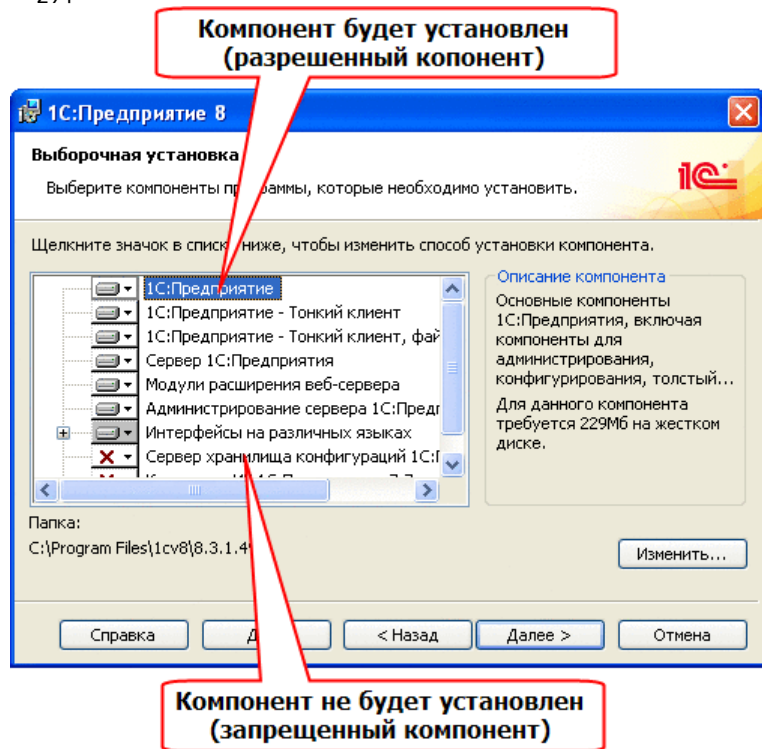


Рис. 20. Разрешенные и запрещенные компоненты

Далее приводится краткое описание устанавливаемых компонентов.

Компонент	Краткое описание
1С:Предприятие	Основные компоненты «1С:Предприятия», включая компоненты для администрирования, конфигурирования, толстый и тонкий клиент
1С:Предприятие – тонкий клиент	Компоненты тонкого клиента только для работы в клиент-серверном варианте
1С:Предприятие – тонкий клиент, файловый вариант	Компоненты тонкого клиента, включая компоненты для работы с файловым вариантом информационной базы
Сервер 1С:Предприятия 8	Компоненты сервера «1С:Предприятия», включая сервер администрирования и утилиту администрирования
Модули расширения веб-сервера	Модули расширения веб-серверов, необходимые для работы веб-клиента и Web-сервисов
Администрирование сервера 1С:Предприятия 8	Дополнительные компоненты для администрирования кластера серверов «1С:Предприятия»
Интерфейсы на разных языках	Пользовательские интерфейсы на различных языках
Сервер хранилища конфигураций 1С:Предприятия 8	Компоненты сервера хранилища конфигураций «1С:Предприятия», в том числе сервер администрирования и утилита администрирования
Конвертор ИБ 1С:Предприятия 7.7	Конвертор информационных баз «1С:Предприятия 7.7»

3.1.2.2.3. Выбор языка интерфейса по умолчанию

На следующем шаге программа установки предложит выбрать язык интерфейса по умолчанию.

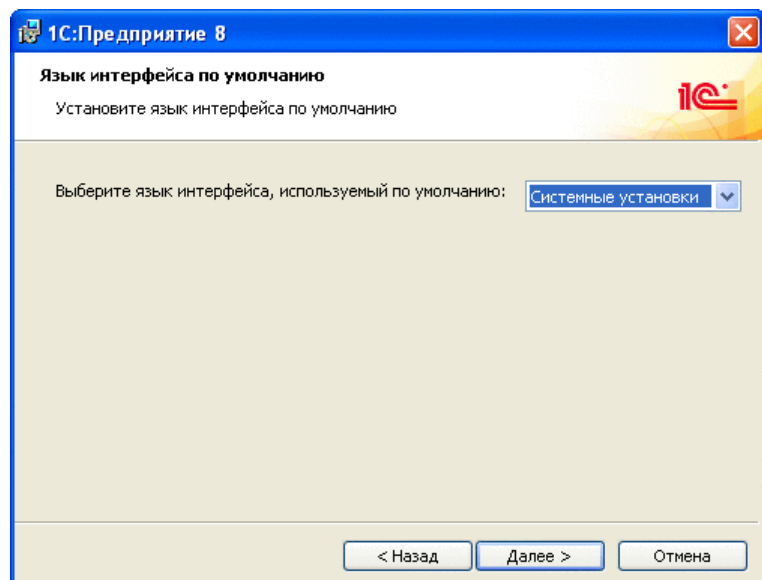


Рис. 21. Выбор языка интерфейса

Следует указать один из языков интерфейса в качестве языка интерфейса по умолчанию.

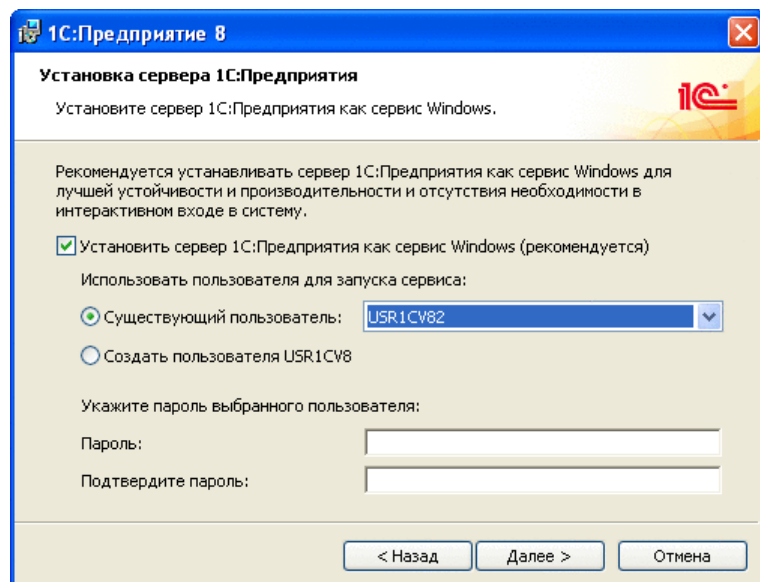
После завершения работы программы установки в каталоге `C:\Program Files\1Cv8\conf` будет создан файл `conf.cfg` (описание файла приводится в книге «1С:Предприятие 8.3. Руководство администратора»), описывающий язык интерфейса по умолчанию.

Если в процессе работы необходимо использование языка интерфейса, отличного от установленного по умолчанию, нужно указать его с помощью ключа командной строки `/L`.

Язык интерфейса	Код языка
Азербайджанский	az
Английский	en
Болгарский	bg
Вьетнамский	vi
Грузинский	kk
Казахский	ka
Китайский	zh
Латышский	lv
Литовский	lt
Немецкий	de
Польский	pl
Румынский	ro
Русский	ru
Турецкий	tr
Украинский	uk

3.1.2.2.4. Установка сервера «1С:Предприятия»

Если для установки выбран компонент [Сервер 1С:Предприятия 8](#), то будет доступна страница помощника, на которой следует выбрать режим установки сервера «1С:Предприятия» и пользователя, под которым будет работать сервер в случае его установки в режиме сервиса ОС Windows.

**Рис. 22. Режим установки сервера «1С:Предприятия»**

ПРИМЕЧАНИЕ. Если выбран режим установки сервера как сервиса, то необходимо указать пароль для выбранного пользователя, иначе программа установки не сможет запустить сервер.

Если на компьютере существует установленная версия «1С:Предприятия» с сервером, установленным в качестве сервиса ОС Windows, то программа установки переустановит сервис.

3.1.2.2.5. Начало установки

После нажатия кнопки [Установить](#) начнется процесс установки:

- создаются необходимые папки;
- копируются файлы выбранных компонентов;
- создаются конфигурационные файлы;

- выполняется регистрация программных компонентов системы;
- на рабочем столе создается ярлык для запуска «1С:Предприятия»;
- выполняется запуск сервера «1С:Предприятия», если выбрана установка сервера как сервиса ОС Windows.

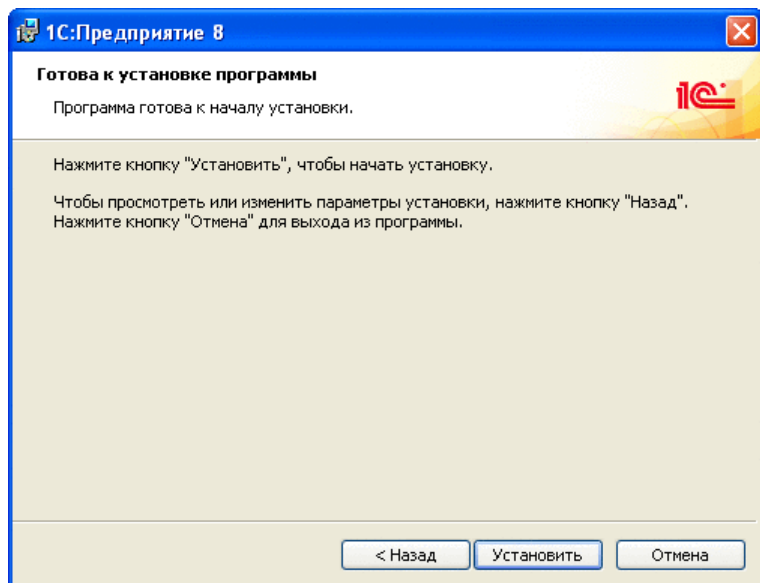


Рис. 23. Начать установку

При этом для каждой версии в компоненте [Установка и удаление программ](#) панели управления Windows будет своя запись вида [1С:Предприятие 8.3 \(8.3.3.100\)](#).

3.1.2.2.6. Установка драйвера защиты

После окончания процесса установки помощник установки предлагает произвести установку драйвера защиты от несанкционированного использования – HASP Device Driver.

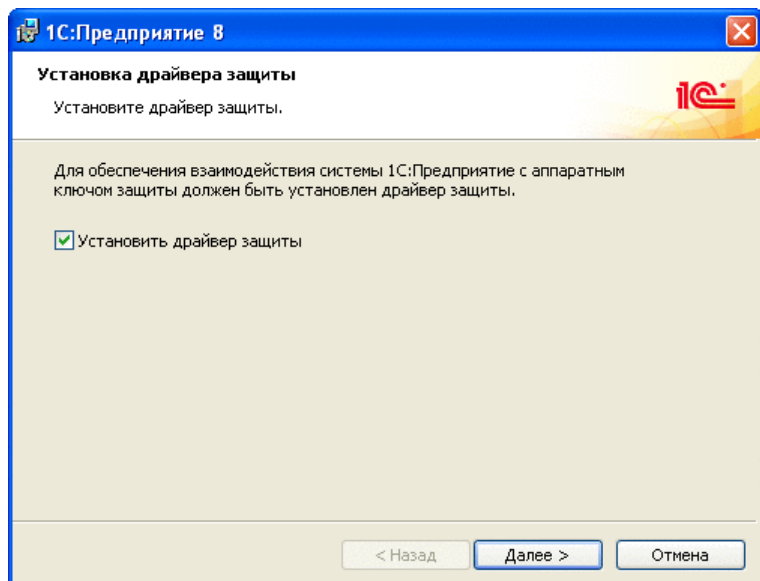


Рис. 24. Установка драйвера защиты

Установка драйвера требуется, если к USB-порту данного компьютера будет присоединяться аппаратный ключ защиты:

- пользователь владеет Лицензионным соглашением на использование системы «1С:Предприятие» на одном рабочем месте;
- пользователь владеет дополнительным Лицензионным соглашением на использование системы «1С:Предприятие» на одном дополнительном рабочем месте;
- пользователь владеет Лицензионным соглашением на использование сервера «1С:Предприятия».

ПРИМЕЧАНИЕ. Рекомендуется выполнять **установку** драйвера защиты **до того**, как ключ защиты будет присоединен к USB-порту компьютера.

Также установку драйвера можно выполнить, используя пункт меню [Пуск – Программы – 1С Предприятие 8 – Дополнительно – Установка HASP Device Driver](#).

3.1.2.2.7. Завершение установки

Если установка завершена успешно, откроется завершающая страница помощника установки. После нажатия на этой странице кнопки

Готово установка будет завершена.

Если поставлен флажок [Открыть файл Readme](#), то будет открыт файл с информацией, которую рекомендуется прочитать перед использованием данной версии системы.

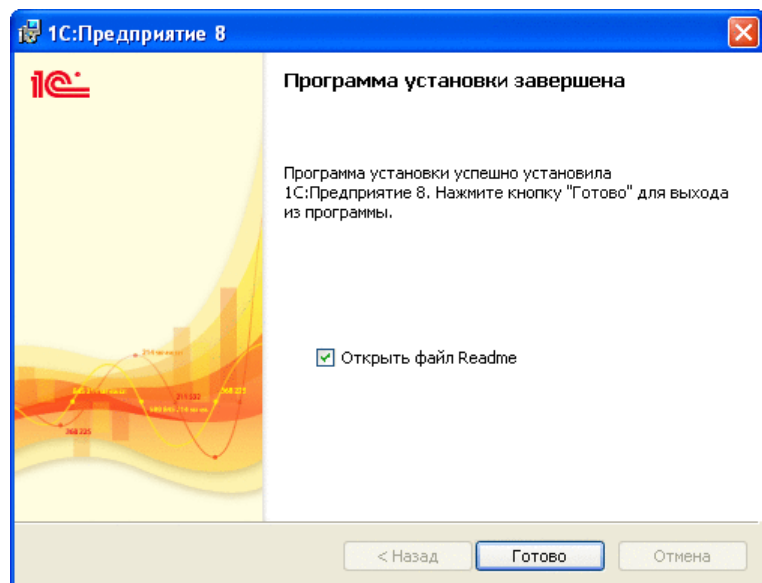


Рис. 25. Завершение установки

3.1.2.3. Особенности регистрации компонент

Программа установки выполняет регистрацию некоторых компонент (консоль кластера, COM-соединение и т. д.). При этом регистрация выполняется следующим образом:

- Консоль кластера. Программа установки выполняет регистрацию «для компьютера». Регистрация с помощью пункта меню [Пуск – Программы – 1С Предприятие 8 – Дополнительно – <Номер версии> – Регистрация утилиты администрирования серверов 1С Предприятия](#) (командный файл `RegMSC.cmd`) выполняется «для пользователя».
- COM-соединение. Программа установки выполняет регистрацию «для компьютера». Регистрацию «для пользователя» можно выполнить с помощью команды:

```
regsvr32 -n -i:user comcntr.dll
```

[Копировать в буфер обмена](#)

3.1.3. Типовые сценарии установки «1С:Предприятия»

В данном разделе приведены рекомендации по выполнению некоторых наиболее распространенных вариантов установки «1С:Предприятия».

3.1.3.1. Для ОС Windows

Данный раздел содержит типовые примеры установки компонент системы программ «1С:Предприятия» для операционной системы Windows.

Для каждого варианта установки будет указан перечень устанавливаемых компонентов и особенности, которые необходимо учесть при установке (если они есть).

3.1.3.1.1. Установка главного сервера кластера

Программа установки копирует на компьютер необходимые файлы и может настроить запуск агента центрального сервера как приложения или как сервиса ОС Windows. Сервис выполняется от лица специального пользователя `USR1CV8`, который также создается в процессе установки кластера серверов «1С:Предприятия» и включается в группу `USERS` или ее локализованные аналоги.

Для установки кластера серверов нужно выбрать следующие компоненты: [Сервер 1С:Предприятия](#), [Администрирование сервера 1С:Предприятия 8](#).

Если выбрана установка сервера «1С:Предприятия» как сервиса Windows (рекомендованный способ установки), то следует выбрать пользователя и ввести для него пароль. Если пароль будет пустым, то программа установки не сможет запустить сервер.

Пользователь выбирается из списка существующих, или создается стандартный локальный `USR1CV8` (при существовании такого пользователя он пересоздается). В последнем случае пользователю `USR1CV8` устанавливается указанный пароль. Пользователь не удаляется при деинсталляции «1С:Предприятия». Пользователю назначаются права: [Вход в систему как сервис \(Log on as a service\)](#), [Вход в систему как пакетное задание \(Log on as a batch job\)](#) и он включается в группу [Пользователи журналов производительности \(Performance Log Users\)](#).

Пользователь и пароль используются только при создании сервера «1С:Предприятия» как сервиса. Пользователю в любом случае даются необходимые права на каталог служебных файлов сервера. Запуск кластера серверов будет выполнен в процессе установки. После завершения установки кластер будет полностью работоспособен.

Если выбран вариант установки сервера как приложения, то после окончания установки необходимо самостоятельно запустить кластер серверов.

ВНИМАНИЕ! В зависимости от варианта установки сервера (сервис или приложение) будут выбраны различные каталоги для размещения файлов сервера. Подробнее про эту особенность см. [здесь](#).

3.1.3.1.2. Добавление сервера в кластер серверов

Данный сценарий используется в том случае, если необходимо добавить в существующий кластер серверов еще один физический сервер (например, для повышения производительности). Допустим, что расширяемый кластер серверов расположен на компьютере **COMP1**, а дополнительный рабочий сервер необходимо установить на компьютер **COMP2**. Тогда для добавления рабочего сервера нужно выполнить следующие действия:

- установить сервер «1С:Предприятия» на компьютер **COMP2** (подробнее см. [здесь](#)).
- Затем подключиться с помощью консоли серверов к кластеру серверов (**COMP1**), в который необходимо добавить сервер (см. [здесь](#)).
- добавить новый рабочий сервер (на компьютере **COMP2**) к кластеру, расположенному на компьютере **COMP1** (см. [здесь](#)).
- для рабочего сервера **COMP2** задать указать требования назначения функциональности (см. [здесь](#)), если они необходимы.

После окончания процесса добавления желательно удалить регистрацию главного сервера кластера на том компьютере, который был добавлен в качестве дополнительного сервера кластера (**COMP2**).

Действия по управлению кластером можно выполнить как с помощью консоли кластера (ссылки на работу с которой даны выше), так и с помощью сервера и утилиты администрирования (см. [здесь](#)).

3.1.3.2. Для ОС Linux

Данный раздел содержит типовые примеры установки сервера «1С:Предприятия» для операционной системы Linux.

Примечание 1. Для 64-разрядного сервера «1С:Предприятия» в именах файлов вместо строки **.i386** должна указываться строка **.x86_64**. Так, файл **1C_Enterprise83-ws-8.3.3-100.i386.rpm** будет иметь имя **1C_Enterprise83-ws-8.3.3-100.x86_64.rpm**.

ПРИМЕЧАНИЕ 2. Для 64-разрядного сервера «1С:Предприятия» (для ОС Linux семейства Debian) в именах файлов вместо строки **_i386** должна указываться строка **_amd64**. Так, файл **1c-enterprise83-ws_8.3.3-100_i386.deb** будет иметь имя **1c-enterprise83-ws_8.3.3-100_amd64.deb**.

ПРИМЕЧАНИЕ 3. Файлы установки для ОС Linux семейства Debian будут иметь имена, которые отличаются от своих rpm-версий. Эти файлы имеют расширение **deb**, а также несколько другое построение собственно имени пакета. Так, файл **1C_Enterprise83-ws-8.3.3-100.i386.rpm** будет иметь имя **1c-enterprise83-ws_8.3.3-100_i386.deb**.

В дальнейшем вся информация будет приводиться для rpm-варианта установочных пакетов. Установка осуществляется с помощью менеджера пакетов используемой операционной системы.

Дистрибутив серверной части «1С:Предприятия» для Linux представлен в виде нескольких rpm-пакетов. В общем случае имена файлов rpm-пакетов имеют вид:

- **1C_Enterprise83-common-8.3.<X>-<Y>.i386.rpm**,
- **1C_Enterprise83-common-nls-8.3.<X>-<Y>.i386.rpm**,
- **1C_Enterprise83-server-8.3.<X>-<Y>.i386.rpm**,
- **1C_Enterprise83-server-nls-8.3.<X>-<Y>.i386.rpm**,
- **1C_Enterprise83-ws-8.3.<X>-<Y>.i386.rpm**,
- **1C_Enterprise83-ws-nls-8.3.<X>-<Y>.i386.rpm**,
- **1C_Enterprise83-crs-8.3.<X>-<Y>.i386.rpm** (только для архитектуры i386).

Где **<X>** и **<Y>** – соответствующие позиции в версии «1С:Предприятия». Например, для «1С:Предприятия» версии 8.3.3.100 упомянутые файлы будут иметь следующие имена:

- **1C_Enterprise83-common-8.3.3-100.i386.rpm**,
- **1C_Enterprise83-common-nls-8.3.3-100.i386.rpm**,
- **1C_Enterprise83-server-8.3.3-100.i386.rpm**,
- **1C_Enterprise83-server-nls-8.3.3-100.i386.rpm**,
- **1C_Enterprise83-ws-8.3.3-100.i386.rpm**,
- **1C_Enterprise83-ws-nls-8.3.3-100.i386.rpm**,
- **1C_Enterprise83-crs-8.3.3-100.i386.rpm**.

Пакеты содержат в себе следующие компоненты:

- **1C_Enterprise83-common** – общие компоненты «1С:Предприятия»;
- **1C_Enterprise83-server** – компоненты сервера «1С:Предприятия»;

• [1C_Enterprise83-ws](#) – адаптер для публикации Web-сервисов «1С:Предприятия» на веб-сервере на основе Apache HTTP Server 2.0 или Apache HTTP Server 2.2;

• [1C_Enterprise83-crs](#) – компоненты сервера хранилища конфигурации «1С:Предприятия» (сервер хранилища является 32-разрядным приложением).

Пакеты, содержащие в названии суффикс «-nls», – это дополнительные национальные ресурсы для соответствующего пакета (кроме русского и английского языков). Например, пакет [1C_Enterprise83-server-nls](#) содержит дополнительные национальные ресурсы для компонентов сервера «1С:Предприятия».

При установке нужно учитывать следующие зависимости между пакетами:

- [1C_Enterprise83-common](#) не имеет зависимостей;
- [1C_Enterprise83-server](#) зависит от [1C_Enterprise83-common](#);
- [1C_Enterprise83-ws](#) зависит от [1C_Enterprise83-common](#);
- [1C_Enterprise83-crs](#) зависит от [1C_Enterprise83-common](#), [1C_Enterprise83-server](#) и [1C_Enterprise83-ws](#);
- пакеты национальных ресурсов зависят от соответствующего компонента.

Поэтому для того, чтобы успешно установить пакет, предварительно нужно установить все пакеты, от которых он зависит. Например, для установки компонентов сервера «1С:Предприятия» необходимо сначала установить пакет [1C_Enterprise83-common](#) и после этого – [1C_Enterprise83-server](#).

3.1.3.2.1. Установка рабочего сервера

Для того чтобы на компьютере можно было развернуть рабочий сервер кластера, необходимо установить следующие пакеты:

- [1C_Enterprise83-common-8.3.<X>-<Y>.i386.rpm](#),
- [1C_Enterprise83-common-nls-8.3.<X>-<Y>.i386.rpm](#),
- [1C_Enterprise83-server-8.3.<X>-<Y>.i386.rpm](#),
- [1C_Enterprise83-server-nls-8.3.<X>-<Y>.i386.rpm](#).

Если предполагается, что компьютер должен использоваться для публикации Web-сервисов «1С:Предприятия», то должен быть установлен пакет [1C_Enterprise83-ws](#). Пакеты [1C_Enterprise83-server](#) и [1C_Enterprise83-ws](#) не зависят друг от друга. Соответственно, они могут быть установлены на одном компьютере как вместе, так и по отдельности.

В процессе установки компонентов сервера «1С:Предприятия» создается пользователь операционной системы с именем `usr1cv8`, под учетной записью которого будут исполняться серверные процессы «1С:Предприятия». Сразу после установки пакета [1C_Enterprise83-server](#) сервер «1С:Предприятия» готов к запуску.

Установка должна выполняться от лица пользователя `root`.

3.1.3.2.2. Добавление сервера в кластер серверов

Данный сценарий используется в том случае, если необходимо добавить в существующий кластер серверов еще один физический сервер (например, для повышения производительности). Допустим, что расширяемый кластер серверов расположен на компьютере `COMP1`, а дополнительный рабочий сервер необходимо установить на компьютере `COMP2`. Тогда для добавления рабочего сервера следует выполнить следующие действия:

- Необходимо установить сервер «1С:Предприятия» на компьютер `COMP2` (подробнее см. [здесь](#)).
- Затем необходимо подключиться с помощью консоли серверов к кластеру серверов (`COMP1`), в который необходимо добавить сервер (см. [здесь](#)).
- Необходимо добавить новый рабочий сервер (на компьютере `COMP2`) к кластеру, расположенному на компьютере `COMP1` (см. [здесь](#)).
- Для рабочего сервера `COMP2` задать указать требования назначения функциональности (см. [здесь](#)), если они необходимы.

После окончания процесса добавления желательно удалить регистрацию главного сервера кластера на том компьютере, который был добавлен в качестве дополнительного сервера кластера (`COMP2`).

Действия по управлению кластером можно выполнить как с помощью консоли кластера (ссылки на работу с которой даны выше), так и с помощью сервера и утилиты администрирования (см. [здесь](#)).

3.1.3.2.3. Работа веб-клиента

Если необходимо использовать веб-клиент для доступа к информационной базе, то на компьютер с веб-сервером нужно установить следующие пакеты:

- [1C_Enterprise83-common-8.3.<X>-<Y>.i386.rpm](#),
- [1C_Enterprise83-common-nls-8.3.<X>-<Y>.i386.rpm](#),
- [1C_Enterprise83-server-8.3.<X>-<Y>.i386.rpm](#),
- [1C_Enterprise83-server-nls-8.3.<X>-<Y>.i386.rpm](#),

- [1C_Enterprise83-ws-8.3.<X>-<Y>.i386.rpm](#),
- [1C_Enterprise83-ws-nls-8.3.<X>-<Y>.i386.rpm](#).

Установка должна выполняться от лица пользователя [root](#).

Публикацию веб-клиента следует выполнять с помощью утилиты командной строки [webinst](#) (описание утилиты приведено в книге «1С:Предприятие 8.3. Руководство администратора»).

3.1.4. Установка и настройка дополнительного программного обеспечения

3.1.4.1. Для ОС Linux

3.1.4.1.1. Настройка Kerberos-аутентификации

В данном разделе описывается настройка Kerberos-аутентификации сервера «1С:Предприятия» для некоторой базовой системы, состоящей из трех компьютеров: контроллера домена, центрального сервера кластера «1С:Предприятия», рабочей станции.

Описание базовой системы

В базовой системе присутствуют следующие компьютеры:

- Контроллер домена Active Directory:
 - имя компьютера (hostname) – [main](#);
 - IP-адрес – [192.168.29.150](#);
 - имя домена – [krb.local](#);
- Центральный сервер кластера «1С:Предприятия»:
 - операционная система: [Fedora 7](#);
 - имя компьютера (hostname) – [srv1c](#);
 - IP-адрес – [192.168.29.151](#);
 - установлена реализация Kerberos от MIT, поддерживающая алгоритм RC4-HMAC (пакет [krb5-workstation](#));
- Рабочая станция.

Настройка контроллера домена

В данном разделе предполагается, что контроллер домена Active Directory настроен и работает. Исходя из этого, для настройки Kerberos-аутентификации нужно выполнить следующие шаги:

1. В DNS-сервере следует «вручную» зарегистрировать все Linux-компьютеры. Регистрация Windows-компьютеров происходит автоматически при включении их в домен. В нашем случае «вручную» необходимо зарегистрировать центральный сервер кластера «1С:Предприятия» (т. к. на нем исполняется Linux-версия сервера), а рабочую станцию включить в домен (зарегистрирована она будет автоматически).
2. После этого следует создать учетную запись пользователя домена, с которой будут ассоциироваться запросы авторизации к серверу «1С:Предприятия». В нашем примере это будет пользователь [usr1cv8](#) с паролем [pass1cv8](#). В свойствах этой учетной записи следует снять флажок [Use DES encryption types with this account](#). Если ваша реализация Kerberos не поддерживает алгоритм шифрования RC4-HMAC, то флажок обязательно нужно поставить.
3. Затем для пользователя домена [usr1cv8](#) следует сгенерировать файл с секретным ключом. Для этого используется утилита [ktpass](#), входящая в состав пакета [Windows Support Tools](#) (его можно найти в подкаталоге [SUPPORT](#) установочного диска Microsoft Windows).

В командной строке необходимо запустить утилиту [ktpass](#). Для используемого примера командная строка должна выглядеть следующим образом:

[Копировать в буфер обмена](#)

```
C:\>ktpass -princ usr1cv8/srv1c.krb.local@KRB.LOCAL -mapuser usr1cv8 -pass pass1cv8 -out usr1cv8.keytab
C:\>
```

Если алгоритм RC4-HMAC не поддерживается, командная строка должна выглядеть так:

[Копировать в буфер обмена](#)

```
C:\>ktpass -crypto DES-CBC-CRC -princ usr1cv8/srv1c.krb.local@KRB.LOCAL -mapuser usr1cv8 -pass pass1cv8 -out usr1cv8.keytab
C:\>
```

В результате будет создан файл [usr1cv8.keytab](#) в текущей директории (в примере – это корень диска [C:](#)), а с пользователем [usr1cv8](#) будет ассоциировано имя участника службы [usr1cv8/srv1c.krb.local](#).

ПРИМЕЧАНИЕ 1. Необходимо обратить внимание, что в случае, если сервер «1С:Предприятия» версии 8.3 будет работать от имени пользователя [usr1cv82](#), то следует использовать имя участника службы [usr1cv82/srv1c.krb.local](#). При этом имя доменного пользователя может оставаться неизменным ([usr1cv8](#) в этом примере).

ПРИМЕЧАНИЕ 2. При генерации файла с секретным ключом (с помощью утилиты [ktpass](#)) следует полное имя компьютера (на котором установлен сервер «1С:Предприятия») в имени участника службы (параметр [-princ](#)) указывать в нижнем регистре. В противном случае работа механизма аутентификации не гарантируется.

В параметре `pass` передается пароль учетной записи пользователя домена `usr1cv8 – pass1cv8`.

В параметре `out` указывается имя файла с ключом. В нашем случае это `usr1cv8.keytab`. Однако, если сервер «1С:Предприятия» версии 8.3 работает от имени пользователя `usr1cv82`, то в качестве имени файла с ключом следует указывать `usr1cv82.keytab`.

Настройка центрального сервера кластера «1С:Предприятия»

В данном разделе предполагается, что кластер серверов «1С:Предприятия» уже установлен и работает на центральном сервере кластера.

Прежде всего, следует указать DNS-сервер для центрального сервера кластера. Это должен быть DNS контроллер домена.

ПРИМЕЧАНИЕ. Процесс настройки зависит от конкретного дистрибутива Linux. В данном примере вручную отредактируем файл `/etc/resolv.conf`, указав в нем IP-адрес контроллера домена.

В результате файл должен содержать следующие строки:

```
nameServer 192.168.29.150
search krb.local
```

[Копировать в буфер обмена](#)

Затем следует проверить работу DNS. Для этого следует выполнить команду `ping`:

```
srvlc:~#ping main -c 1
PING main.krb.local (192.168.29.150)56(84)bytesofdata.
64 bytes from 192.168.29.150: icmp_seq=1 ttl=128 time=0.177 ms
--- main.krb.localpingstatistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.177/0.177/0.177/0.000 ms
srvlc:~#
```

[Копировать в буфер обмена](#)

ВНИМАНИЕ! Для компьютеров, участвующих в процессе аутентификации, не допускается большого расхождения системных часов, так как аутентификационные пакеты (тикеты) имеют ограниченное время действия.

Соответственно, нужно синхронизировать время центрального сервера кластера с контроллером домена. Для этого следует использовать команду `ntpdate`:

```
srvlc:~#ntpdate main
4 Jun 11:51:53 ntpdate[2527]: step time Server 192.168.29.150 offset -56.766439 sec
srvlc:~#
```

[Копировать в буфер обмена](#)

Теперь необходимо выполнить настройку Kerberos. Для этого нужно отредактировать файл `/etc/krb5.conf`. При этом нам понадобится NETBIOS-имя контроллера домена. Оно, как правило, представляет собой имя домена в верхнем регистре. Поэтому в нашем случае NETBIOS-имя будет `KRB.LOCAL`. В результате файл `/etc/krb5.conf` должен выглядеть следующим образом:

```
srvlc:~#cat/etc/krb5.conf
[logging]
Default = FILE:/var/log/krb5libs.log
Kdc = FILE:/var/log/krb5kdc.log
admin_Server = FILE:/var/log/kadmind.log
[libdefaults]
default_realm = KRB.LOCAL
dns_lookup_realm = false
dns_lookup_kdc = false
default_tkt_enctypes = rc4-hmac
default_tgs_enctypes = rc4-hmac
[realms]
KRB.LOCAL = {
kdc=main.krb.local:88
default_domain=krb.local
}
[domain_realm]
krb.local = KRB.LOCAL
.krb.local = KRB.LOCAL
KRB.LOCAL = KRB.LOCAL
.KRB.LOCAL = KRB.LOCAL
[kdc]
Profile = /var/kerberos/krb5kdc/kdc.conf
[appdefaults]
Pam = {
Debug = true
ticket_lifetime = 36000
renew_lifetime = 36000
forwardable = false
krb4_convert = false
}
srvlc:~#
```

[Копировать в буфер обмена](#)

Если алгоритм RC4-НМАС не поддерживается, то значения параметров `default_tkt_enctypes` и `default_tgs_enctypes` должны быть следующими:

```
...
default_tkt_enctypes = des-cbc-crc des-cbc-md5
default_tgs_enctypes = des-cbc-crc des-cbc-md5
...
```

[Копировать в буфер обмена](#)

Теперь нужно проверить работу системы аутентификации. Для этого следует выполнить команду `kinit <имя>`, где имя – это имя произвольного пользователя, зарегистрированного в домене `krb.local`. В примере это будет имя `user`. Далее вводится пароль этого пользователя, и завершается команда нажатием кнопки `Enter`. Если после этого программа не выдаст никаких сообщений, значит все выполнено правильно.

Убедиться в этом можно с помощью команды `klist`. Как видно далее, мы получили от KDC (Key Distribution Center – центр распределения ключей, эту функцию выполняет контроллер домена) так называемый `ticket-granting ticket`. После этого следует с помощью команды `kdestroy` очистить локальный кеш тикетов, чтобы вернуться в исходное состояние.

[Копировать в буфер обмена](#)

```
srvlc:~#kinit user
Password for user@KRB.LOCAL:
srvlc:~#klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: user@KRB.LOCAL
Valid starting Expires Service principal
06/04/08 11:29:21 06/04/08 21:28:28 krbtgt/KRB.LOCAL@KRB.LOCAL
renew until 06/05/08 11:29:21
Kerberos 4 ticket cache: /tmp/tkt0
Klist : You have no tickets cached
srvlc:~#kdestroy
srvlc:~#
```

Далее любым способом следует передать файл с секретным ключом `usr1cv8.keytab`, полученный во время настройки контроллера домена, на центральный сервер кластера «1С:Предприятия». Этот файл следует скопировать в директорию, где установлен сервер «1С:Предприятия» (по умолчанию это `/opt/1C/v8.3/i386` или `/opt/1C/v8.3/x86_64` для 64-разрядной версии сервера), и установить права и владельца файла, как показано ниже:

[Копировать в буфер обмена](#)

```
srvlc:~#cd /opt/1C/v8.3/i386
srvlc:i386# chown usrlcv8:grp1cv8 usrlcv8.keytab
srvlc:i386# chmod 600 usrlcv8.keytab
srvlc:i386#
```

При желании файл можно разместить в любом другом месте, нужно только изменить переменную `SRV1CV8_KEYTAB` в конфигурационном файле, чтобы она указывала на новое местоположение файла с секретным ключом.

После этого с помощью команды `klist` необходимо проверить, что все настройки выполнены правильно. Для этого следует выполнить команду:

[Копировать в буфер обмена](#)

```
srvlc:~#klist -e -k -t /opt/1C/v8.3/i386/usrlcv8.keytab
```

Для используемого примера результат выполнения команды должен выглядеть следующим образом:

[Копировать в буфер обмена](#)

```
Keytab name: FILE:/opt/1C/v8.3/i386/usrlcv8.keytab
KVNO Principal
-----
13 usrlcv8/srvlc.krb.local@KRB.LOCAL (ArcFour with HMAC/md5)
```

Если алгоритм RC4-HMAC не поддерживается, результат выполнения команды будет выглядеть следующим образом:

[Копировать в буфер обмена](#)

```
Keytab name:FILE:/opt/1C/v8.3/i386/usrlcv8.keytab
KVNO Principal
-----
13 usrlcv8/srvlc.krb.local@KRB.LOCAL (DES cbc mode with RSA-MD5)
```

Видно, что файл с секретным ключом содержит именно то, что нам нужно (в колонке `Principal` указано то самое имя службы, которое мы задавали при создании файла с секретным ключом, и правильный алгоритм шифрования (`ArcFour with HMAC/md5` для RC4-HMAC или `DES cbc mode with RSA-MD5` для DES)).

Далее следует проверить возможность работы `Kerberos` без пароля с использованием секретного ключа. С помощью команды `kinit` укажем, что нужно использовать аутентификационную информацию из файла (в нашем случае `/opt/1C/v8.3/i386/usrlcv8.keytab` или `/opt/1C/v8.3/x86_64/usrlcv8.keytab` для 64-разрядной версии сервера) и прочитать оттуда ключ для сервиса `usr1cv8/srv1c.krb.local@KRB.LOCAL`. В результате программа `kinit` должна отработать без каких-либо сообщений, не спрашивать никаких паролей и вернуть управление обратно в командную строку:

[Копировать в буфер обмена](#)

```
srvlc:~#kinit -k -t /opt/1C/v8.3/i386/usrlcv8.keytab usrlcv8/srvlc.krb.local@KRB.LOCAL
srvlc:~#
```

Теперь рассмотрим результаты работы с помощью команды `klist`. В случае успеха должна быть выведена информация, аналогичная следующей:

[Копировать в буфер обмена](#)

```
srvlc:~#klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: usrlcv8/srvlc.krb.local@KRB.LOCAL
Valid starting Expires Service principal
06/04/08 11:44:54 06/04/08 21:43:58 krbtgt/KRB.LOCAL@KRB.LOCAL
renew until 06/05/08 11:44:54
Kerberos 4 ticket cache: /tmp/tkt0
klist: You have no tickets cached
srvlc:~#
```

Если что-то настроено не так, то эта команда выведет следующее:

```

srvlc:~#klist
klist: No credential cache found (ticket cache FILE:/tmp/krb5cc_1000)
Kerberos 4 ticket cache: /tmp/tkt1000
klist: You have no tickets cached
srvlc:~#

```

Если проверка работоспособности прошла успешно, это значит, что с данного момента сервер кластера «1С:Предприятия» способен обрабатывать запросы на аутентификацию. При этом перезапуск сервера не требуется, кроме того случая, когда в конфигурационном файле было изменено место расположения файла с секретным ключом.

3.2. Установка серверов баз данных

В качестве сервера баз данных «1С:Предприятия» могут использоваться:

- IBM DB2 (для Windows и Linux);
- Microsoft SQL Server (только для Windows);
- Oracle Database (для Windows и Linux);
- PostgreSQL (для Windows и Linux).

3.2.1. Установка IBM DB2

Установка сервера баз данных производится с дистрибутивов IBM DB2.

Версии IBM DB2, поддерживаемые системой «1С:Предприятие», опубликованы на сайте: <http://v8.1c.ru/requirements/>.

Чтобы облегчить конфигурирование IBM DB2 для работы с платформой «1С:Предприятие», в IBM DB2 введено значение 1C для групповой переменной реестра DB2_WORKLOAD.

Если задать DB2_WORKLOAD=1C, то IBM DB2 автоматически сконфигурирует все необходимые значения переменных реестра для оптимизации работы DB2 с платформой «1С:Предприятие».

Значение переменной реестра задается при помощи команды:

```
db2set DB2_WORKLOAD=1C
```

Копировать в буфер обмена

После установки значения 1C для групповой переменной реестра DB2_WORKLOAD необходимо перезапустить сервер баз данных.

ПРИМЕЧАНИЕ 1. Более подробно с предназначением переменных реестра профиля IBM DB2 можно ознакомиться в документации по IBM DB2 по адресу:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/c0004934.htm>.

ПРИМЕЧАНИЕ 2. Более подробно описание синтаксиса команды db2set приведено в документации по IBM DB2 по адресу:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/c0007340.htm>.

Чтобы узнать, какие значения будут использоваться при установленном DB2_WORKLOAD=1C, можно использовать команду:

```
db2set -gd DB2_WORKLOAD=1C
```

Копировать в буфер обмена

Данная команда отобразит список переменных реестра IBM DB2 и их значений, соответствующих групповой переменной реестра DB2_WORKLOAD=1C.

Если сервер «1С:Предприятия» запущен как сервис, необходимо выполнить следующие действия:

- включить пользователя, от имени которого запускается сервер «1С:Предприятия» (по умолчанию USR1CV8), в группу DB2ADMNS;
- для используемой копии DB2 установить параметр SYSADM_GROUP в значение DB2ADMNS.

Информацию по установке можно получить в документации к серверу:

- IBM DB2 v9.1: <http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.doc/doc/t0024287.htm> (на английском языке).
- IBM DB2 v9.5: <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.qb.server.doc/doc/r0025127.html>.
- IBM DB2 v9.7: <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.qb.server.doc/doc/r0025127.html>.
- IBM DB2 v10.1: <http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.qb.server.doc/doc/r0025127.html>.

3.2.2. Установка Microsoft SQL Server

Установка сервера баз данных производится с дистрибутивных носителей Microsoft SQL Server.

ПРИМЕЧАНИЕ. Если выбран Microsoft SQL Server 2000, то для безошибочной работы «1С:Предприятия» требуется, чтобы для Microsoft SQL Server 2000 был установлен Service Pack 2 (рекомендуется Service Pack 4).

Пользователь, от имени которого сервер «1С:Предприятия» осуществляет доступ к MS SQL Server, должен быть членом фиксированной серверной роли `processadmin` или `sysadmin`.

Информацию по установке можно получить в документации к серверу:

- Версия Microsoft SQL Server 2005:
[http://msdn.microsoft.com/ru-ru/library/ms143516\(SQL_90\).aspx](http://msdn.microsoft.com/ru-ru/library/ms143516(SQL_90).aspx).
- Версия Microsoft SQL Server 2008:
[http://msdn.microsoft.com/ru-ru/library/bb500469\(v=sql.100\).aspx](http://msdn.microsoft.com/ru-ru/library/bb500469(v=sql.100).aspx).
- Версия Microsoft SQL Server 2008 R2:
[http://msdn.microsoft.com/ru-ru/library/bb500469\(v=sql.105\).aspx](http://msdn.microsoft.com/ru-ru/library/bb500469(v=sql.105).aspx).
- Версия Microsoft SQL Server 2012:
[http://msdn.microsoft.com/ru-ru/library/bb500469\(v=sql.11\).aspx](http://msdn.microsoft.com/ru-ru/library/bb500469(v=sql.11).aspx).
- Версия Microsoft SQL Server 2014:
[http://msdn.microsoft.com/ru-ru/library/bb500469\(v=sql.120\).aspx](http://msdn.microsoft.com/ru-ru/library/bb500469(v=sql.120).aspx).

3.2.3. Установка Oracle Database

Установка сервера баз данных производится с дистрибутивов Oracle Database.

Версии Oracle Database, поддерживаемые системой «1С:Предприятие», опубликованы на сайте: <http://v8.1c.ru/requirements/>.

Информацию по установке можно получить в документации к серверу:

- Oracle Database 10g Release 2: http://download.oracle.com/docs/cd/B19306_01/em.102/b16227/toc.htm (на английском языке).
- Oracle Database 11g Release 1: http://download.oracle.com/docs/cd/B28359_01/em.111/b31207/toc.htm (на английском языке).
- Oracle Database 11g Release 2: http://docs.oracle.com/cd/E11882_01/em.112/e12255/toc.htm (на английском языке).

Серверу баз данных в терминах «1С:Предприятия» соответствует понятие **DATABASE** в терминах Oracle Database. Базе данных в терминах «1С:Предприятия» соответствует **схема данных** в терминах Oracle Database. При создании информационной базы данных в «1С:Предприятии» в Oracle Database создается пользователь и его схема данных.

«1С:Предприятие 8» использует при работе с Oracle Database следующие **tablespace**:

- для данных – **V81C_DATA**;
- для индексов – **V81C_INDEX**;
- для LOB – **V81C_LOB**.

Если **tablespace** с такими именами существуют, то они будут использованы; если нет, то будут созданы при создании информационной базы, при этом **datafiles** будут иметь пути по умолчанию.

При создании **DATABASE** необходимо для данной **DATABASE** установить параметр **CHARACTER SET** в значение **AL32UTF8**.

До начала использования сервера Oracle Database с «1С:Предприятием» необходимо выполнить настройку **multilingual collation** для сервера баз данных. Для этого нужно выполнить следующее:

- Скопировать файл **lx327c6.nlt** из каталога **Additional\OracleDatabase** дистрибутивного диска «1С:Предприятия» в пустой каталог на жестком диске компьютера, на котором установлен Oracle Database.
- Запустить **Oracle Locale Builder (lbuilder)**.
- Запустить генерацию nls-файлов (**Tools – Generate NLB**), указав папку с файлом **lx327c6.nlt**.
- Остановить все сервисы Oracle Database, запущенные из домашнего каталога Oracle Database (**ORACLE_HOME**).
- Создать резервную копию файлов **lx0boot.nlb** и **lx1boot.nlb** из папки **ORACLE_HOME/nls/data**.
- Скопировать файлы **lx1boot.nlb** и **lx327c6.nlb** из папки, где они были созданы утилитой **Oracle Locale Builder** в **ORACLE_HOME/nls/data**. В процессе копирования утвердительно ответить на предложение перезаписать файл **lx1boot.nlb**.
- Запустить сервисы Oracle Database.

3.2.4. Установка PostgreSQL

Установка сервера баз данных производится с дистрибутивных носителей «1С:Предприятия». СУБД PostgreSQL может функционировать как под управлением операционной системы Windows, так и под управлением операционной системы Linux.

При инициализации кластера базы данных следует указывать кодировку **UTF-8**.

«1С:Предприятие 8» поддерживает использование следующих табличных пространств (**tablespace**) при работе с PostgreSQL:

- для данных – [v81c_data](#);
- для индексов – [v81c_index](#).

Если табличное пространство с такими именами существуют, то они будут использованы; если нет, то рекомендуется их создать:

```
create tablespace v81c_index location <Имя папки>;
create tablespace v81c_data location <Имя папки>;
```

[Копировать в буфер обмена](#)

Где **<Имя папки>** – полный путь к каталогу, в котором будут располагаться файлы создаваемого табличного пространства.

В случае отсутствия указанных табличных пространств, будет использоваться табличное пространство с именем [pg_default](#).

Версии PostgreSQL, поддерживаемые системой «1С:Предприятие», опубликованы на сайте: <http://v8.1c.ru/requirements/>.

Для совместной работы с системой «1С:Предприятие» необходима модифицированная версия PostgreSQL. Получить эту версию можно на диске ИТС или по адресу: <http://users.v8.1c.ru/AddComp.aspx> (необходим доступ к разделу технической поддержки).

«1С:Предприятие» может работать с кластером, созданным с использованием следующих версий СУБД PostgreSQL:

- PostgreSQL 8.1.5-X.1С – 9.1.2-1.1С;
- PostgreSQL 9.2.1-2.1С;
- Стандартной версией PostgreSQL при условии совпадения национальных установок (locale) кластера и создаваемой информационной базы.

Изменение версии «1С:Предприятия» (как увеличение, так и уменьшение версии), без смены версии PostgreSQL, не приводит к необходимости выполнять какие-либо действия.

Информационные базы, созданные с использованием PostgreSQL версии 9.2.1-2.1С и выше невозможно использовать с более ранними версиями «1С:Предприятия» (8.3.2 и ранее). Не следует создавать информационные базы с использованием PostgreSQL версии 9.2.1-2.1С, если есть необходимость работы с такой информационной базой с помощью более ранних версий платформы.

3.2.4.1. Установка PostgreSQL для Windows

Для установки версии необходимо запустить файл [setup.bat](#) и далее следовать указаниям программы установки.

3.2.4.2. Установка PostgreSQL для Linux

Дистрибутив модифицированной версии PostgreSQL состоит из 11 пакетов. Необходимо установить перечисленные пакеты в указанном порядке:

- [postgresql-libs-8.3.<X>-<Y>.1С.i386.rpm](#),
- [postgresql-8.3.<X>-<Y>.1С.i386.rpm](#),
- [postgresql-Server-8.3.<X>-<Y>.1С.i386.rpm](#),
- [postgresql-contrib-8.3.<X>-<Y>.1С.i386.rpm](#).

Где **<X>** и **<Y>** – соответствующие позиции в версии PostgreSQL.

После этого в системе появится пользователь [postgres](#), будет создан скрипт [/etc/init.d/postgresql](#) для старта и остановки СУБД.

Далее нужно установить желаемое значение переменной [LANG](#) и запустить [/etc/init.d/postgresql](#) для первичного создания базы. Это может быть выполнено командой:

```
LANG=ru_RU.utf-8 /etc/init.d/postgresql start
```

[Копировать в буфер обмена](#)

В результате будет создана база, размещенная в каталоге [/var/lib/pgsql/data](#). Все вышеперечисленные действия должны выполняться от лица пользователя [root](#).

Глава 4. Запуск компонентов системы

При установке «1С:Предприятия» в меню **Пуск – Программы** будет создана группа **1С Предприятие 8**, в составе которой сформируется структура меню, аналогичная представленной на [рис. 26](#).

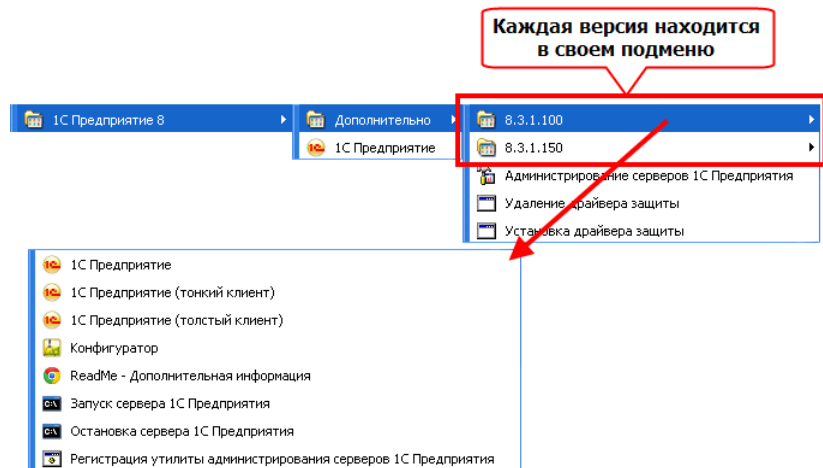


Рис. 26. Структура меню

Пункты	Назначение
1С Предприятие	Вызов программы запуска (1CEStart.exe)
8.3.3.100	Папки, содержащие ссылки на запуск компонентов системы конкретной версии (в данном случае установлены две версии: 8.3.3.100 и 8.3.3.150)
8.3.3.150	
Установка драйвера защиты	Запуск установки драйвера защиты
Удаление драйвера защиты	Запуск удаления драйвера защиты
1С Предприятие (тонкий клиент)	Запуск системы в режиме тонкого клиента «1С:Предприятия»
1С Предприятие (толстый клиент)	Запуск системы в режиме толстого клиента «1С:Предприятия»
Конфигуратор	Запуск системы в режиме Конфигуратор
ReadMe – Дополнительная информация	Дополнительная информация, не вошедшая в документацию
Конвертер ИБ 1С Предприятия 7.7	Программа для конвертации информационных баз в формате «1С:Предприятия 7.7»
Администрирование серверов 1С Предприятия	Утилита администрирования кластера серверов (если были установлены компоненты доступа к кластеру серверов «1С:Предприятия»)
Запуск сервера 1С Предприятия	Запуск сервера «1С:Предприятия», работающего как служба ОС Windows (если при установке сервера был установлен флажок Установить сервер 1С:Предприятия 8 как сервис Windows) или как приложения (если при установке сервера был сброшен флажок Установить сервер 1С:Предприятия 8 как сервис Windows). Остановка сервера в этом случае выполняется как закрытие обычного приложения
Остановка сервера 1С Предприятия	Остановка сервера «1С:Предприятия», работающего как служба ОС Windows (если при установке сервера был установлен флажок Установить сервер 1С:Предприятия 8 как сервис Windows)
Регистрация утилиты администрирования серверов 1С Предприятия	Выполняет регистрацию утилиты администрирования серверов «1С:Предприятие» (radmin.dll) конкретной версии, после чего можно подключаться к серверам этой версии с помощью утилиты администрирования

4.1. Запуск агента сервера

Для того чтобы выполнить запуск кластера серверов «1С:Предприятия», следует запустить агент сервера ([agent](#)). Все дальнейшие действия будут выполнены системой автоматически. При запуске агент сервера выполняет поиск списка кластеров, зарегистрированных на данном компьютере.

Если список кластеров обнаружен, то агент сервера запускает указанные менеджеры кластеров. С их помощью он получает информацию о рабочих процессах, которые должны быть запущены в каждом из кластеров, и выполняет их запуск самостоятельно или с помощью агентов других рабочих серверов кластера.

Если список кластеров не обнаружен, агент сервера создает кластер по умолчанию. Кластер по умолчанию имеет следующие характеристики:

- номер сетевого порта – **1541**;
- диапазон сетевых портов – **1560:1591**;
- поддержка многих рабочих процессов выключена;
- один рабочий процесс, номер порта устанавливается из указанного диапазона.

4.1.1. Для ОС Windows

4.1.1.1. Запуск как приложения

Агент сервера может быть запущен как приложение. Для этого нужно выполнить следующую команду:

```
agent.exe -debug
-port <порт> -regport <порт> -range <диапазоны>
-seclev <уровень> -d <каталог>
```

[Копировать в буфер обмена](#)

ВНИМАНИЕ! Название и значение параметра должны разделяться символом «пробел».

В команде запуска могут использоваться следующие ключи:

```
-port <порт>
```

Номер сетевого порта агента сервера ([agent](#)). Этот порт используется консолью кластера для обращения к центральному серверу. Порт агента кластера также указывается в качестве сетевого порта рабочего сервера. Значение по умолчанию: **1540**.

`-regport <порт>`

Номер сетевого порта главного менеджера кластера (**rmngr**), создаваемого по умолчанию при первом запуске **ragent**. Значение по умолчанию: **1541**.

`-secllev <уровень>`

Уровень безопасности процесса агента кластера. Определяет уровень безопасности соединений, устанавливаемых с процессом **ragent**. Уровень может принимать значения:

- **0** (по умолчанию) – соединения незащищенные;
- **1** – защищенные соединения только на время выполнения аутентификации пользователей;
- **2** – постоянно защищенные соединения.

`-range <диапазоны>`

Диапазоны сетевых портов для динамического выбора. Из них выбираются служебные порты процессов кластера при невозможности их выбора из настроек соответствующего рабочего сервера. Значение по умолчанию: **1560:1591**. Примеры значений диапазонов: **4549:4567, 7072:7790**.

`-debug`

Запуск кластера серверов в режиме отладки конфигураций.

`-d <каталог>`

Каталог, в котором будут расположены (или располагаются) служебные файлы кластера серверов (в том числе список кластеров и список информационных баз кластера). Если параметр не указан, используется каталог по умолчанию: **%USERPROFILE%\Local Settings\Application Data\1C\1Cv8** (**%LOCALAPPDATA%\1C\1Cv8** для ОС Windows Vista и старше). Если путь к каталогу содержит пробелы, путь нужно заключать в кавычки, например:

`-d "c:\Server data\cluster 2"`

[Копировать в буфер обмена](#)

ПРИМЕЧАНИЕ. Имя каталога не должно завершаться символом **"\"**, если оно заключено в кавычки. Правильно: **"c:\my path"**, неправильно: **"c:\my path\"**.

Подробнее об уровне безопасности соединения см. [здесь](#).

Остановка агента сервера, запущенного как приложение, выполняется нажатием клавиш **Ctrl + C**.

4.1.1.2. Запуск как сервиса

Если при установке кластера серверов был выбран вариант запуска агента центрального сервера как сервиса, то данный сервис будет запущен автоматически, в процессе установки, и также будет запускаться при старте операционной системы.

Если агент центрального сервера был установлен как приложение, то существует возможность зарегистрировать сервис вручную и затем осуществить его запуск.

Имя сервиса различается в 32- и 64-разрядных версиях «1С:Предприятия»:

Версия «1С:Предприятия»	Имя сервиса
32-разрядная версия	1C:Enterprise 8.3 Server Agent
64-разрядная версия	1C:Enterprise 8.3 Server Agent (x86-64)

Регистрация сервиса выполняется следующей командой:

```
ragent.exe -instsrvc|-rmsrvc -usr <имя> -pwd <пароль>
-start|-stop -debug
-port <порт> -regport <порт> -range <диапазоны>
-secllev <уровень> -d <каталог>
```

[Копировать в буфер обмена](#)

ВНИМАНИЕ! Название и значение параметра должны разделяться символом пробел.

ПРИМЕЧАНИЕ. Выполнение операций регистрации, отмены регистрации, запуска и остановки сервиса агент кластера (**ragent**) необходимо выполнять от имени администратора. В процессе работы проверяет наличие необходимых для работы привилегий, и в случае их отсутствия выполняется запрос повышения привилегий.

`-instsrvc`

Регистрация агента кластера как сервиса Windows. Если **ragent** запущен с этим ключом, то он выполняет регистрацию в списке сервисов Windows и завершается.

Ключ **-instsrvc** несовместим с ключом **-rmsrvc**.

`-rmsrvc`

Отмена регистрации агента кластера как сервиса Windows. Если **ragent** запущен с этим ключом, то он отменяет свою регистрацию в списке сервисов Windows и завершается.

Ключ **-rmsrvc** несовместим с ключом **-instsrvc**.

`-start`

Запустить **ragent**, зарегистрированный как сервис Windows. Выполняет запуск **ragent**, ранее зарегистрированного как сервис Windows, после чего завершается.

`-stop`

Остановить **ragent**, зарегистрированный и запущенный как сервис Windows. Выполняет остановку **ragent**, ранее зарегистрированного и запущенного как сервис Windows, после чего завершается.

`-debug`

Запуск кластера серверов в режиме отладки конфигураций.

СОВЕТ. В связи с тем, что в режиме отладки производительность сервера падает, рекомендуется использовать отладочный режим **только** для тех серверов, на которых выполняется отладка.

`-usr <имя>, -pwd <пароль>`

Имя и пароль пользователя Windows, от имени которого должен запускаться **ragent** как сервис Windows. Могут использоваться только совместно с ключом **-instsrvc** при регистрации **ragent** как сервиса Windows.

`-port <порт>`

Номер сетевого порта агента сервера (**ragent**). Этот порт используется консолью кластера для обращения к центральному серверу. Порт агента кластера также указывается в качестве сетевого порта рабочего сервера. Значение по умолчанию: **1540**.

`-regport <порт>`

Номер сетевого порта главного менеджера кластера (**rmngr**), создаваемого по умолчанию при первом запуске **ragent**. Значение по умолчанию: **1541**.

`-range <диапазоны>`

Диапазоны сетевых портов для динамического выбора. Из них выбираются служебные порты процессов кластера при невозможности их выбора из настроек соответствующего рабочего сервера. Значение по умолчанию: **1560:1591**. Примеры значений диапазонов: **45:49, 45:67, 70:72, 77:90**.

`-secllev <уровень>`

Уровень безопасности процесса агента кластера. Определяет уровень безопасности соединений, устанавливаемых с процессом **ragent**. Уровень может принимать значения:

- **0** (по умолчанию) – соединения незащищенные;
- **1** – защищенные соединения только на время выполнения аутентификации пользователей;
- **2** – постоянно защищенные соединения.

`-d <каталог>`

Каталог, в котором будут расположены (или располагаются) служебные файлы кластера серверов (в том числе список кластеров и список информационных баз кластера). Если параметр не указан, используется каталог по умолчанию: **%USERPROFILE%\Local Settings\Application Data\1C\1cv8** (**%LOCALAPPDATA%\1C\1cv8** для ОС Windows Vista и старше).

ПРИМЕЧАНИЕ. Имя каталога не должно завершаться символом **"\"**, если оно заключено в кавычки. Правильно: **"c:\my path"**, неправильно: **"c:\my path\"**.

Например:

```
ragent.exe -instsrvc -usr .\usr1cv8 -pwd SuperSecurePassword
```

[Копировать в буфер обмена](#)

Подробнее об уровне безопасности соединения см. [здесь](#).

Запуск сервиса по умолчанию осуществляется автоматически при включении компьютера. Также запуск сервиса можно выполнить средствами Windows: **Мой компьютер – Управление – Службы и приложения – Службы – Агент сервера 1С:Предприятия 8 (My computer – Manage – Computer Management – Services and Applications – Services – 1C:Enterprise 8 Server Agent)**. Остановка сервиса также выполняется средствами Windows.

Для отмены регистрации сервиса следует выполнить команду:

```
ragent.exe -rmsrvc
```

[Копировать в буфер обмена](#)

4.1.2. Для ОС Linux

Программа установки настраивает запуск серверных процессов так, что они запускаются в режиме «демонов», то есть без привязки к управляющему терминалу. При необходимости агент сервера может быть запущен с указанием ключей командной строки.

4.1.2.1. Запуск агента сервера

Для запуска агента сервера используются следующие ключи командной строки:

```
./ragent -daemon -debug
        -port <порт> -regport <порт> -range <диапазоны>
        -secllev <уровень> -d <каталог>
```

[Копировать в буфер обмена](#)

ВНИМАНИЕ! Название и значение параметра должны разделяться символом «пробел».

`-daemon`

Данный ключ позволяет запустить агента сервера в режиме «демона», т. е. в режиме фонового приложения, не выполняющего взаимодействия с терминалом, откуда это приложение запущено. Запуск агента сервера с этим ключом не означает, что после перезагрузки системы агент сервера будет автоматически запущен на исполнение.

`-debug`

Запуск кластера серверов в режиме отладки конфигураций.

`-port <порт>`

Номер сетевого порта агента сервера (**ragent**). Этот порт используется консолью кластера для обращения к центральному серверу. Порт агента кластера также указывается в качестве сетевого порта рабочего сервера. Значение по умолчанию: **1540**.

`-regport <порт>`

Номер сетевого порта главного менеджера кластера (**rmngr**), создаваемого по умолчанию при первом запуске **ragent**. Значение по умолчанию: **1541**.

`-range <диапазоны>`

Диапазоны сетевых портов для динамического выбора. Из них выбираются служебные порты процессов кластера при невозможности их выбора из настроек соответствующего рабочего сервера. Значение по умолчанию: **1560:1591**. Примеры значений диапазонов: **45:49, 45:67, 70:72, 77:90**.

`-secllevel <уровень>`

Необязательный. Уровень безопасности процесса агента кластера. Определяет уровень безопасности соединений, устанавливаемых с процессом **ragent**. Уровень может принимать значения:

- **0** (по умолчанию) – соединения незащищенные;
- **1** – защищенные соединения только на время выполнения аутентификации пользователей;
- **2** – постоянно защищенные соединения.

Подробнее об уровне безопасности соединения см. [здесь](#).

`-d <каталог>`

Каталог, в котором будут расположены (или располагаются) служебные файлы кластера серверов (в том числе список кластеров и список информационных баз кластера). Если параметр не указан, используется каталог по умолчанию: **~/1cv8**. Если путь к каталогу содержит пробелы, путь нужно заключать в кавычки, например:

```
-d "~/cluster data"
```

[Копировать в буфер обмена](#)

Остановка агента сервера, запущенного как приложение, выполняется нажатием клавиш **Ctrl + C**.

4.1.2.2. Запуск агента сервера при помощи скрипта

Для управления агентом сервера «1С:Предприятия» предназначен специализированный скрипт `/etc/init.d/srv1cv8`. Скрипт всегда запускает сервер в режиме «демона». Скрипт использует следующие ключи командной строки:

[Копировать в буфер обмена](#)

```
/etc/init.d/srv1cv8 start|stop|restart|info|status
```

```
-start
```

Запускает сервер. Скрипт позволяет запустить единственный экземпляр сервера «1С:Предприятие».

```
-stop
```

Останавливает сервер. При этом останавливается только тот сервер, который ранее был запущен этим скриптом (см. команду [start](#)).

```
-restart
```

Перезапускает сервер. Эквивалентна последовательности команд [stop](#) и [start](#).

```
-info
```

Отображает информацию о настройках сервера: порты, указанные при запуске, каталог кластера, статус режима отладки конфигурации, уровень безопасности соединений.

```
-status
```

Отображает информацию о состоянии сервера (запущен/не запущен, и если запущен, то работает ли в данный момент).

Для задания параметров запуска агента сервера «1С:Предприятия» существует конфигурационный файл `/etc/sysconfig/srv1cv8` (если установка выполнялась для RPM-системы) или файл `/etc/init.d/srv1cv8` (если установка выполнялась для DEB-системы). Описание параметров конфигурационного файла – см. книгу «1С:Предприятие 8.3. Руководство администратора».

Пример конфигурационного файла:

[Копировать в буфер обмена](#)

```
#-----
# 1C:Enterprise Server configuration parameters
#-----
# 1C:Enterprise Server keytab file.
# default - usr1cv8.keytab file in 1C:Enterprise Server
# installation directory
#SRV1CV8_KEYTAB=
# Number of the cluster port created by default during first
# launch of ragent
# default - 1540
SRV1CV8_PORT=1540
# Number of cluster agent main port. This port is used by the
# cluster console to address the central Server. Cluster agent
# port is also specified as the IP port of the working Server.
# default - 1541
SRV1CV8_REGPORT=1541
# Port range for connection pool
# example values:
# 45:49
# 45:67,70:72,77:90
# default - 1560:1691
SRV1CV8_RANGE=1560:1691
# 1C:Enterprise Server configuration debug mode
# 0 - default - off
# 1 - on
SRV1CV8_DEBUG=0
# Path to directory with cluster data
# default - $HOMEDIR/.1cv8/1C/1Cv8
SRV1CV8_DATA=$HOMEDIR/.1cv8/1C/1Cv8
# Security level:
# 0 - default - unprotected connections
# 1 - protected connections only for the time of user
# authentication
# 2 - permanently protected connections
SRV1CV8_SECLEV=0
#-----
# end of config
#-----
```

4.1.2.3. Установка автоматического запуска сервера «1С:Предприятия»

Для того чтобы сервер «1С:Предприятия» автоматически запускался во время загрузки операционной системы, нужно выполнить одну из следующих команд:

Для RPM-систем:

[Копировать в буфер обмена](#)

```
chkconfig --add srv1cv8
```

Для DEB-систем:

[Копировать в буфер обмена](#)

```
update-rc.d srv1cv8 defaults
```

Указанные команды добавляют скрипт запуска сервера «1С:Предприятия» (подробнее см. [здесь](#)) в список автоматически запускаемых служб. В этом случае параметры сервера будут получены из конфигурационного файла `/etc/sysconfig/srv1cv8` (если установка выполнялась для RPM-системы) или файл `/etc/init.d/srv1cv8` (если установка выполнялась для DEB-системы). Описание параметров конфигурационного файла – см. книгу «1С:Предприятие 8.3. Руководство администратора».

4.2. Обеспечение совместной работы нескольких серверных процессов

В подавляющем большинстве случаев на одном рабочем сервере функционирует один агент сервера.

Когда несколько кластеров создаются под управлением одного агента сервера, агент сервера обеспечивает отсутствие конфликтов их сетевых портов. В случае, когда кластеры создаются под управлением разных агентов сервера, отсутствие конфликтов сетевых портов менеджеров кластера следует обеспечивать самостоятельно.

Отсутствие конфликтов диапазонов сетевых портов рабочих процессов на данном сервере (если данный сервер задействован в разных кластерах) всегда следует обеспечивать самостоятельно даже в том случае, когда такие кластеры функционируют под управлением одного и того же агента сервера.

Ситуация, когда на одном компьютере функционируют параллельно два и более агента серверов, каждый из которых управляет своим набором кластеров, является вполне нормальной, но достаточно редкой и специфичной. Например, необходимость в этом может возникнуть тогда, когда на одном компьютере нужно использовать различные версии сервера «1С:Предприятия».

Для того чтобы обеспечить параллельное функционирование двух агентов сервера, обслуживающих различные кластеры, нужно соблюсти следующие условия:

- агенты серверов должны иметь различные сетевые порты;

- агенты серверов должны обращаться к различным каталогам служебных файлов;
- кластеры серверов, создаваемые для каждого из агентов серверов, должны иметь различные сетевые порты;
- диапазоны сетевых портов, используемых рабочими процессами на данном сервере, не должны пересекаться (если данный сервер задействован в различных кластерах).

4.2.1. Для ОС Windows

Рассмотрим действия, которые необходимо предпринять для того, чтобы запустить на компьютере второй экземпляр сервера «1С:Предприятие».

СОВЕТ. Установку второго экземпляра сервера «1С:Предприятия» рекомендуется выполнять как приложение, а не сервис ОС Windows. При необходимости регистрацию сервера в качестве сервиса можно выполнить позднее, указав при регистрации нужные параметры.

ПРИМЕЧАНИЕ. Операции регистрации, отмены регистрации, запуска и остановки сервиса агент кластера (**ragent**) необходимо выполнять от имени администратора. В процессе работы проверяется наличие необходимых для работы привилегий, и в случае их отсутствия выполняется запрос повышения привилегий.

Следует помнить, что программа установки не позволяет поменять сетевые порты сервера, поэтому после установки новый экземпляр сервера не сможет работать.

Примеры необходимых действий будут выполняться при запуске сервера «1С:Предприятия» на ОС аналогичной разрядности, т. е. 32-разрядного сервера на 32-разрядной ОС или 64-разрядного сервера на 64-разрядной ОС. В случае запуска 32-разрядного сервера «1С:Предприятия» на 64-разрядной ОС Windows следует путь `C:\Program Files` заменить на `C:\Program Files (x86)` во всех примерах в следующих разделах.

4.2.1.1. Одновременная работа нескольких серверов разных версий «1С:Предприятия»

4.2.1.1.1. Как сервис

Для того чтобы запустить сервер «1С:Предприятия» версии 8.3 в качестве сервиса ОС Windows при работающем сервере «1С:Предприятия» версии 8.1, необходимо проделать следующие действия:

1. Перейти в каталог `bin` вновь установленной версии сервера «1С:Предприятия». В примере используется «1С:Предприятие» версии 8.3.3.150.

```
c:
cd "c:\Program Files\1cv8\8.3.3.150\bin"
```

[Копировать в буфер обмена](#)

2. Удалить регистрацию сервера «1С:Предприятия» версии 8.3.

```
ragent -rmsrvc
```

[Копировать в буфер обмена](#)

3. Удалить содержимое каталога реестра кластера. Положение каталога зависит от того, каким образом выполнялась установка сервера «1С:Предприятия» версии 8.3 (подробнее см. [здесь](#)).

```
rmdir -s -q "c:\Program Files\1cv8\srvinfo"
```

[Копировать в буфер обмена](#)

4. Выполнить регистрацию сервиса с другими значениями сетевых портов.

```
ragent -instsrvc -port 2540 -regport 2541 -range 2560:2590 -usr .\usr1cv8 -pwd UsrPwd8 -d "d:\DbData\srvinfo"
```

[Копировать в буфер обмена](#)

В примере выполняется регистрация сервера со следующими значениями портов:

- номер сетевого порта агента сервера – **2540**;
- номер сетевого порта менеджера кластера – **2541**;
- диапазон портов для динамического выбора – **2560:2590**;
- данные реестра кластера расположены в каталоге `d:\DbData\srvinfo`;
- пользователь, от имени которого выполняется сервис сервера «1С:Предприятия», – `usr1cv8`;
- пароль пользователя, от имени которого выполняется сервис сервера «1С:Предприятия», – `UsrPwd8`;
- при необходимости включить отладку для регистрируемого сервиса следует в командную строку добавить ключ `-debug`.

5. Запустить сервер «1С:Предприятия».

```
ragent -start
```

[Копировать в буфер обмена](#)

4.2.1.1.2. Как приложение

Если сервер «1С:Предприятия» запускается как приложение, то для смены сетевых портов необходимо выполнить следующие действия:

1. Завершить работу экземпляра сервера, нажав **Ctrl + C** в консольном окне с работающим сервером.
2. Перейти в каталог `bin` вновь установленной версии сервера «1С:Предприятия». В примере используется «1С:Предприятие» версии 8.3.3.150.

```
c:
cd "c:\Program Files\1cv8\8.3.3.150\bin"
```

[Копировать в буфер обмена](#)

3. Удалить содержимое каталога реестра кластера. Положение каталога зависит от того, каким образом выполнялась установка сервера «1С:Предприятия» версии 8.3 (подробнее см. [здесь](#)).

```
rmdir -s -q "%USERPROFILE%\Local Settings\Application Data\1C\1cv8"
```

[Копировать в буфер обмена](#)

4. Запустить сервер «1С:Предприятия» с новыми значениями сетевых портов и другими данными.

```
ragent -port 3540 -regport 3541 -range 3560:3590 -d "d:\DbData\srvinfo"
```

[Копировать в буфер обмена](#)

В примере выполняется регистрация сервера со следующими значениями портов:

- номер сетевого порта агента сервера – **2540**;
- номер сетевого порта менеджера кластера – **2541**;
- диапазон портов для динамического выбора – **2560:2590**;
- данные реестра кластера расположены в каталоге `d:\DbData\srvinfo`;

- при необходимости включить отладку для регистрируемого сервиса следует в командную строку добавить ключ `-debug`;
- при последующих запусках следует использовать точно такую же строку запуска. Для этого ее можно поместить в командный файл ОС Windows.

4.2.1.2. Одновременная работа нескольких серверов одинаковой версии «1С:Предприятия»

4.2.1.2.1. Как сервис

В «1С:Предприятии» отсутствуют штатные средства регистрации нескольких экземпляров сервиса сервера «1С:Предприятия» одной версии. Для того чтобы выполнить такую регистрацию, следует воспользоваться утилитой `sc.exe`. При выполнении такой регистрации, должны различаться имена служб, номера сетевых портов и адреса каталогов кластера.

В качестве примера рассмотрим командный файл, выполняющий регистрацию службы сервера.

Файл `register-service.bat`:

```
@echo off
rem %1 - полный номер версии 1С:Предприятия
rem %2 - первые две цифры номеров портов. Для портов 1540,1541,1560:1591 - это цифра 15
rem %3 - каталог с данными реестра кластера
set SrvUserName=<имя пользователя>
set SrvUserPwd=<пароль пользователя>
set RangePort=%260:%291
set BasePort=%241
set CtrlPort=%240
set SrvName="1С:Enterprise 8.3 Server Agent %CtrlPort% %1"
set BinPath="%C:\Program Files\1cv8\%1\bin\ragent.exe" -srvc -agent -regport %BasePort% -port %CtrlPort% -range %RangePort% -d \"%-3\" -debug"
set Description="Агент сервера 1С:Предприятия 8.3. Параметры: %1, %CtrlPort%, %BasePort%, %RangePort%"
if not exist "%-3" mkdir "%-3"
sc stop %SrvName%
sc delete %SrvName%
sc create %SrvName% binPath= %BinPath% start= auto obj= %SrvUserName% password= %SrvUserPwd% displayName= %Description% depend= Dnscache/Tcpip/Tcpip6/lanman
```

[Копировать в буфер обмена](#)

Перед применением данного командного файла необходимо указать в нем данные реального пользователя (имя и пароль) от имени которого будет работать служба кластера серверов (строки `set SrvUserName=` и `set SrvUserPwd=`). Данный командный файл выполняет регистрацию указанной версии сервера «1С:Предприятия». Имя службы является строкой, содержащей следующую информацию:

- **1С:Enterprise 8.3 Server Agent**,
- Номер сетевого порта главного менеджера кластера,
- Полный номер версии «1С:Предприятия».

При регистрации сервера версии **8.3.3.100**, использующего сетевой порт главного менеджера кластера с номером **2540**, имя службы будет выглядеть следующим образом: **1С:Enterprise 8.3 Server Agent 2540 8.3.3.100**.

Пример использования:

```
register-service 8.3.3.100 25 "c:\cluster_data\cluster 1"
register-service 8.3.3.100 35 "c:\cluster_data\cluster 2"
```

[Копировать в буфер обмена](#)

В данном примере первая строка выполняется регистрацию службы сервера со следующими параметрами:

- Имя службы: **1С:Enterprise 8.3 Server Agent 2540 8.3.3.100**;
- Порты сервера: **2540, 2541, 2560:2591**;
- Каталог с данными реестра кластера: **c:\cluster_data\cluster 1**;
- Описание службы: **Агент сервера 1С:Предприятия 8.3. Параметры: 8.3.3.100, 2540, 2541, 2560:2591**.

Вторая строка выполняет регистрацию службы сервера со следующими параметрами:

- Имя службы: **1С:Enterprise 8.3 Server Agent 3540 8.3.3.100**;
- Порты сервера: **3540, 3541, 3560:3591**;
- Каталог с данными реестра кластера: **c:\cluster_data\cluster 2**;
- Описание службы: **Агент сервера 1С:Предприятия 8.3. Параметры: 8.3.3.100, 3540, 3541, 3560:3591**.

Если необходимо отменить регистрацию службы сервера, то можно воспользоваться следующим примером командного файла.

Файл `unregister-service.bat`:

```
@echo off
rem %1 - полный номер версии 1С:Предприятия
rem %2 - первые две цифры номеров портов. Для портов 1540,1541,1560:1591 - это цифра 15
set SrvName="1С:Enterprise 8.3 Server Agent %240 %1"
sc stop %SrvName%
sc delete %SrvName%
```

[Копировать в буфер обмена](#)

Пример использования:

```
unregister-service 8.3.3.100 25
```

[Копировать в буфер обмена](#)

Командный файл останавливает службу и удаляет ее регистрацию. Имя службы формируется по тем же правилам, что и при регистрации новой (нестандартной) службы сервера «1С:Предприятия».

4.2.1.2.2. Как приложение

Одновременный запуск нескольких экземпляров сервера одной версии, работающего как приложение, возможен из командной строки. В этом случае параметры командной строки должны отличаться не только номерами сетевых портов, но и адресами каталогов кластера.

```
start "Server 1" "C:\Program Files\1cv8\8.3.3.100\bin\ragent.exe" -port 2540 -regport 2541 -range 2560:2590 -d d:\ClusterData\Srv1
start "Server 2" "C:\Program Files\1cv8\8.3.3.100\bin\ragent.exe" -port 3540 -regport 3541 -range 3560:3590 -d d:\ClusterData\Srv2
```

[Копировать в буфер обмена](#)

В примере выполняется запуск двух экземпляров сервера «1С:Предприятие» со следующими параметрами:

- 1-й сервер имеет заголовок окна **Server 1**, обслуживает сетевые порты **25xx** и хранит данные кластера по адресу **d:\ClusterData\Srv1**.
- 2-й сервер имеет заголовок окна **Server 2**, обслуживает сетевые порты **35xx** и хранит данные кластера по адресу **d:\ClusterData\Srv2**.

4.2.1.3. Замена сетевых портов работающего экземпляра сервера «1С:Предприятие»

Возможность замены сетевых портов уже работающего экземпляра сервера «1С:Предприятия» не предусмотрена. При возникновении такой необходимости следует:

- создать новый экземпляр сервера с необходимыми значениями сетевых портов и остальных параметров;
- зарегистрировать в новом сервере существующие информационные базы;
- перевести клиентов на новый сервер;
- остановить и удалить старый экземпляр сервера «1С:Предприятия» (вместе с данными кластера).

4.2.2. Для ОС Linux

Рассмотрим действия, которые необходимо предпринять для того, чтобы запустить на компьютере второй экземпляр сервера «1С:Предприятие».

Следует помнить, что программа установки не позволяет менять сетевые порты сервера, поэтому после установки новый экземпляр сервера не сможет работать.

ПРИМЕЧАНИЕ. Установка сервера «1С:Предприятия» для ОС Linux всегда выполняется в режиме демона.

Под ОС Linux возможна совместная работа серверов «1С:Предприятия» только разных версий.

4.2.2.1. Как демон

Чтобы изменить сетевые порты работающего экземпляра сервера, нужно выполнить следующие действия:

- Остановить работу сервера «1С:Предприятия».

```
/etc/init.d/srv1cv83 stop
```

[Копировать в буфер обмена](#)

- Удалить каталог кластера.

```
rm -rf /home/usr1cv8/.1cv8
```

[Копировать в буфер обмена](#)

- Изменить параметры запуска сервера «1С:Предприятия», указав нужные значения сетевых портов и остальных параметров (включая каталог реестра кластера). Описание конфигурационного файла можно получить в книге «1С:Предприятие 8.3. Руководство администратора».

- Запустить сервер «1С:Предприятия».

```
/etc/init.d/srv1cv83 start
```

[Копировать в буфер обмена](#)

4.2.2.2. Как приложение

Если сервер «1С:Предприятия» выполняется как приложение, то для изменения сетевых портов нужно выполнить следующие действия:

- Завершить работу экземпляра сервера, нажав **Ctrl + C** в консольном окне с работающим сервером.
- Удалить содержимое каталога реестра кластера. Как правило, это каталог `.1cv8` домашнего каталога пользователя, от имени которого работает сервер.

```
rm -rf /home/<ПользовательСервера>/.1cv8
```

[Копировать в буфер обмена](#)

- Перейти в каталог с двоичными файлами «1С:Предприятия».

Для 32-разрядной версии:

```
cd /opt/1C/v8.3/i386
```

[Копировать в буфер обмена](#)

Для 64-разрядной версии:

```
cd /opt/1C/v8.3/x86_64
```

[Копировать в буфер обмена](#)

- Запустить сервер «1С:Предприятия» с новыми значениями сетевых портов и другими данными.

```
./ragent -port 2040 -regport 2041 -range 2060:2090 -d /home/usr1cv8/dbinfo/.1cv8
```

[Копировать в буфер обмена](#)

В примере выполняется запуск сервера со следующими значениями портов:

- номер сетевого порта агента сервера – **2040**;
- номер сетевого порта менеджера кластера – **2041**;
- диапазон портов для динамического выбора – **2060:2090**;
- данные реестра кластера расположены в каталоге `/home/usr1cv8/dbinfo/.1cv8`;
- при необходимости включить отладку для регистрируемого сервиса следует в командную строку добавить ключ `-debug`;
- при последующих запусках следует использовать точно такую же строку запуска, для этого ее можно поместить в командный файл ОС Linux.

Глава 5. Администрирование

Данный раздел содержит описание элементов администрирования системы «1С:Предприятие», характерных для клиент-серверного варианта работы системы.

5.1. Администрирование информационной базы

5.1.1. Создание резервной копии в клиент-серверном варианте

ВНИМАНИЕ! Резервное копирование необходимо выполнять перед любой операцией, которая может повредить данные, находящиеся в информационной базе.

Формирование резервной копии средствами СУБД (или копированием файла **1Cv8.1CD**) позволяет получить максимально точную копию существующей базы данных с возможностью возврата именно к тому состоянию, из которого выполнялась резервная копия. В то же время при выгрузке информационной базы в файл и последующей загрузке предполагается, что данные в базе данных находятся в корректном состоянии. Если в базе данных информация находится с нарушениями, то выгрузка может быть выполнена, а процесс загрузки может завершиться неудачно.

Рекомендуется использовать для создания резервной копии:

- Для файлового варианта – копирование файла **1Cv8.1CD**, при этом к информационной базе не должно быть никаких подключений (в том числе и конфигуратором).
- Для клиент-серверного варианта – средствами резервного копирования соответствующей СУБД.

Материалы по резервному копированию конкретных SQL-серверов:

- Microsoft SQL Server 2000 (на английском языке):
<http://msdn.microsoft.com/ru-ru/library/aa196685>.
- Microsoft SQL Server 2005:
[http://msdn.microsoft.com/ru-ru/library/ms187048\(v=sql.90\).aspx](http://msdn.microsoft.com/ru-ru/library/ms187048(v=sql.90).aspx).
- Microsoft SQL Server 2008:
[http://msdn.microsoft.com/ru-ru/library/ms187048\(v=sql.100\).aspx](http://msdn.microsoft.com/ru-ru/library/ms187048(v=sql.100).aspx).
- Microsoft SQL Server 2008 R2:
[http://msdn.microsoft.com/ru-ru/library/ms187048\(v=sql.105\).aspx](http://msdn.microsoft.com/ru-ru/library/ms187048(v=sql.105).aspx).
- Microsoft SQL Server 2012:
[http://msdn.microsoft.com/ru-ru/library/ms187048\(v=sql.110\).aspx](http://msdn.microsoft.com/ru-ru/library/ms187048(v=sql.110).aspx).
- Microsoft SQL Server 2014:
[http://msdn.microsoft.com/ru-ru/library/ms187048\(v=sql.120\).aspx](http://msdn.microsoft.com/ru-ru/library/ms187048(v=sql.120).aspx).
- PostgreSQL 8.1 (на английском языке):
<http://www.postgresql.org/docs/8.1/static/backup.html>.
- PostgreSQL 8.2 (на английском языке):
<http://www.postgresql.org/docs/8.2/interactive/backup.html>.
- PostgreSQL 8.3 (на английском языке):
<http://www.postgresql.org/docs/8.3/interactive/backup.html>.
- PostgreSQL 8.4 (на английском языке):
<http://www.postgresql.org/docs/8.4/interactive/backup.html>.
- PostgreSQL 9.0 (на английском языке):
<http://www.postgresql.org/docs/9.0/interactive/backup.html>.
- PostgreSQL 9.1 (на английском языке):
<http://www.postgresql.org/docs/9.1/interactive/backup.html>.
- PostgreSQL 9.2 (на английском языке):
<http://www.postgresql.org/docs/9.2/interactive/backup.html>.
- IBM DB2 v9.1:
<http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/c0006150.htm>.
- IBM DB2 v9.5 (на английском языке):
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.admin.ha.doc/doc/c0052073.html>.
- IBM DB2 v9.7 (на английском языке): <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.ha.doc/doc/c0052073.html>.
- IBM DB2 v10.1 (на английском языке):
<http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ha.doc/doc/c0052073.html>.
- Oracle Database 10g Release 2 (на английском языке):
http://download.oracle.com/docs/cd/B19306_01/backup.102/b14192/toc.htm.
- Oracle Database 11g Release 1 (на английском языке):
http://download.oracle.com/docs/cd/B28359_01/backup.111/b28270/toc.htm.
- Oracle Database 11g Release 2 (на английском языке):
http://docs.oracle.com/cd/E11882_01/backup.112/e10642/toc.htm.

5.1.2. Преобразование информационной базы для использования в клиент-серверном варианте

Выгрузка/загрузка информационной базы в файл используется для преобразования файлового варианта в клиент-серверный.

ВНИМАНИЕ! Перед выполнением выгрузки и загрузки все сеансы работы с данной информационной базой необходимо закрыть.

Следует выполнить выгрузку информационной базы (пункт **Администрирование – Выгрузить информационную базу данных**). Затем создать пустую информационную базу в клиент-серверном варианте. Открыть базу в режиме **Конфигуратор** и выполнить загрузку информационной базы (пункт **Администрирование – Загрузить информационную базу данных**).

5.2. Администрирование кластера серверов

Данная утилита представляет собой подключаемый (snap-in) модуль **Консоли управления (MMC)** и может быть использована на компьютерах, на которых установлено программное обеспечение **Microsoft Management Console**. **Microsoft Management Console** является стандартным средством в операционных системах Microsoft Windows XP/Server 2003/Vista/Server 2008. При установке «1С:Предприятия» для работы с информационными базами в клиент-серверном варианте в этих операционных системах соответствующий пункт для запуска утилиты будет внесен в системное меню **Пуск**, в раздел **1С Предприятие 8**.

Задачи, решаемые с помощью утилиты администрирования кластера серверов, аналогичны возможностям программного администрирования кластера серверов (см. [здесь](#)).

5.2.1. Запуск утилиты администрирования

Утилита может быть запущена только на компьютере, на котором имеется программное обеспечение **Microsoft Management Console**. Запуск утилиты может быть осуществлен с помощью команды меню **Пуск – Программы – 1С Предприятие 8.3 – Дополнительно – Администрирование серверов 1С Предприятия** для 32-разрядной версии утилиты или **Пуск – Программы – 1С Предприятие 8.3 (x86-64) – Дополнительно – Администрирование серверов 1С Предприятия** для 64-разрядной версии утилиты.

Есть и альтернативный путь. Для этого следует запустить **Консоль управления (MMC)**. Сделать это можно с помощью командной строки:

```
mmc
```

[Копировать в буфер обмена](#)

После того как **Консоль управления (MMC)** будет запущена, следует выбрать пункт меню **Консоль – Добавить или удалить оснастку...**

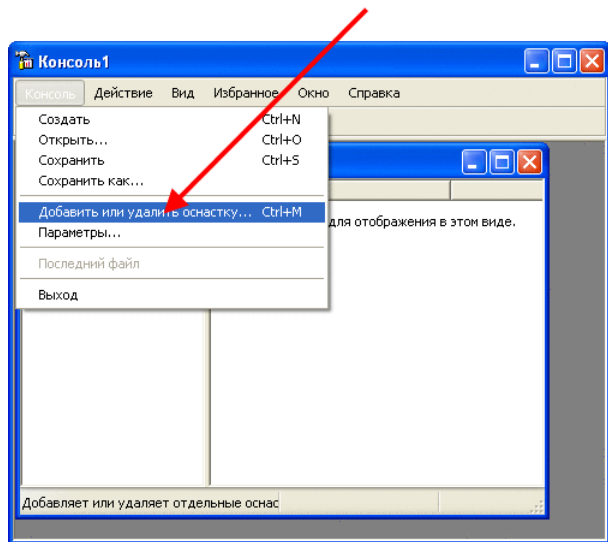


Рис. 27. Добавление оснастки

На экране появится диалог **Добавить/удалить оснастку**. В этом диалоге следует нажать кнопку **Добавить**.

На экране появится диалог **Добавить изолированную оснастку**. В списке этого диалога следует выбрать пункт **1С:Enterprise 8.3 Servers**, нажать кнопку **Добавить** и закрыть диалог с помощью кнопки **Закреть**.

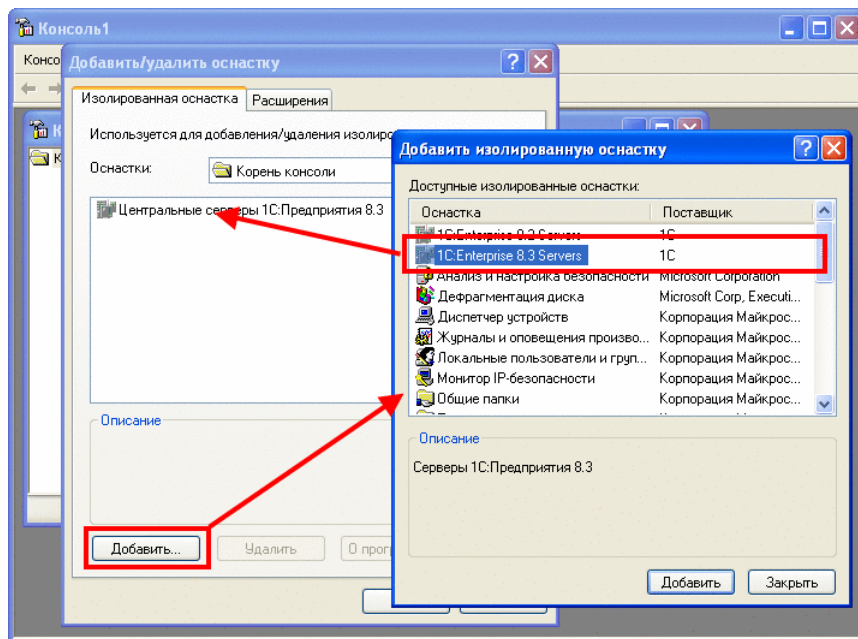


Рис. 28. Выбор оснастки

Затем следует в диалоге **Добавить/удалить оснастку** нажать кнопку **ОК**. Таким образом будет выполнено подключение утилиты администрирования к **Microsoft management console**.

5.2.2. Регистрация экземпляра рабочего сервера

При первом запуске утилита администрирования показывает в дереве центральных серверов только рабочий сервер, установленный на компьютере, на котором запущена сама утилита администрирования (если, конечно, агент сервера запущен на этом компьютере).

Для отображения списка центральных серверов следует в дереве центральных серверов выбрать и раскрыть ветку **1С:Enterprise 8.3 Central Servers**.

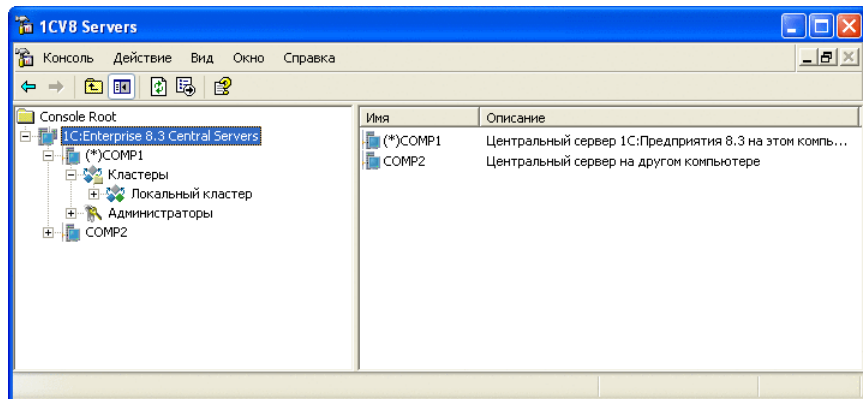


Рис. 29. Дерево центральных серверов

Дерево центральных серверов содержит перечень центральных серверов, к которым подключена утилита. Каждый центральный сервер идентифицируется именем компьютера, на котором он запущен. В поле свойств отображается список центральных серверов, содержащий сетевой адрес центрального сервера и его описание.

5.2.2.1. Подключение утилиты к центральному серверу

Для подключения утилиты к новому центральному серверу следует выполнить команду контекстного меню [Создать – Центральный сервер 1C:Предприятия 8.3](#) или аналогичную команду главного меню утилиты.

В результате выполнения команды на экране появится диалог свойств центрального сервера.

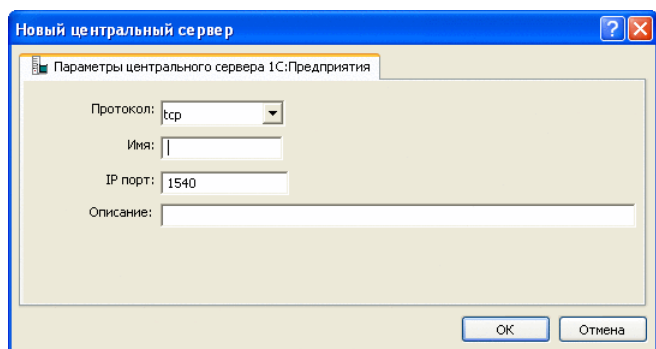


Рис. 30. Новый центральный сервер

В поля диалога необходимо ввести следующие данные:

Протокол

Протокол соединения с центральным сервером. Выбирается из списка. Значение по умолчанию – [tcp](#). В этом поле указывается протокол, по которому утилита администрирования будет подключаться к агенту центрального сервера. Поддерживается единственное значение – [tcp](#).

Имя

Сетевой адрес центрального сервера, на котором запущен агент сервера.

IP порт

Номер сетевого порта агента сервера, который запущен на центральном сервере. Значение по умолчанию – [1540](#). Порт агента сервера указывается при его запуске (см. [здесь](#)).

Описание

Произвольное описание центрального сервера.

ПРИМЕЧАНИЕ. При настройке кластера серверов и центрального сервера адреса одних и тех же серверов должны быть записаны одинаково. Идентичность различных адресов одного и того же сервера не проверяется. Например, если центральный сервер в консоли кластера имеет адрес [Server](#), то в списке рабочих серверов он также должен называться [Server](#), а не [54.34.86.12](#) (даже если в DNS указано, что [Server](#) это [54.34.86.12](#)) или [localhost](#).

5.2.2.2. Просмотр и изменение свойств центрального сервера

Для того чтобы просмотреть и изменить свойства центрального сервера, следует выбрать требуемый сервер в списке центральных серверов и выполнить команду контекстного меню [Свойства](#) или аналогичную команду главного меню утилиты.

В результате выполнения команды на экране появится диалог свойств центрального сервера, описанный в предыдущем разделе (см. [здесь](#)).

5.2.2.3. Отключение утилиты от центрального сервера

Для отключения утилиты от центрального сервера следует выбрать требуемый сервер в списке центральных серверов и выполнить команду контекстного меню [Удалить](#) или аналогичную команду главного меню утилиты.

5.2.2.4. Отсоединение утилиты от центрального сервера

Утилита администрирования кластера серверов «1C:Предприятия» может быть отсоединена от центрального сервера, при этом параметры подключения к центральному серверу удалены не будут.

Для отсоединения от центрального сервера следует выбрать требуемый сервер в списке центральных серверов и выполнить команду контекстного меню [Отсоединить](#) или аналогичную команду главного меню утилиты.

5.2.3. Работа со списком администраторов центрального сервера

Кластер серверов предоставляет возможность создать список администраторов центрального сервера для того, чтобы ряд административных действий (например, добавление нового кластера, просмотр списка администраторов центрального сервера) могли выполнять только аутентифицированные пользователи.

По умолчанию список администраторов центрального сервера пуст. Это означает, что для выполнения перечисленных действий система не будет требовать аутентификацию администратора центрального сервера.

Для отображения списка администраторов центрального сервера следует в дереве центральных серверов выбрать нужный сервер, а затем выбрать и раскрыть ветку **Администраторы**.

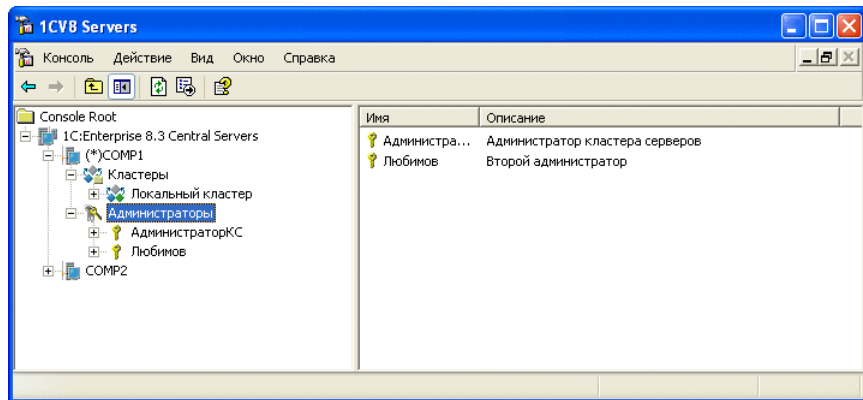


Рис. 31. Список администраторов кластера серверов

Дерево центральных серверов содержит перечень администраторов выбранного центрального сервера. Каждый администратор идентифицируется именем. В поле свойств отображается список администраторов выбранного центрального сервера, содержащий имя администратора и его описание.

Для рабочего сервера, который не является центральным, и в котором имеется непустой список администраторов центрального сервера, в списке администраторов должен присутствовать администратор, у которого определена аутентификация ОС пользователя, от имени которого запущен **ragent** центрального сервера кластера, или должен присутствовать администратор с именем и паролем, совпадающими с именем и паролем одного из администраторов центрального сервера кластера.

5.2.3.1. Добавление администратора центрального сервера

Для добавления нового администратора центрального сервера следует выбрать в дереве центральных серверов требуемый сервер, выбрать ветку **Администраторы** и выполнить команду контекстного меню **Создать – Администратор** или аналогичную команду главного меню утилиты.

В результате выполнения команды на экране появится диалог свойств администратора центрального сервера.

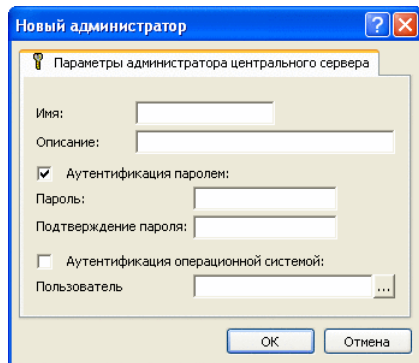


Рис. 32. Новый администратор центрального сервера

В поля диалога необходимо ввести следующие данные:

Имя

Имя администратора центрального сервера.

Описание

Произвольное описание администратора центрального сервера.

Аутентификация паролем

Признак аутентификации паролем. По умолчанию установлен.

Пароль

Пароль администратора центрального сервера.

Подтверждение пароля

Подтверждение пароля.

Аутентификация операционной системы

Признак аутентификации средствами операционной системы.

Пользователь

Пользователь операционной системы. Может быть задан в виде: `\\Имя_домена\Имя_пользователя`. Например: `\\domainname\username`. Пользователь может быть установлен как непосредственным вводом соответствующей строки, так и посредством выбора пользователя операционной системы среди пользователей, видимых с компьютера, на котором запущена утилита администрирования информационных баз. Для этого необходимо нажать кнопку «...» и в открывшемся диалоге выбрать нужного пользователя операционной системы.

ВНИМАНИЕ! Имена администраторов центрального сервера должны быть уникальными в пределах каждого центрального сервера.

Система допускает аутентификацию администратора центрального сервера двумя способами:

- с помощью пароля,
- средствами операционной системы.

При аутентификации с помощью пароля система вызовет диалог аутентификации администратора центрального сервера, в котором необходимо ввести имя пользователя и пароль.

При аутентификации средствами операционной системы не требуется каких-либо действий по вводу логина и пароля; диалог аутентификации не отображается. Система анализирует, от имени какого пользователя операционной системы выполняется подключение, и на основании этого определяет соответствующего администратора центрального сервера.

ВНИМАНИЕ! Если для администратора не указан ни один из видов аутентификации, то такой администратор может выполнять только те действия, которые не требуют аутентификации.

5.2.3.2. Просмотр и изменение свойств администратора центрального сервера

Для просмотра и изменения свойств администратора центрального сервера следует выбрать требуемого администратора в списке администраторов центрального сервера и выполнить команду контекстного меню **Свойства** или аналогичную команду главного меню утилиты.

В результате выполнения команды на экране появится диалог свойств администратора центрального сервера.

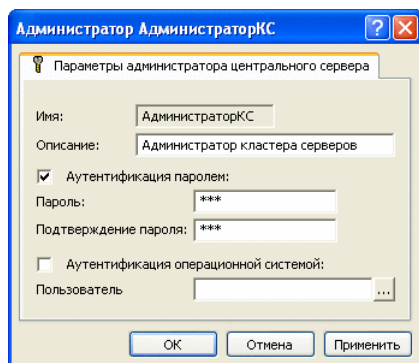


Рис. 33. Свойства администратора центрального сервера

Все свойства, за исключением имени администратора, будут доступны для редактирования. Значения полей **Пароль** и **Подтверждение пароля** будут скрыты.

5.2.3.3. Удаление администратора центрального сервера

Для удаления администратора центрального сервера следует выбрать требуемого администратора в списке администраторов центрального сервера и выполнить команду контекстного меню **Удалить** или аналогичную команду главного меню утилиты.

5.2.3.4. Аутентификация администратора центрального сервера

Аутентификация администратора центрального сервера будет запрошена системой автоматически, при попытке выполнения действия, требующего аутентификации (если список администраторов центрального сервера не пуст). При этом на экране появится диалог аутентификации администратора центрального сервера.

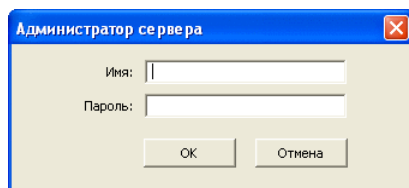


Рис. 34. Аутентификация администратора

В поля диалога необходимо ввести следующие данные:

Имя

Имя администратора центрального сервера.

Пароль

Пароль администратора центрального сервера.

5.2.4. Работа со списком кластеров

Для отображения списка кластеров, зарегистрированных на центральном сервере, следует в дереве центральных серверов выбрать нужный сервер, а затем выбрать и раскрыть ветку **Кластеры**.

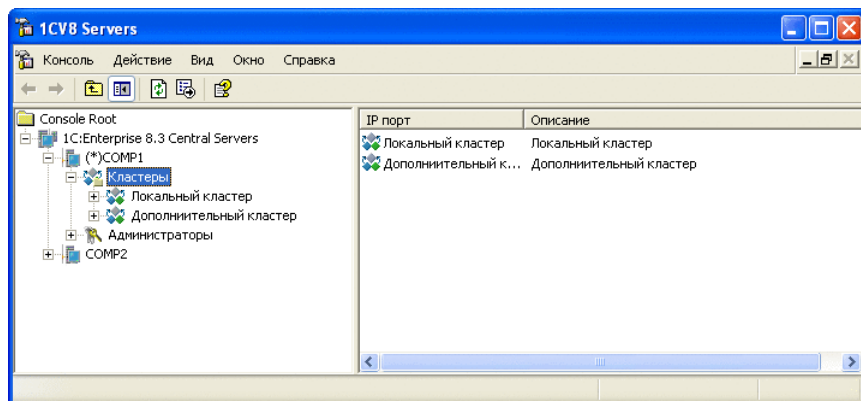


Рис. 35. Список кластеров

Дерево центральных серверов содержит перечень кластеров выбранного центрального сервера. Каждый кластер идентифицируется номером сетевого порта. В поле свойств отображается список кластеров выбранного центрального сервера, содержащий номер порта кластера и описание кластера.

5.2.4.1. Добавление кластера

Для добавления нового кластера в центральный сервер следует выбрать в дереве центральных серверов требуемый сервер, выбрать ветку **Кластеры** и выполнить команду контекстного меню **Создать – Кластер** или аналогичную команду главного меню утилиты.

В результате выполнения команды на экране появится диалог свойств кластера.

Рис. 36. Новый кластер

В поля диалога необходимо ввести следующие данные:

Имя кластера

Произвольное описание кластера.

Компьютер

Имя центрального сервера, на котором размещается кластер. Не редактируется.

IP Порт

Номер сетевого порта менеджера кластера. По умолчанию – 1541.

ВНИМАНИЕ! Номера сетевых портов менеджеров кластеров должны быть уникальными в пределах каждого центрального сервера.

Защищенное соединение

Уровень безопасности кластера. Выбирается из списка (возможные значения: **выключено**, **только соединение**, **постоянно**). Значение по умолчанию – **выключено**. Подробнее об использовании уровней безопасности кластера см. [здесь](#).

Интервал перезапуска __ секунд

Интервал времени после запуска процесса, по истечении которого рабочий процесс перезапускается. Нулевое значение означает, что рабочие процессы не будут перезапускаться автоматически.

Допустимый объем памяти __ KB

Устанавливает предельный объем виртуального адресного пространства, после постоянного превышения которого в течение определенного времени (см. следующий параметр) рабочий процесс будет автоматически перезапущен. Нулевое значение означает, что критический объем памяти не задан и автоматический перезапуск рабочих процессов не выполняется.

Интервал превышения допустимого объема памяти __ секунд

Определяет интервал времени постоянного превышения предельного объема виртуального адресного пространства, занимаемого рабочим процессом (см. предыдущий параметр), после которого выполняется автоматический перезапуск рабочего процесса. Нулевое значение означает, что автоматически перезапуск рабочих процессов не выполняется.

Выключенные процессы останавливать через __ секунд

Интервал времени, по истечении которого выключенный рабочий процесс принудительно останавливается, независимо от наличия соединений. Работа всех соединений с этим процессом завершается аварийно. Значение свойства может быть изменено во время работы кластера. Нулевое значение означает, что принудительное завершение процесса не выполняется.

Уровень отказоустойчивости

Уровень отказоустойчивости определяет максимальное количество рабочих серверов, входящих в состав кластера, одновременный выход из строя которых не приведет к аварийному завершению сеансов подключенных пользователей. Более подробное описание данного свойства см. [здесь](#).

Режим распределения нагрузки

Параметр определяет, по какому критерию будет выбираться рабочий процесс при установке нового соединения. Более подробное описание данного свойства см. [здесь](#).

5.2.4.2. Просмотр и изменение свойств кластера

Для просмотра и изменения свойств кластера следует выбрать требуемый кластер в списке кластеров центрального сервера и выполнить команду контекстного меню **Свойства** или аналогичную команду главного меню утилиты.

В результате выполнения команды на экране появится диалог свойств кластера.

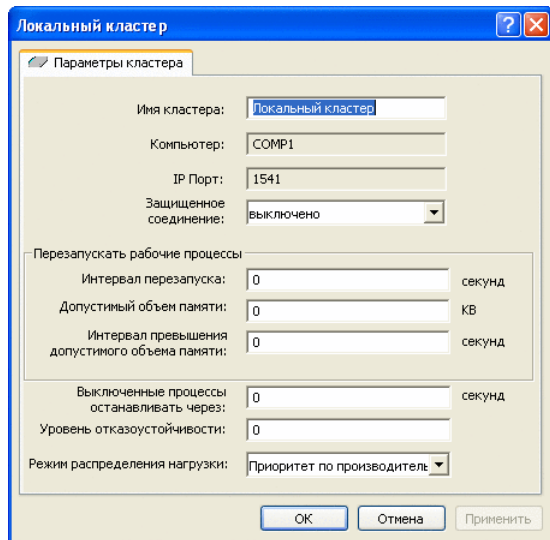


Рис. 37. Свойства кластера серверов

5.2.4.3. Вызов операции применения

Требования начнут работать только после того момента, как будет выполнена операция применения. До момента применения, список требований, редактируемый в консоли кластера, не оказывает влияния на работу кластера серверов. Операция применения может быть **полной** или **частичной**.

В случае частичного применения будут переназначены сервисы, которые поддерживают миграцию между рабочими серверами без потери данных. Переназначение для остальных сервисов будет выполнена только в том случае, если новые требования назначения функциональности не позволяют этим сервисам остаться «на старом месте» или недоступны рабочие сервера, на которых данные сервисы функционировали на момент вызова операции частичного применения. Операция полного применения затрагивает все сервисы и не обращает внимания на формальные характеристики сервисов.

При выполнении операции полного применения могут аварийно завершиться все клиентские приложения, подключенные к кластеру серверов, на котором выполняется операция. Такая ситуация также возможна в случае частичного применения, если принято решение выполнить переназначение для всех сервисов, а не только для тех, которые допускают перемещение без потери данных.

Для вызова операции применения следует в контекстном меню кластера серверов выбрать пункт меню [Применить требования назначения функциональности \(частичное\)](#) или [Применить требования назначения функциональности \(полное\)](#).

5.2.4.4. Удаление кластера

Для удаления кластера следует выбрать требуемый кластер в списке кластеров центрального сервера и выполнить команду контекстного меню [Удалить](#) или аналогичную команду главного меню утилиты.

ВНИМАНИЕ! Удаление кластера серверов приведет к тому, что будут аварийно завершены все соединения с удаляемым кластером.

5.2.5. Работа со списком рабочих серверов кластера

Для отображения списка рабочих серверов кластера следует в дереве центральных серверов выбрать нужный сервер, выбрать нужный кластер, зарегистрированный на данном сервере, а затем выбрать и раскрыть ветку [Рабочие серверы](#).

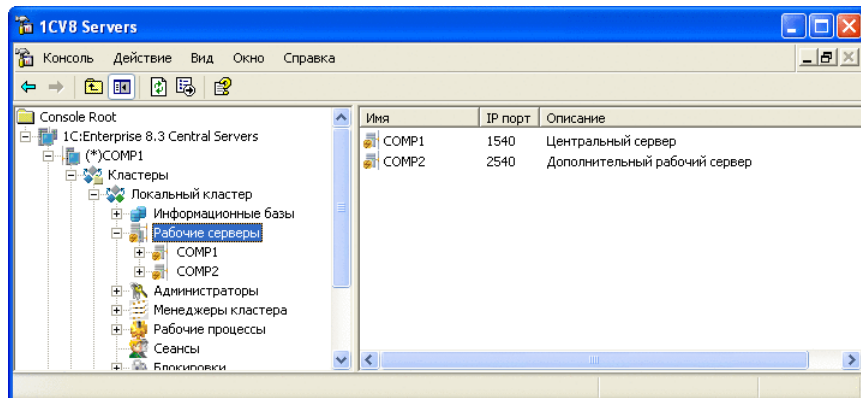


Рис. 38. Список рабочих серверов

Дерево центральных серверов содержит перечень рабочих серверов кластера. Каждый рабочий сервер идентифицируется сетевым именем. В поле свойств отображается список рабочих серверов выбранного кластера, содержащий имя сервера, номер сетевого порта агента сервера, запущенного на этом сервере, и описание сервера.

При создании кластера по умолчанию рабочий сервер, на котором он создан, включается в состав кластера. Для этого рабочего сервера устанавливается флажок [Центральный сервер](#).

5.2.5.1. Добавление рабочего сервера в кластер

Для добавления нового рабочего сервера в кластер следует выбрать в дереве центральных серверов требуемый центральный сервер, выбрать требуемый кластер, зарегистрированный на этом сервере, выбрать ветку [Рабочие серверы](#) и выполнить команду контекстного меню [Создать – Рабочий сервер](#) или аналогичную команду главного меню утилиты.

В результате выполнения команды на экране появится диалог свойств рабочего сервера.

Рис. 39. Новый рабочий сервер

В поля диалога необходимо ввести следующие данные:

Описание

Произвольное описание сервера кластера.

Компьютер

Сетевой адрес рабочего сервера, на котором запущен агент сервера.

ЗАМЕЧАНИЕ. Если в качестве сетевого адреса рабочего сервера «1С:Предприятия» (свойство *Компьютер*) указан IP-адрес в точечной нотации, то его присутствие в DNS (файл *hosts*) не требуется.

IP Порт

Номер сетевого порта агента сервера, который запущен на указанном компьютере. По умолчанию – 1540.

Диапазоны IP портов

Диапазон сетевых портов, который будет использоваться системой для назначения адресов рабочим процессам, создаваемым на этом сервере. По умолчанию – 1560:1591.

Максимальный объем памяти рабочих процессов

Максимальный объем памяти (в байтах), доступный всем рабочим процессам кластера на данном рабочем сервере.

Может принимать значение от -1 до 9 223 372 036 854 775 807:

- -1 – не ограничен максимальный объем памяти, доступный рабочим процессам кластера на данном рабочем сервере;
- 0 – значение определяется автоматически как 80% объема оперативной памяти сервера.

Каждый рабочий процесс кластера определяет объем памяти, занимаемой всеми рабочими процессами кластера на этом рабочем сервере (назовем это значение *ПамятьПроцесса*). Это значение обновляется 1 раз в 2 секунды. При начале вызова сервера фиксируется текущее значение *ПамятьПроцесса* на момент начала вызова (назовем это значение *ПамятьПроцессаТекущая*). В процессе выполнения вызова вычисляется объем памяти, израсходованной при выполнении этого вызова (назовем это значение *ПамятьЗаВызов*).

Если в результате выделения памяти в одном вызове сервера значение *Максимальный объем памяти рабочих процессов* превышено менее чем на значение *Безопасный расход памяти за один вызов*, то такой вызов не прерывается. Если в течение вызова значение *ПамятьЗаВызов* превысило значение *Безопасный расход памяти за один вызов*, и значение *ПамятьПроцессаТекущая+ПамятьЗаВызов* превысило значение *Максимальный объем памяти рабочих процессов*, то вызов прерывается исключением и завершается аварийно. При этом в технологический журнал выводится событие *EXCP*, содержащее:

- Текст исключения;
- Контекст исключения.

Безопасный расход памяти за один вызов

Объем памяти в байтах, использование которого в процессе вызова сервера считается безопасным.

Может принимать значение от -1 до 9 223 372 036 854 775 807:

- -1 – любой вызов сервера считается опасным, если за время вызова сервера достигнут максимальный объем памяти рабочего процесса;
- 0 – значение объема определяется автоматически, как 5% максимального объема памяти рабочих процессов на данном рабочем сервере.

Объем памяти рабочих процессов, до которого сервер считается производительным

Суммарный объем оперативной памяти, занятый рабочими процессами этого рабочего сервера, по достижению которого на этот рабочий сервер перестанут назначаться новые соединения с информационными базами. Значение 0 означает, что никакого ограничения не установлено.

Количество ИБ на процесс

Количество информационных баз, соединения с которыми могут обслуживаться одним рабочим процессом данного сервера. Значение 0 означает, что ограничение не установлено.

Если количество информационных баз превысит это количество – кластер серверов создаст на этом рабочем сервере дополнительный рабочий процесс.

Количество соединений с информационными базами, которые может обслуживать один рабочий процесс данного сервера. Значение 0 означает, что ограничение не установлено.

Если количество соединений, обслуживаемых рабочим процессом, превысит это количество – кластер серверов создаст на этом рабочем сервере дополнительный рабочий процесс.

Порт главного менеджера кластера

Номер сетевого порта главного менеджера кластера, запущенного на данном рабочем сервере. Этот сетевой порт используется при формировании адреса кластера серверов для указания клиентскому приложению. Адрес выглядит следующим образом: <Свойство Компьютер>:<Порт главного менеджера кластера>. Если свойство Компьютер имеет имя COMP1, а свойство Порт главного менеджера кластера равно 2541, то адрес кластера серверов будет выглядеть как COMP1:2541.

Менеджер под каждый сервис

Управляет выделение отдельного менеджера кластера под каждый тип сервиса (см. [здесь](#)). Если флажок установлен, то для каждого типа сервиса кластера будет создаваться свой менеджер кластера. В противном случае все сервисы кластера назначаются на один менеджер кластера данного рабочего сервера.

СОВЕТ. Устанавливать данное свойство можно в процессе опытной эксплуатации.

Центральный сервер

Если флажок установлен, то с этим рабочим сервером будет выполняться синхронизация реестра кластера и адрес этого рабочего сервера может быть использован для подключения к кластеру клиентским приложением.

5.2.5.2. Просмотр и изменение свойств сервера кластера

Для просмотра и изменения свойств сервера кластера следует выбрать требуемый сервер в списке серверов кластера и выполнить команду контекстного меню [Свойства](#) или аналогичную команду главного меню утилиты.

В результате выполнения команды на экране появится диалог свойств рабочего сервера.

Рис. 40. Свойства рабочего сервера

Помимо свойств, которые отображаются в диалоге при создании нового сервера, диалог свойств существующего сервера отображает также служебный сетевой порт агента сервера, который назначается системой автоматически при запуске агента сервера и используется для взаимодействия процессов кластера сервера.

В свойствах существующего сервера доступен для изменения только диапазон сетевых портов.

5.2.5.3. Исключение сервера из кластера

Удаление рабочего сервера может привести к аварийному завершению клиентских соединений. Чтобы этого не произошло, необходимо для удаляемого сервера добавить следующее требование (с максимальным приоритетом) назначения функциональности:

- Тип объекта: [Любой объект требования](#).
- Тип требования: [Не назначать](#).
- Имя ИБ: не указывать.
- Значение дополнительного параметра: не указывать.

Затем необходимо применить новый набор требований и дождаться разрыва существующих соединений. Затем можно удалять рабочий сервер с помощью команды контекстного меню [Удалить](#) или аналогичной команды главного меню.

Удаление последнего рабочего сервера с установленным признаком [Центральный сервер](#) невозможно.

5.2.5.4. Требования назначения функциональности

Для просмотра списка требований назначения функциональности, заданных для конкретного рабочего сервера, необходимо выбрать в дереве центральных серверов выбрать нужный сервер. Затем необходимо последовательно выбрать нужный кластер и рабочий сервер. Для выбранного сервера следует выбрать ветку [Требования назначения функциональности](#).

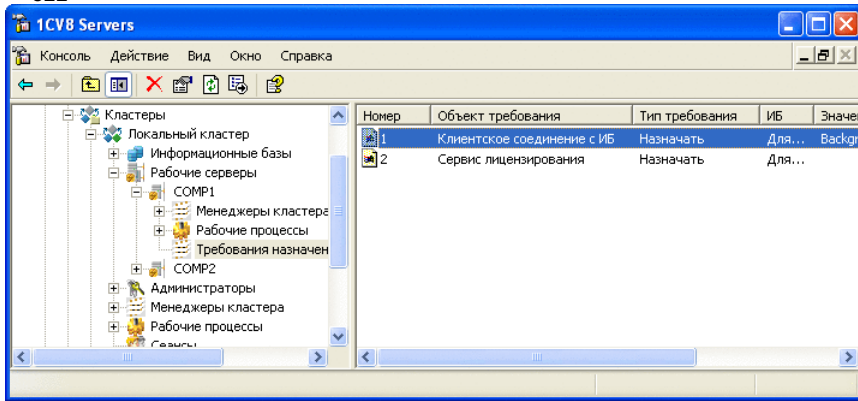


Рис. 41. Требования назначения функциональности

Для создания нового требования необходимо выбрать команду **Создать – Требование назначения функциональности** в контекстном меню или в меню **Действие** главного меню.

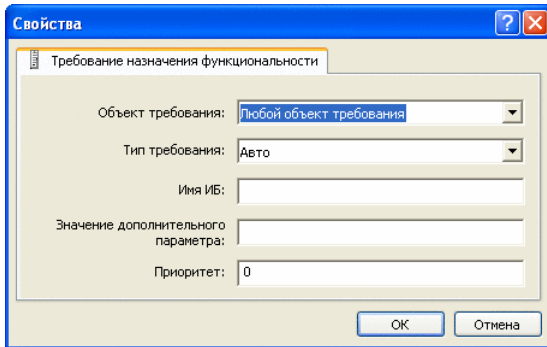


Рис. 42. Новое требование назначения функциональности

Подробное описание требований назначения функциональности см. [здесь](#).

Заданные требования обрабатываются в том порядке, в котором они расположены в списке требований конкретного сервера. Фактически, порядок обработки (приоритет требования) определяется колонкой **Номер** списка требований назначения функциональности. Чем меньше номер, тем раньше будет обрабатываться требование (тем выше приоритет). Чтобы изменить приоритет конкретного требования, необходимо встать курсором на нужное требование и в контекстном меню (или меню **Действие** главного меню) выбрать команды **Повысить приоритет требования** или **Понизить приоритет требования** в зависимости от того, необходимо повысить или понизить приоритет требования.

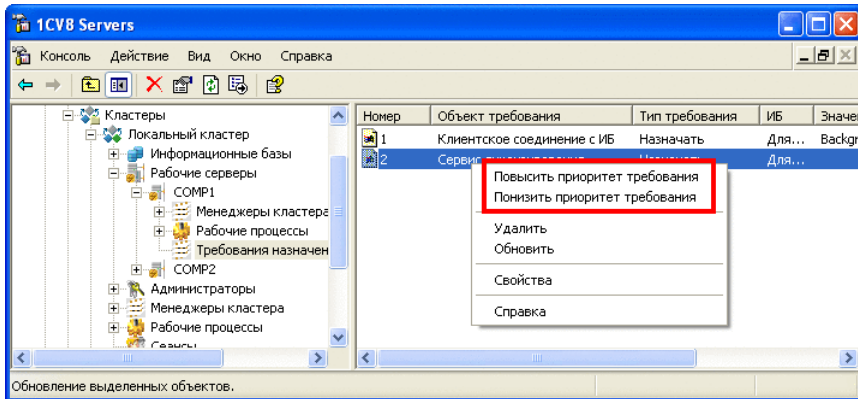


Рис. 43. Изменение приоритета требований

5.2.6. Работа со списком информационных баз

Для отображения списка информационных баз, зарегистрированных в кластере, следует в дереве центральных серверов выбрать нужный сервер, выбрать нужный кластер, зарегистрированный на данном сервере, а затем выбрать и раскрыть ветку **Информационные базы**.

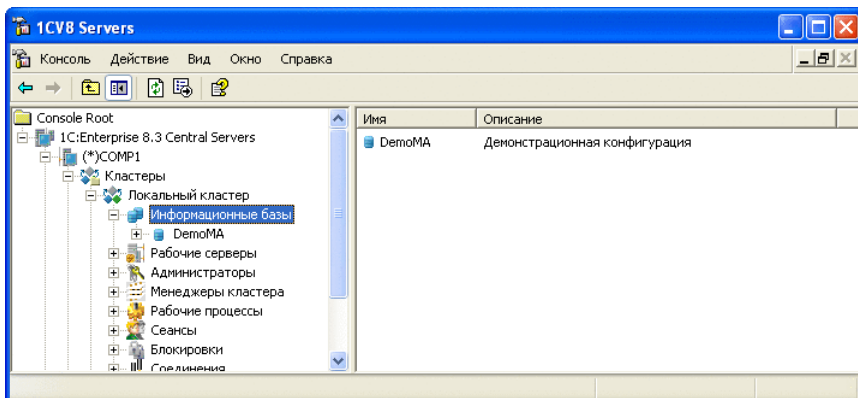


Рис. 44. Список информационных баз

Дерево центральных серверов содержит перечень информационных баз кластера. Каждая информационная база идентифицируется именем. В поле свойств отображается список информационных баз выбранного кластера, содержащий имя информационной базы и ее описание.

5.2.6.1. Регистрация новой информационной базы

Регистрация новой информационной базы в кластере серверов может быть выполнена двумя способами:

- из клиентского приложения,
- непосредственно в кластере серверов.

При добавлении новой информационной базы в клиентском приложении ее регистрация в кластере серверов выполняется системой автоматически.

Для регистрации новой информационной базы с помощью утилиты администрирования кластера серверов следует выбрать в дереве центральных серверов требуемый центральный сервер, требуемый кластер, зарегистрированный на этом сервере, ветку [Информационные базы](#) и выполнить команду контекстного меню [Создать – Информационная база](#) или аналогичную команду главного меню утилиты.

В результате выполнения команды на экране появится диалог свойств информационной базы.

Рис. 45. Новая информационная база

Параметры информационной базы эквивалентны параметрам новой информационной базы, создаваемой с помощью окна запуска «1С:Предприятия». Описание этих параметров см. книгу «1С:Предприятие 8.3. Руководство администратора».

Следует обратить внимание на параметр [Разрешить выдачу лицензий сервером 1С:Предприятия](#). Этот параметр управляет выдачей клиентских лицензий сервером «1С:Предприятия». Если параметр установлен в значение [Да](#), то сервер «1С:Предприятия» будет выдавать клиентские лицензии в тех случаях, когда клиентское приложение не смогло получить клиентскую лицензию самостоятельно (особенности получения клиентских лицензий через сервер «1С:Предприятия» описаны в книге «1С:Предприятие 8.3. Руководство администратора»). Если параметр установлен в значение [Нет](#), то сервер «1С:Предприятия» не будет выдавать клиентские лицензии. В этом случае клиентское приложение, которое самостоятельно не смогло получить лицензию, будет отображать диагностическое сообщение: [Не найдена лицензия. Не обнаружен ключ защиты программы или полученная программная лицензия!](#)

ВНИМАНИЕ! Имена информационных баз должны быть уникальными в пределах одного кластера.

В процессе регистрации новой информационной базы система проверяет, существует ли на указанном сервере баз данных база с таким именем. Если база существует, то будет установлено соединение с ней. Если существующая база данных уже содержит данные информационной базы «1С:Предприятия», то будет установлена связь с уже существующей информационной базой. А если база данных не содержит данных информационной базы, то в ней будет проинициализирована новая информационная база «1С:Предприятия».

5.2.6.2. Просмотр свойств информационной базы

Для просмотра и изменения свойств информационной базы следует выбрать требуемую информационную базу в списке информационных баз и выполнить команду контекстного меню [Свойства](#) или аналогичную команду главного меню утилиты.

Рис. 46. Свойства информационной базы

В окне свойств параметров информационной базы можно отредактировать имена сервера базы данных, используемой базы данных, изменить тип используемой СУБД, имя пользователя базы данных и его пароль.

Также для редактирования будут доступны свойства, относящиеся к блокировке установки сеансов пользователей с данной базой.

Блокировка начала сеансов включена

Если флажок установлен, то включена блокировка начала сеансов с информационной базой. При этом:

- существующие сеансы могут продолжать работу;
- возможен запуск фоновых заданий существующими сеансами;
- существующие сеансы могут устанавливать соединения;
- начало новых сеансов запрещено;
- установка новых соединений запрещена, кроме соединений от имени существующих сеансов.

Начало (дата/время начала актуальности блокировки)

Блокировка начинает действовать, если текущее время превышает значение данного свойства.

Конец (дата/время конца актуальности блокировки)

Если значение данного свойства отлично от нулевой даты и меньше или равно текущему времени, то действие блокировки заканчивается.

Сообщение

Текст, который будет частью сообщения об ошибке при попытке установки соединения с заблокированной информационной базой.

Код разрешения

Строка, которая должна быть добавлена к параметру командной строки `/UC` или к параметру строки соединения `UC`, чтобы установить соединение с информационной базой вопреки блокировке установки соединений.

Параметр блокировки

Произвольный текст. Может использоваться в конфигурациях для различных целей.

Блокировка регламентных заданий включена

Если флажок установлен, это означает, что включена блокировка регламентных заданий данной информационной базы.

Внешнее управление сеансами

Строка, описывающая параметры Web-сервиса внешнего управления сеансами. Строка параметров Web-сервиса имеет следующий формат: `Параметр=Значение; .` Строка параметров содержит 4 обязательных параметра (`wsdl`, `ns`, `srvc`, `port`) и два необязательных (`tout`, `wsver`):

- `wsdl` – URL для получения описания Web-сервиса в формате WSDL.
- `ns` – пространство имен Web-сервиса.
- `srvc` – имя Web-сервиса, который будет использоваться для внешнего управления сеансами.
- `port` – имя порта используемого Web-сервиса.
- `tout` – максимальное время ожидания обращения к Web-сервису внешнего управления сеансами, в секундах. Значение по умолчанию – 5 секунд.

- **wsver** – номер версии используемого Web-сервиса внешнего управления сеансами. Значение по умолчанию равно 2. Подробнее о сервисе внешнего управления сеансами см. [здесь](#).

Пример строки с описанием Web-сервиса:

```
wsdl=http://server/sm/ws/manager?wsdl;ns=http://v8.1c.ru/SessionManager;srvc=manager;port=managerSoap;tout=10;wsver=1;
```

[Копировать в буфер обмена](#)

Описание работы внешнего управления сеансами см. [здесь](#).

Обязательное использование внешнего управления

Если флажок установлен, то при недоступности Web-сервиса внешнего управления сеансами происходит ошибка и установка подключения к информационной базе невозможна. Если флажок сброшен и Web-сервис недоступен – то подключение возможно и ограничений на количество одновременных сеансов не накладывается.

Профиль безопасности

В случае указания в данном поле имени профиля безопасности (см. [здесь](#)), прикладное решение (на стороне сервера «1С:Предприятия») начинает работать с учетом ограничений, которые накладывает указанный профиль безопасности. Более подробное описание профиля безопасности см. [здесь](#).

Профиль безопасности безопасного режима

В случае указания имени профиля безопасности в данном поле, ограничения, накладываемые на фрагменты прикладного решения, работающие в безопасном режиме, будут определяться указанным профилем. Более подробное описание профиля безопасности см. [здесь](#).

5.2.6.3. Удаление информационной базы

Для удаления информационной базы следует выбрать требуемую информационную базу в списке информационных баз и выполнить команду контекстного меню **Удалить** или аналогичную команду главного меню утилиты.

В результате выполнения команды на экране появится предупреждающий вопрос: **Удалить информационную базу?** При положительном ответе на этот вопрос система предложит один из трех вариантов удаления информационной базы.

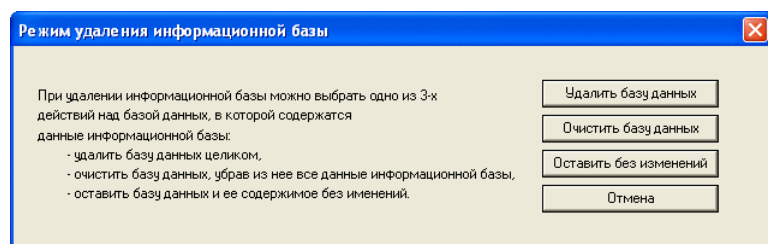


Рис. 47. Режим удаления информационной базы

- **Удалить базу данных** – при выборе этого варианта будет удалена регистрация информационной базы в кластере сервера, и кроме этого будет удалена соответствующая ей база данных на сервере баз данных.
- **Очистить базу данных** – при выборе этого варианта будет удалена регистрация информационной базы в кластере сервера, и кроме этого будут удалены все данные из базы данных на сервере баз данных. Сама база данных не будет удалена из сервера баз данных.
- **Оставить без изменений** – при выборе этого варианта будет удалена только регистрация информационной базы в кластере сервера. Никаких изменений в базе данных произведено не будет.

В том случае, если выбран вариант **Удалить базу данных**, но есть соединения пользователей с этой базой данных, будет удалена регистрация информационной базы в кластере сервера, а при попытке удаления базы данных сервер баз данных выдаст сообщение об ошибке, например:

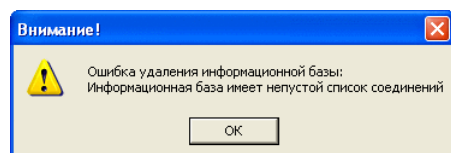


Рис. 48. Ошибка удаления информационной базы

5.2.7. Работа со списком администраторов кластера

Система предоставляет возможность создать отдельный список администраторов для каждого кластера, зарегистрированного на центральном сервере, для того, чтобы административные действия с кластером могли выполнять только аутентифицированные пользователи.

По умолчанию список администраторов кластера пуст. Это означает, что система не будет требовать аутентификацию администратора кластера.

Для отображения списка администраторов кластера следует в дереве центральных серверов выбрать нужный сервер, выбрать нужный кластер, зарегистрированный на этом сервере, а затем выбрать и раскрыть ветку **Администраторы**.

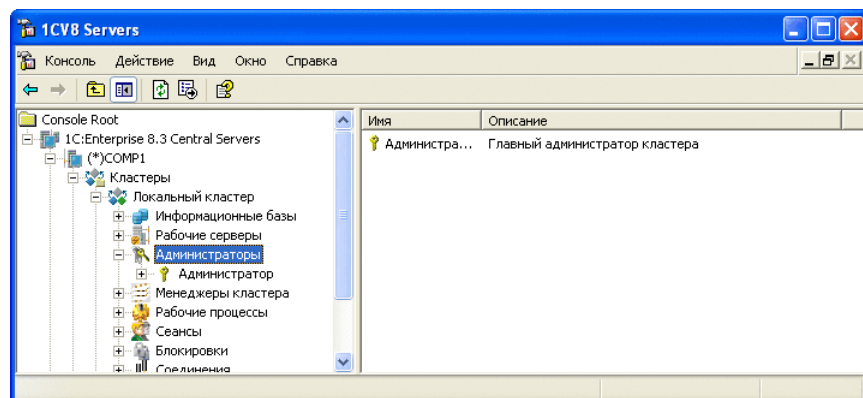


Рис. 49. Список администраторов кластера

Дерево центральных серверов содержит перечень администраторов выбранного кластера. Каждый администратор идентифицируется именем. В поле свойств отображается список администраторов выбранного кластера, содержащий имя администратора и его описание.

5.2.7.1. Добавление администратора кластера

Для добавления нового администратора кластера следует выбрать в дереве центральных серверов требуемый сервер, выбрать требуемый кластер, зарегистрированный на этом сервере, выбрать ветку [Администраторы](#) и выполнить команду контекстного меню [Создать – Администратор](#) или аналогичную команду главного меню утилиты.

В результате выполнения команды на экране появится диалог свойств администратора кластера.

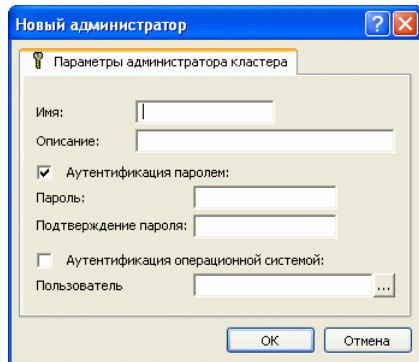


Рис. 50. Новый администратор кластера

В поля диалога необходимо ввести следующие данные:

Имя

Имя администратора кластера.

Описание

Произвольное описание администратора кластера.

Аутентификация паролем

Признак аутентификации паролем. По умолчанию установлен.

Пароль

Пароль администратора кластера.

Подтверждение пароля

Подтверждение пароля.

Аутентификация операционной системы

Признак аутентификации средствами операционной системы.

Пользователь

Пользователь операционной системы. Может быть задан в виде: `\\Имя_домена\Имя_пользователя`. Например: `\\domainname\username`. Пользователь может быть установлен как непосредственным вводом соответствующей строки, так и посредством выбора пользователя операционной системы среди пользователей, видимых с компьютера, на котором запущена утилита администрирования информационных баз. Для этого необходимо нажать кнопку «...» и в открывшемся диалоге выбрать нужного пользователя операционной системы.

ВНИМАНИЕ! Имена администраторов кластера должны быть уникальными в пределах каждого кластера.

Система допускает аутентификацию администратора кластера двумя способами:

- с помощью пароля,
- средствами операционной системы.

В случае аутентификации с помощью пароля система вызовет диалог аутентификации администратора кластера (см. [здесь](#)), в который необходимо ввести имя пользователя и пароль.

В случае аутентификации средствами операционной системы не требуется каких-либо действий по вводу имени пользователя и пароля, диалог аутентификации не отображается. Система анализирует, от имени какого пользователя операционной системы выполняется подключение, и на основании этого определяет соответствующего администратора кластера.

Если для администратора не указан ни один из видов аутентификации, то такой администратор может выполнять только те действия, которые не требуют аутентификации.

5.2.7.2. Просмотр и изменение свойств администратора кластера

Для просмотра и изменения свойств администратора кластера следует выбрать требуемого администратора в списке администраторов кластера и выполнить команду контекстного меню [Свойства](#) или аналогичную команду главного меню утилиты.

В результате выполнения команды на экране появится диалог свойств администратора кластера.

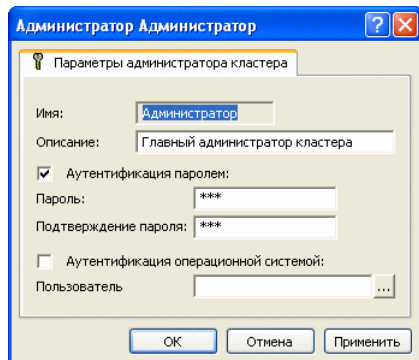


Рис. 51. Свойства администратора кластера

Все свойства, за исключением имени администратора, будут доступны для редактирования, значения полей **Пароль** и **Подтверждение пароля** будут скрыты.

5.2.7.3. Удаление администратора кластера

Для удаления администратора кластера следует выбрать требуемого администратора в списке администраторов кластера и выполнить команду контекстного меню **Удалить** или аналогичную команду главного меню утилиты.

5.2.7.4. Аутентификация администратора кластера

Аутентификация администратора кластера будет запрошена системой автоматически, при попытке выполнения действия, требующего аутентификации (если список администраторов кластера не пуст). При этом на экране появится диалог аутентификации администратора кластера.

В поля диалога необходимо ввести следующие данные:

Имя

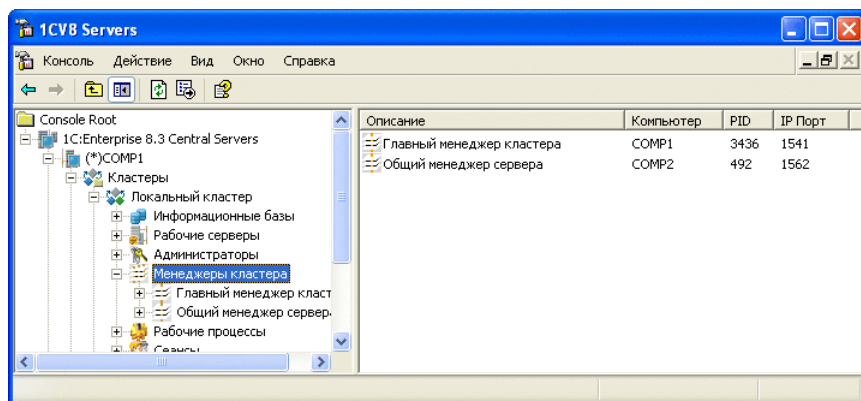
Имя администратора кластера.

Пароль

Пароль администратора кластера.

5.2.8. Просмотр списка менеджеров кластера

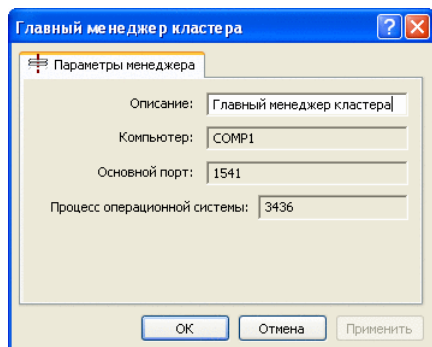
Система предоставляет возможность для просмотра и изменения списка менеджеров кластера. По умолчанию существует один менеджер кластера, который является основным и который определен в любом кластере.

**Рис. 52. Менеджеры кластера**

Кластер сам определяет количество и размещение менеджеров кластера. На это влияют требования назначения функциональности (см. [здесь](#)) и свойства рабочего сервера **Менеджер под каждый сервис** и **Центральный сервер** (см. [здесь](#)). Администратор не имеет возможности вручную добавлять и удалять менеджеры кластера.

Для просмотра описания менеджера кластера необходимо воспользоваться командой **Свойства** контекстного меню менеджера кластера.

В появившемся окне можно изменить описание менеджера кластера.

**Рис. 53. Свойства менеджера кластера**

5.2.9. Просмотр списка рабочих процессов

Список рабочих процессов может быть отображен двумя способами:

- для всего кластера в целом,
- отдельно для выбранного сервера кластера.

Чтобы отобразить список рабочих процессов для всего кластера в целом, следует в дереве центральных серверов выбрать нужный сервер, выбрать нужный кластер, зарегистрированный на данном сервере, а затем выбрать и раскрыть ветку **Рабочие процессы**.

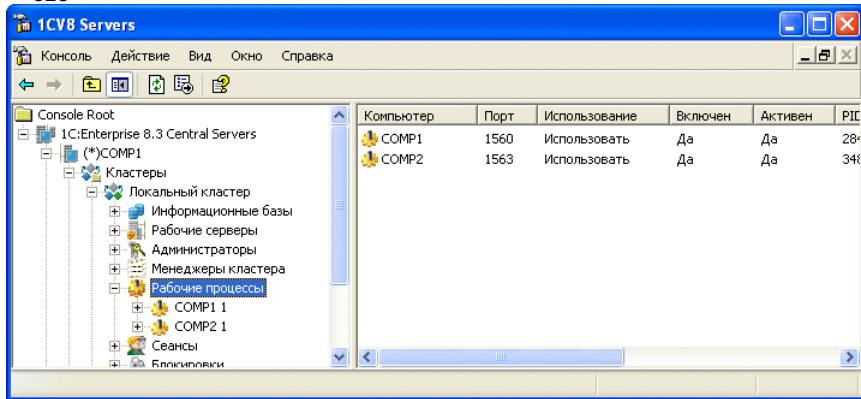


Рис. 54. Список рабочих процессов

Для отображения списка рабочих процессов только для выбранного сервера кластера следует в дереве центральных серверов выбрать нужный сервер, выбрать нужный кластер, выбрать нужный сервер кластера, а затем выбрать и раскрыть ветку **Рабочие процессы**.

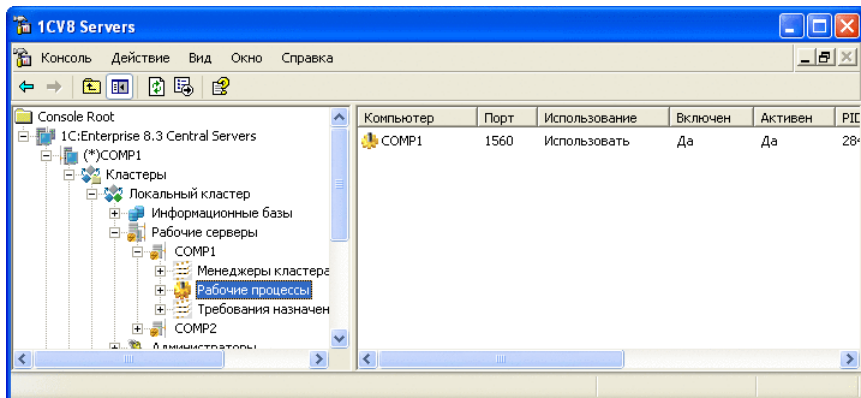


Рис. 55. Список рабочих процессов конкретного рабочего сервера

Дерево центральных серверов содержит перечень рабочих процессов. Каждый рабочий процесс идентифицируется именем сервера и порядковым номером в данном рабочем сервере. В поле свойств отображается служебная информация, которая описывает конкретный рабочий процесс. Описание отображаемых параметров приведено ниже.

Для просмотра свойств рабочего процесса следует выбрать требуемый рабочий процесс в списке рабочих процессов и выполнить команду контекстного меню **Свойства** или аналогичную команду главного меню утилиты. Администратор не имеет возможности вручную добавлять и удалять рабочие процессы.

В результате выполнения команды на экране появится диалог свойств рабочего процесса.

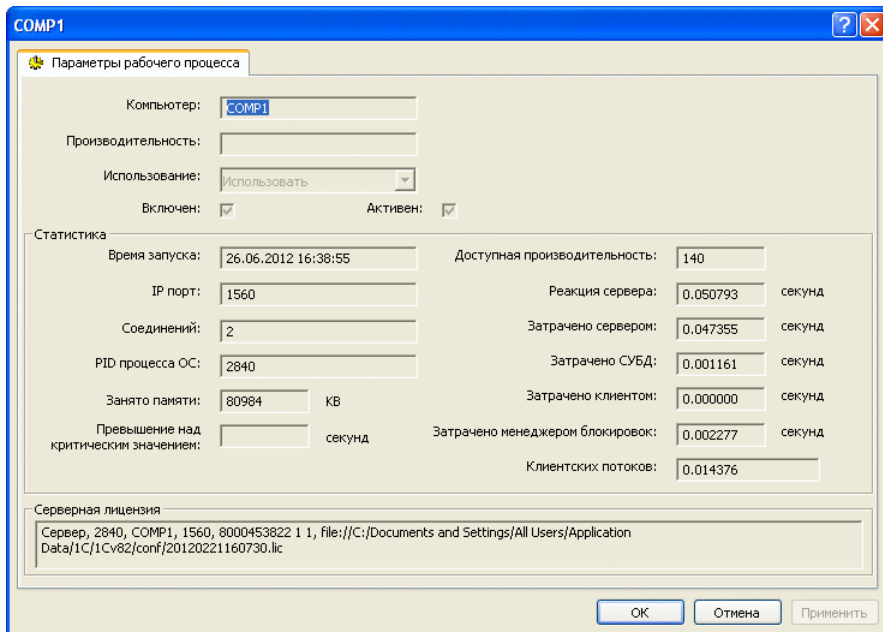


Рис. 56. Свойства рабочего процесса

Помимо свойств, которые отображаются в диалоге при создании нового рабочего процесса, диалог свойств существующего рабочего процесса содержит следующие поля, которые недоступны для редактирования:

Компьютер

На каком рабочем сервере работает рабочий процесс.

Производительность

Относительное значение производительности.

Использование

Признак использования рабочего процесса. Данное свойство может принимать следующие значения:

- [Использовать](#),
- [Не использовать](#),
- [Использовать как резервный](#).

Агент центрального сервера кластера запускает все используемые рабочие процессы. Если какой-то из используемых процессов оказывается неактивным, то кластер запускает его повторно. Если рабочий процесс помечен как не используемый, то он будет остановлен после отключения всех соединений или через время, определяемое свойством кластера [Выключенные процессы останавливать через](#).

Количество рабочих процессов и признак их использования определяется сервером автоматически. Управлять этим процессом можно с помощью настроек рабочего сервера (см. [здесь](#)) и требований назначения функциональности (см. [здесь](#)).

Включен

В данный момент рабочий процесс включен и может использоваться системой.

Активен

В данный момент рабочий процесс используется и активен.

Время запуска

Время последнего запуска рабочего процесса.

IP порт

Сетевой порт рабочего процесса, который выделяется системой динамически при запуске рабочего процесса из диапазона сетевых портов, указанного для данного сервера.

Соединений

Текущее количество соединений, обслуживаемых рабочим процессом.

PID процесса ОС

Номер процесса (в терминах ОС, под управлением которой выполняется рабочий процесс).

Занято памяти

Объем памяти, занятый рабочим процессом.

Превышение над критическим значением

Время (в секундах) превышения параметра [Занято памяти](#) над параметром [Допустимый объем памяти](#) кластера серверов. Подробнее см. [здесь](#).

Доступная производительность

Текущая доступная производительность. Подробнее см. [здесь](#).

Реакция сервера

Среднее время, затраченное на обслуживание одного соединения. Равно сумме значений следующих четырех полей:

- Затрачено сервером,
- Затрачено СУБД,
- Затрачено клиентом,
- Затрачено менеджером блокировок.

Затрачено сервером

Среднее время, затраченное рабочим процессом на обслуживание одного соединения.

Затрачено СУБД

Среднее время, затраченное СУБД на обслуживание одного соединения.

Затрачено клиентом

Среднее время, затраченное клиентом на обслуживание одного соединения.

Затрачено менеджером блокировок

Среднее время, затраченное менеджером блокировок на обслуживание одного соединения.

Клиентских потоков

Среднее количество клиентских потоков, обработанное рабочим процессом. Используется системой для вычисления производительности рабочего процесса.

Подробнее о параметрах производительности рабочего процесса см. [здесь](#).

Серверная лицензия

В данном поле отображается информация о том, какая серверная лицензия используется данным рабочим процессом и приводится характеристика этой лицензии. Описание формируемой строки см. [здесь](#).

5.2.10. Работа со списком сеансов

Список сеансов может быть отображен двумя способами:

- для всего кластера в целом,
- для отдельной информационной базы.

Чтобы отобразить список сеансов для всего кластера в целом, следует в дереве центральных серверов выбрать нужный центральный сервер, выбрать нужный кластер, зарегистрированный на данном сервере, а затем выбрать и раскрыть ветку [Сеансы](#).

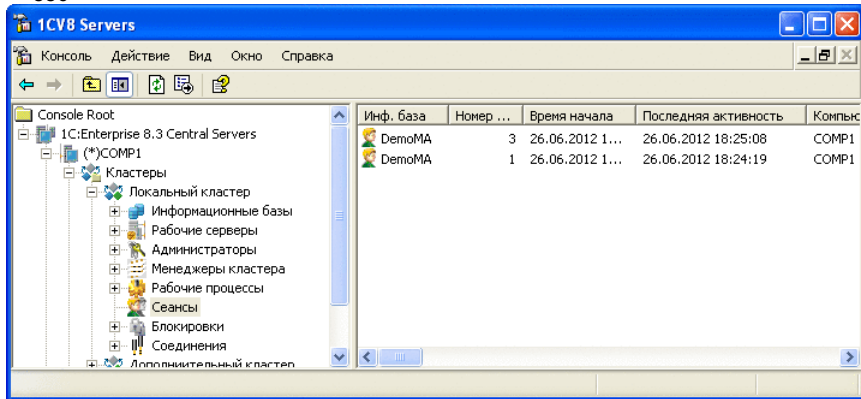


Рис. 57. Список сеансов кластера серверов

Чтобы отобразить список соединений для отдельной информационной базы, следует в дереве центральных серверов выбрать нужный центральный сервер, выбрать нужный кластер, выбрать нужную информационную базу, а затем выбрать и раскрыть ветку **Сеансы**.

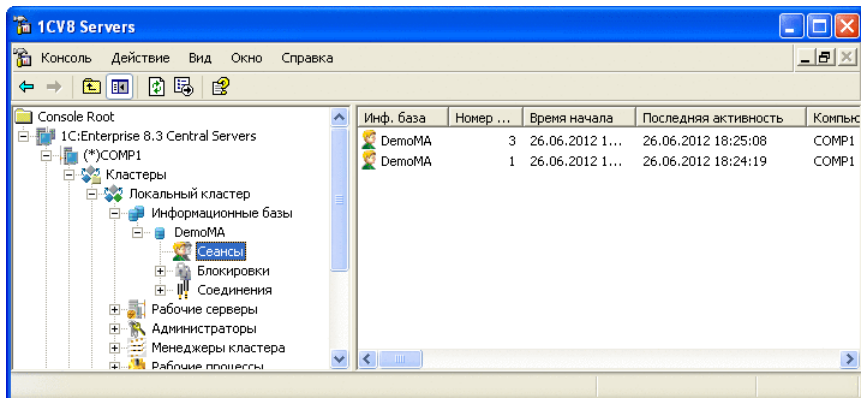


Рис. 58. Список сеансов информационной базы

В поле свойств отображается список сеансов, содержащий следующую информацию:

Информационная база

Имя информационной базы, с которой установлено соединение.

Номер сеанса

Номер сеанса.

Начало начала

Время, в которое был создан данный сеанс.

Последняя активность

Время, в которое была зафиксирована последняя активность сеанса.

Компьютер

Сетевое имя компьютера пользователя, клиентское приложение с которого инициировало создание сеанса. Имя компьютера будет пустым, если сеанс создан для обслуживания веб-клиента, тонкого клиента, подключенного через веб-сервер, или Web-сервиса.

Пользователь

Имя пользователя информационной базы.

Приложение

Режим запуска клиентского приложения (см. [здесь](#)).

Язык

Язык локализации приложения.

Сервер

Имя сервера кластера, с которым установлено соединение.

Порт

Номер сетевого порта рабочего процесса, обслуживающего это соединение.

Процесс ОС

Номер процесса рабочего сервера (в терминах операционной системы), обслуживающего данный сеанс.

Соединение

Номер соединения, которое подключено к данному сеансу.

Соединение с СУБД

Идентификатор процесса сервера баз данных. Отображается в том случае, если в данный момент соединение с базой данных захвачено сеансом: либо выполняется вызов СУБД, либо открыта транзакция, либо удерживается объект **МенеджерВременныхТаблиц**, в котором создана хотя бы одна временная таблица.

Захвачено СУБД

Длительность захвата соединения с базой данных текущим сеансом с момента захвата по текущий момент. Отображается только если соединение с СУБД захвачено сеансом.

Заблокировано СУБД

Идентификатор процесса, который заблокировал данный процесс.

Заблокировано упр.

Имя пользователя (номер сеанса) в том случае, если процесс ожидает освобождения транзакционной блокировки.

Время вызова СУБД (текущее)

Время текущего обращения СУБД с начала обращения (в секундах).

Время вызовов СУБД (5 мин.)

Время обращения к СУБД за последние пять минут (в секундах).

Время вызова СУБД (всего)

Время обращения к СУБД с начала первого обращения (в секундах).

Данных СУБД (5 минут)

Объем данных, переданных данным клиентским соединением между сервером «1С:Предприятия» и сервером баз данных, за последние 5 минут (в байтах).

Данных СУБД (всего)

Объем данных, переданных данным клиентским соединением между сервером «1С:Предприятия» и сервером баз данных, с начала данного сеанса (в байтах).

Время вызова (текущее)

Текущее время исполнения последнего незавершенного серверного вызова.

Время вызовов (5 мин.)

Время обращения этого соединения к серверу за последние пять минут.

Время вызовов (всего)

Общее время серверных вызовов в секундах с момента старта клиентского соединения (в секундах).

Количество вызовов (5 мин.)

Количество обращений этого соединения к серверу за последние пять минут.

Количество вызовов (всего)

Количество обращений этого соединения к серверу с начала клиентского соединения (в единицах).

Объем данных (5 мин.)

Объем переданных и полученных данных с момента начала клиентского соединения за последние пять минут (в байтах).

Объем данных (всего)

Объем переданных и полученных данных с момента начала клиентского соединения (в байтах).

Память (текущая)

Объем оперативной памяти, занятый с начала выполнения текущего вызова (в байтах).

Память (5 мин)

Объем оперативной памяти, занятый в процессе вызовов за последние 5 минут (в байтах).

Память (всего)

Объем оперативной памяти, занятый в процессе вызовов с момента начала сеанса (в байтах).

Чтение (текущее)

Объем данных, прочитанных с диска с начала выполнения текущего вызова (в байтах).

Чтение (5 мин)

Объем данных, прочитанных с диска данным сеансом за последние 5 минут (в байтах).

Чтение (всего)

Объем данных, прочитанных с диска данным сеансом с момента начала сеанса (в байтах).

Запись (текущая)

Объем данных, записанных на диск с начала выполнения текущего вызова (в байтах).

Запись (5 мин)

Объем данных, записанных на диск данным сеансом за последние 5 минут (в байтах).

Запись (всего)

Объем данных, записанных на диск данным сеансом с момента начала сеанса (в байтах).

Лицензия

Краткая информация о клиентской лицензии, используемой данным сеансом. Описание формируемой строки см. [здесь](#).

5.2.10.1. Просмотр свойств сеанса

Для просмотра свойств сеанса следует выбрать требуемый сеанс в списке сеансов и выполнить команду контекстного меню [Свойства](#) или аналогичную команду главного меню утилиты.

В результате выполнения команды на экране появится диалог свойств сеанса.

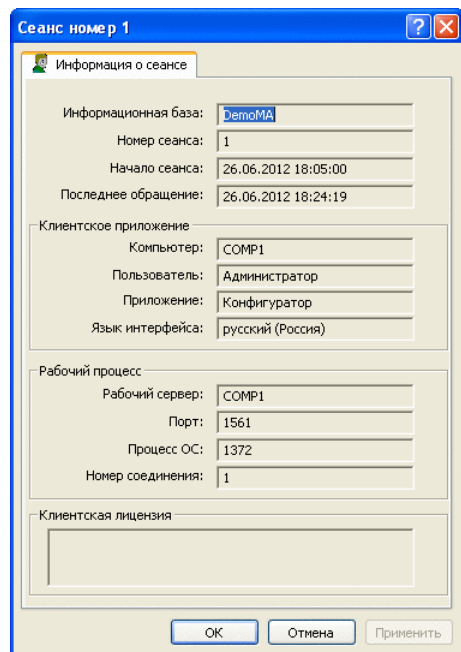


Рис. 59. Свойства сеанса

Диалог свойств сеанса содержит следующую информацию (все свойства сеанса недоступны для редактирования):

Информационная база

Имя информационной базы, с которой установлен сеанс.

Номер сеанса

Номер сеанса.

Начало сеанса

Время, в которое был создан данный сеанс.

Последнее обращение

Время, в которое была зафиксирована последняя активность сеанса.

Компьютер

Сетевое имя компьютера пользователя, клиентское приложение с которого инициировало создание сеанса. Имя компьютера будет пустым, если сеанс создан для обслуживания веб-клиента, тонкого клиента, подключенного через веб-сервер, или Web-сервиса.

Пользователь

Имя пользователя информационной базы.

Приложение

Режим запуска клиентского приложения (см. [здесь](#)).

Язык интерфейса

Язык локализации клиентского приложения.

Рабочий сервер

Имя сервера кластера, с которым установлено соединение.

Порт

Номер сетевого порта рабочего процесса, обслуживающего это соединение.

Процесс ОС

Номер процесса рабочего сервера (в терминах операционной системы), обслуживающего данный сеанс.

Номер соединения

Номер соединения, которое подключено к данному сеансу.

Клиентская лицензия

В данном поле отображается информация о том, какая клиентская лицензия используется данным сеансом и приводится характеристика этой лицензии. Описание формируемой строки см. [здесь](#).

5.2.10.2. Завершение сеанса

Для завершения сеанса следует выбрать требуемый сеанс в списке сеансов и выполнить команду контекстного меню [Удалить](#) или аналогичную команду главного меню утилиты. Если в момент завершения сеанса этот сеанс назначен соединению, то выполняется попытка разрыва соединения (см. [здесь](#)).

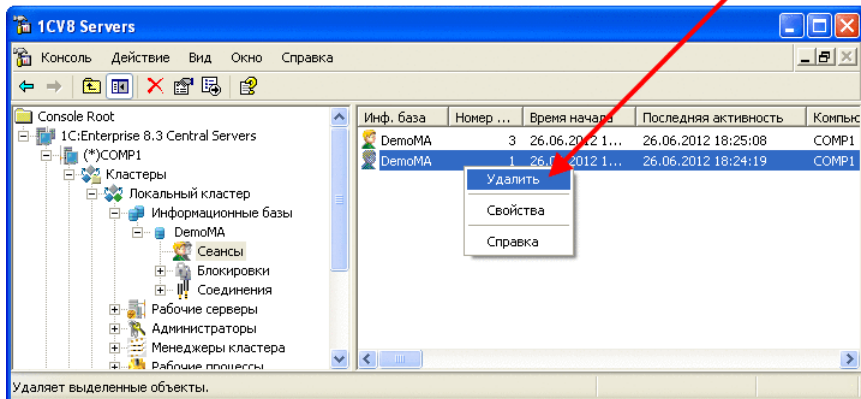


Рис. 60. Завершение сеанса

ВНИМАНИЕ! К использованию данного средства следует подходить с осторожностью, так как завершение сеанса пользователя, активно работающего с информационной базой, может привести к потере данных.

Для завершения сеанса необходимо обладать полномочиями администратора кластера серверов (см. [здесь](#)).

5.2.11. Работа со списком соединений

Список соединений может быть отображен несколькими способами:

- для всего кластера в целом,
- отдельного рабочего процесса кластера,
- отдельной информационной базы,
- отдельного рабочего процесса сервера кластера.

Чтобы отобразить список соединений для всего кластера в целом, следует в дереве центральных серверов выбрать нужный центральный сервер, нужный кластер, зарегистрированный на данном сервере, а затем выбрать и раскрыть ветку **Соединения**.

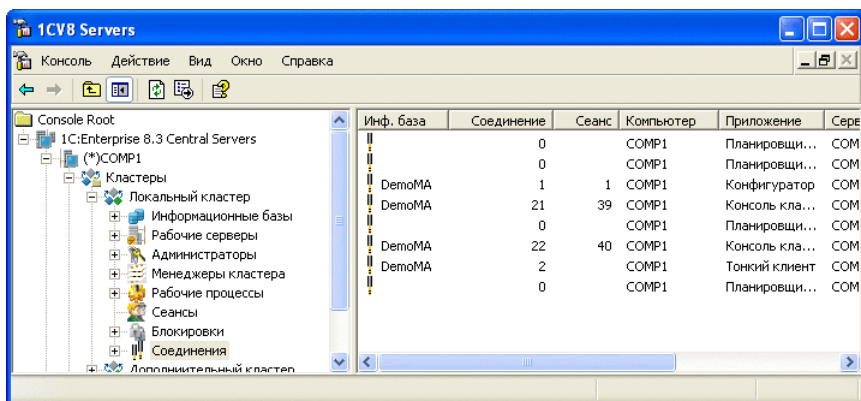


Рис. 61. Соединения кластера

Чтобы отобразить список соединений для отдельного рабочего процесса кластера, следует в дереве центральных серверов выбрать нужный центральный сервер, нужный кластер, ветку **Рабочие процессы**, нужный рабочий процесс, а затем выбрать и раскрыть ветку **Соединения**.

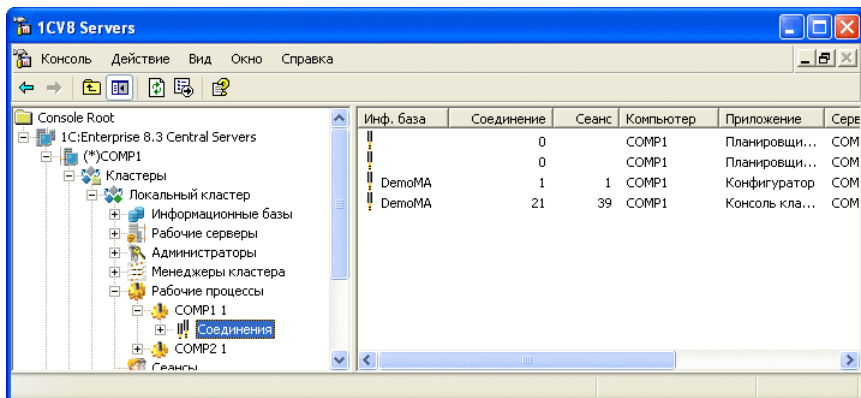


Рис. 62. Соединения рабочего процесса

Чтобы отобразить список соединений для отдельной информационной базы, следует в дереве центральных серверов выбрать нужный центральный сервер, нужный кластер, нужную информационную базу, а затем выбрать и раскрыть ветку **Соединения**.

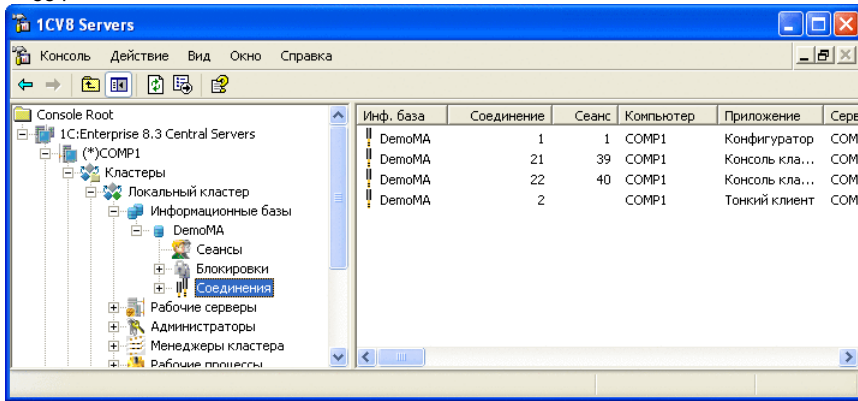


Рис. 63. Соединения информационной базы

Чтобы отобразить список соединений для отдельного рабочего процесса сервера кластера, следует в дереве центральных серверов выбрать нужный центральный сервер, нужный кластер, нужный сервер кластера, нужный рабочий процесс, а затем выбрать и раскрыть ветку **Соединения**.

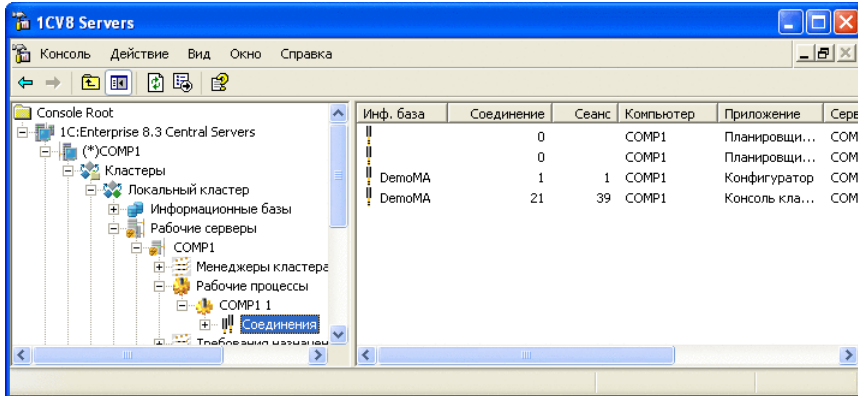


Рис. 64. Соединения рабочего процесса

В дереве центральных серверов соединения не отображаются. В поле свойств отображается список соединений, содержащий следующую информацию:

Инф. база

Имя информационной базы, с которой установлено соединение. Данное поле пустое для служебных соединений.

Соединение

Номер соединения. Номер каждого следующего соединения с информационной базой на 1 больше, чем номер предыдущего соединения. Новое соединение получает номер 1, только если до него с информационной базой не было ни одного соединения. Номера 0 имеют только служебные соединения, не связанные ни с какой информационной базой. Таким образом, как в файловом, так и в клиент-серверном варианте нумерация соединений начинается с 1, только после того, как от информационной базы отсоединятся все клиенты, включая регламентные и фоновые задания.

Сеанс

Номер сеанса, с которым связано выбранное соединение.

Компьютер

Сетевое имя компьютера пользователя, установившего соединение.

Приложение

Идентифицирует приложение, использующее данное соединение (см. [здесь](#)).

Сервер

Имя сервера кластера, с которым установлено соединение.

Порт сервера

Номер сетевого порта рабочего процесса, обслуживающего это соединение.

Начало работы

Время, в которое это соединение было установлено.

Если список соединений открыт для отдельной информационной базы, то в поле свойств отображаются дополнительные колонки, позволяющие оперативно анализировать блокировки базы данных.

5.2.11.1. Просмотр свойств соединения

Для просмотра свойств соединения следует выбрать требуемое соединение в списке соединений и выполнить команду контекстного меню **Свойства** или аналогичную команду главного меню утилиты.

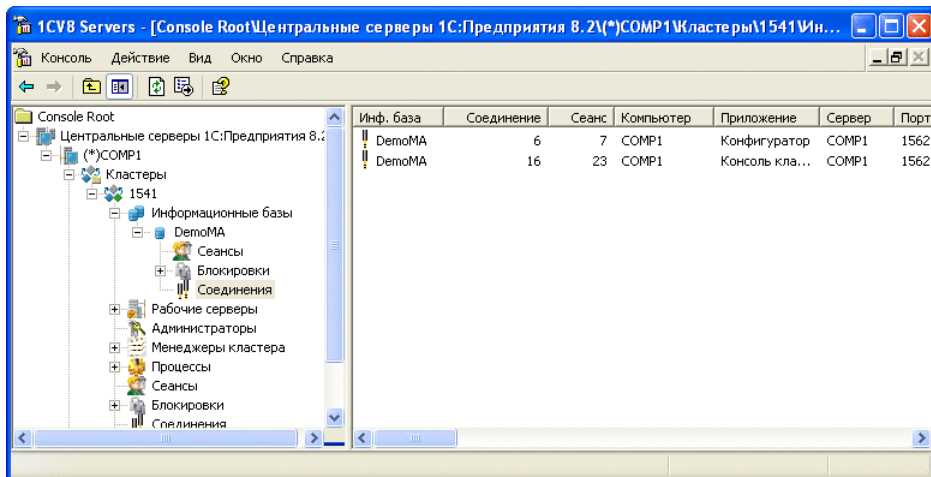


Рис. 65. Список соединений информационной базы

В результате выполнения команды на экране появится диалог свойств соединения.

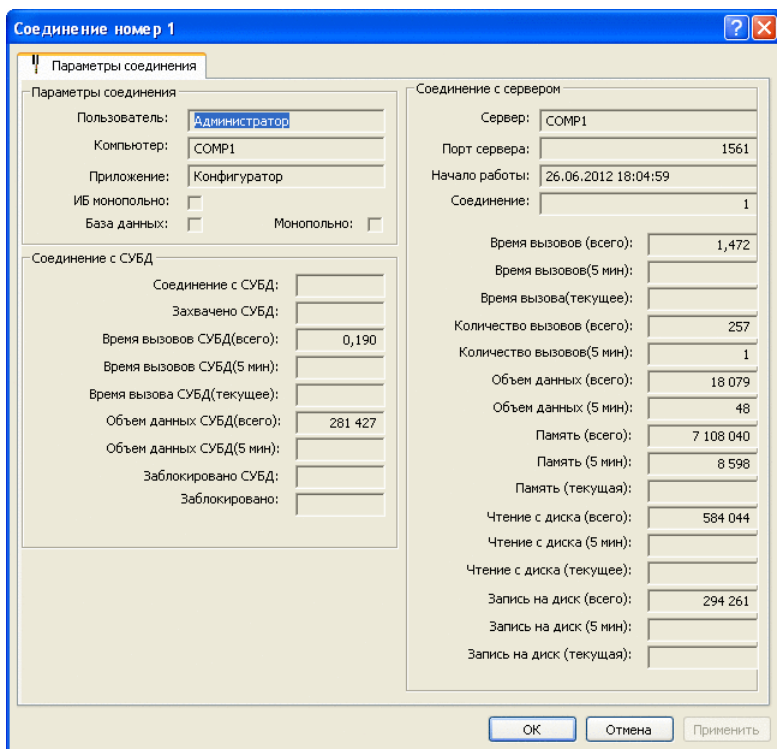


Рис. 66. Свойства соединения

Диалог свойств соединения содержит следующую информацию (все свойства соединения недоступны для редактирования):

Пользователь

Пользователь, от имени которого выполняется подключение с помощью данного соединения.

Компьютер

Имя компьютера, с которого выполнено соединение.

Приложение

Имя приложения (см. [здесь](#)), которое установило соединение с информационной базой.

ИБ монопольно

Признак того, что установлена исключительная блокировка информационной базы (см. [здесь](#)).

База данных

Признак установленного соединения с базой данных.

Монопольно

Признак установки монопольного режима работы с информационной базой.

Сервер

Имя сервера, с которым установлено соединение.

Порт сервера

Сетевой порт сервера, через который происходит взаимодействие.

Начало работы

Время, когда соединение было установлено.

Соединение

Идентификатор текущего соединения.

Соединение с СУБД

Идентификатор процесса сервера баз данных. Отображается в том случае, если в момент открытия диалога свойств соединение выполняет обращение к базе данных.

Захвачено СУБД

Длительность обращения к серверу баз данных на момент открытия диалога свойств. Отображается в том случае, если в момент открытия диалога свойств соединение выполняет обращение к базе данных.

Время вызова СУБД (всего)

Время обращения к СУБД с начала первого обращения (в секундах).

Время вызовов СУБД (5 мин)

Время обращения к СУБД за последние 5 минут (в секундах).

Время вызова СУБД (текущее)

Время текущего обращения СУБД с начала обращения (в секундах).

Объем данных СУБД (всего)

Объем данных, переданных данным клиентским соединением между сервером «1С:Предприятия» и сервером баз данных, с начала данного сеанса (в байтах).

Объем данных СУБД (5 мин)

Объем данных, переданных данным клиентским соединением между сервером «1С:Предприятия» и сервером баз данных, за последние 5 минут (в байтах).

Заблокировано СУБД

Идентификатор процесса, который заблокировал данный процесс.

Заблокировано

Имя пользователя (номер соединения) в том случае, если процесс ожидает освобождения транзакционной блокировки.

Время вызовов (всего)

Общее время серверных вызовов в секундах с момента старта клиентского соединения (в секундах).

Время вызовов (5 мин)

Время обращения этого соединения к серверу за последние 5 минут.

Время вызова (текущее)

Текущее время исполнения последнего незавершенного серверного вызова.

Количество вызовов (всего)

Количество обращений этого соединения к серверу с начала соединения (в единицах).

Количество вызовов (5 мин)

Количество обращений этого соединения к серверу за последние пять минут.

Объем данных (всего)

Объем переданных и полученных данных с момента начала соединения (в байтах).

Объем данных (5 мин)

Объем переданных и полученных данных с момента начала соединения за последние 5 минут (в байтах).

Память (всего)

Объем оперативной памяти, занятый в процессе вызовов с момента начала соединения (в байтах).

Память (5 мин)

Объем оперативной памяти, занятый в процессе вызовов за последние 5 минут (в байтах).

Память (текущая)

Объем оперативной памяти, занятый с начала выполнения текущего вызова (в байтах).

Чтение с диска (текущее)

Объем данных, прочитанных с диска с начала выполнения текущего вызова (в байтах).

Чтение с диска (всего)

Объем данных, прочитанных с диска данным сеансом с момента начала соединения (в байтах).

Чтение с диска (5 мин)

Объем данных, прочитанных с диска данным соединением за последние 5 минут (в байтах).

Запись на диск (всего)

Объем данных, записанных на диск данным соединением с момента начала сеанса (в байтах).

Запись на диск (5 мин)

Объем данных, записанных на диск данным сеансом за последние 5 минут (в байтах).

Объем данных, записанных на диск с начала выполнения текущего вызова (в байтах).

5.2.11.2. Разрыв соединения

Для разрыва соединения следует выбрать требуемое соединение в списке соединений и выполнить команду контекстного меню **Удалить** или аналогичную команду главного меню утилиты.

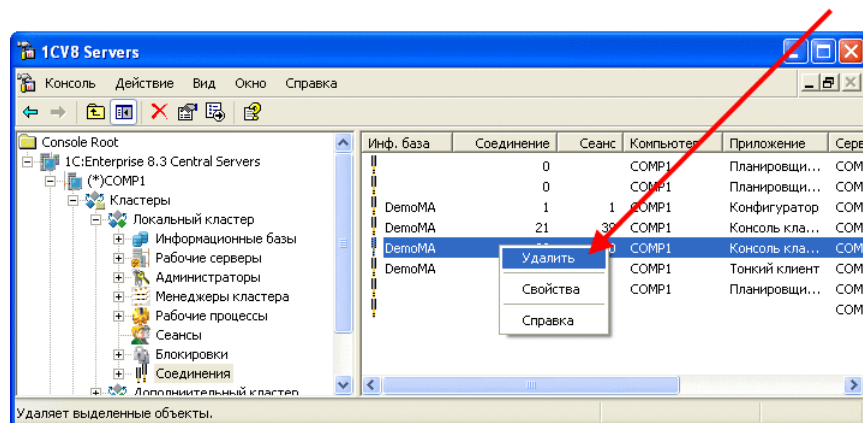


Рис. 67. Удаление соединения

ВНИМАНИЕ! К использованию данного средства следует подходить с осторожностью, так как разрыв соединения пользователя, активно работающего с информационной базой, может привести к потере данных.

Если выполняется длительный запрос к базе данных Microsoft SQL Server, IBM DB2 или Oracle Database, то сервер «1С:Предприятия» выполняет попытку разрыва соединения. Попытка завершается успешно при наличии соответствующих прав у пользователя, от лица которого выполняется подключение к серверу базы данных (подробнее о необходимых правах пользователя базы данных см. в документации на используемую СУБД). В случае успешного завершения разрыва соединения пользователь получит сообщение: **Сеанс работы завершен администратором**. Удаление соединения для базы данных PostgreSQL невозможно. При попытке выполнения команды удаления никаких действий выполнено не будет.

Если клиент выполняет код на сервере «1С:Предприятия», то сервер «1С:Предприятия» выполнит попытку разрыва соединения клиентского приложения и сервера. Для разрыва соединения необходимо обладать полномочиями администратора кластера серверов (см. [здесь](#)) и администратора информационной базы. В случае успешного завершения разрыва соединения пользователь получит сообщение: **Сеанс работы завершен администратором**.

Соединение не может быть разорвано тогда, когда сервер «1С:Предприятия» исполняет клиентский вызов и при этом исполнение не покидает одну строку кода на встроенном языке, кроме случая обращения к СУБД. Например, соединение не может быть разорвано при выполнении долгого вызова метода COM-объекта или выполнении http-вызова из кода на встроенном языке.

5.2.12. Работа со списком блокировок

Список блокировок может быть отображен несколькими способами:

- для всего кластера в целом (все блокировки или по соединениям);
- для отдельной информационной базы (все блокировки или по соединениям).

Чтобы отобразить список блокировок для всего кластера в целом, следует в дереве центральных серверов выбрать нужный центральный сервер, нужный кластер, зарегистрированный на данном сервере, а затем выбрать и раскрыть ветку **Блокировки**.

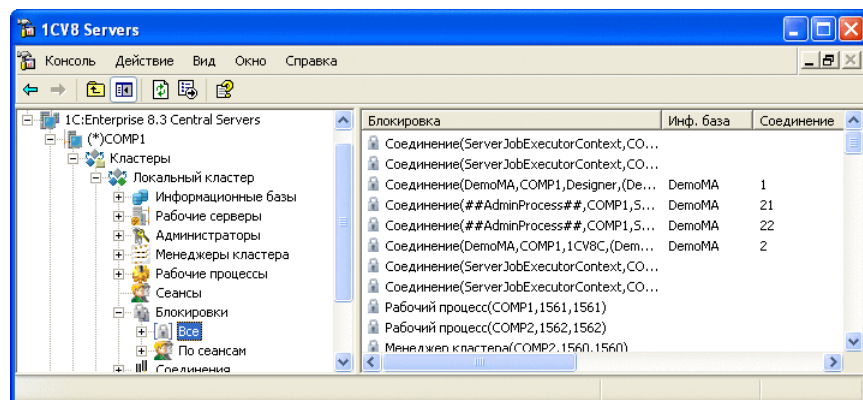


Рис. 68. Список блокировок кластера

Если затем выбрать ветку **Все**, то будет отображен список всех блокировок кластера.

Также можно раскрыть ветку **По сеансам** и выбрать требуемый сеанс. В этом случае будет отображен список блокировок для выбранного сеанса.

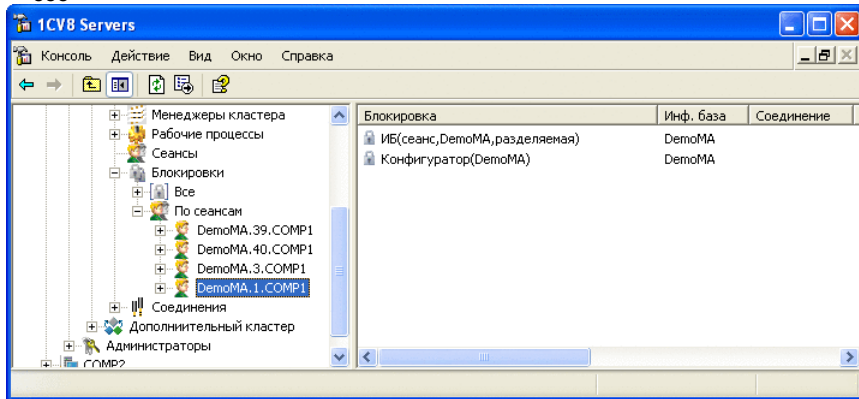


Рис. 69. Список блокировок кластера по сеансам

Для отображения списка блокировок отдельной информационной базы следует в дереве центральных серверов выбрать нужный центральный сервер, нужный кластер, нужную информационную базу, а затем выбрать и раскрыть ветку **Блокировки**.

Если затем выбрать ветку **Все**, то будет отображен список всех блокировок данной информационной базы.

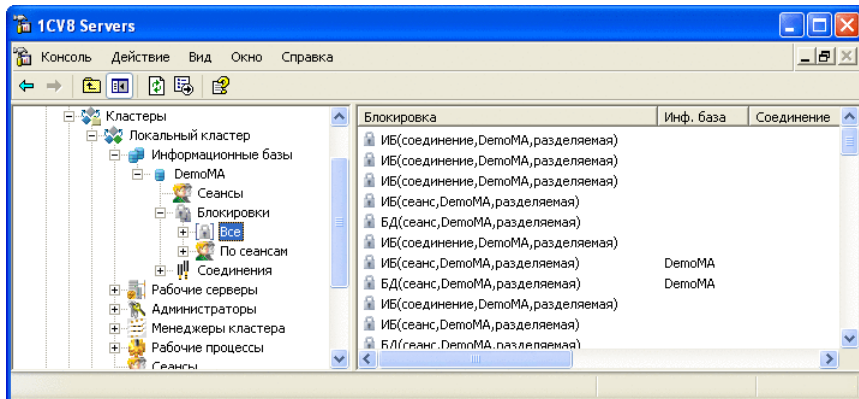


Рис. 70. Список всех блокировок

Также можно раскрыть ветку **По сеансам** и выбрать требуемый сеанс. В этом случае будет отображен список блокировок для выбранного сеанса.

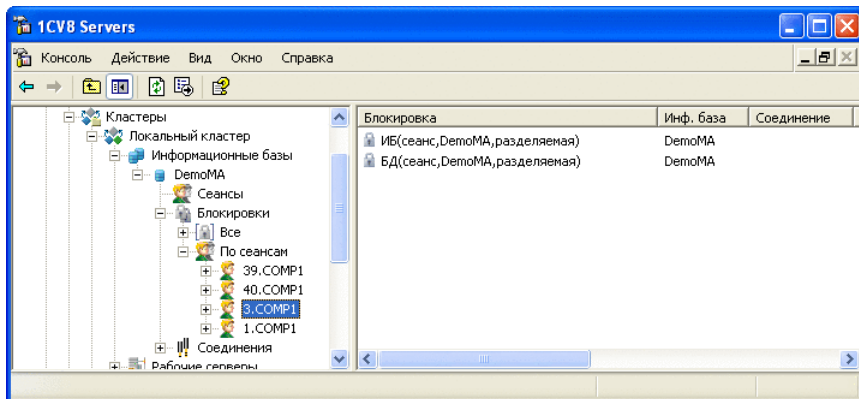


Рис. 71. Список блокировок по сеансам

Если выбран просмотр списка блокировок для конкретного соединения, то дерево центральных серверов содержит перечень соединений. Каждое соединение идентифицируется номером и именем пользовательского компьютера.

В поле свойств отображается список блокировок, содержащий следующую информацию:

Блокировка

Содержит представление вида блокировок и их основные параметры. Существуют следующие виды блокировок:

- Блокировки информационной базы:
 - **БД** – блокировка базы данных «1С:Предприятия». Параметры:
 - источник блокировки (сеанс или соединение);
 - имя информационной базы;
 - разделяемая или исключительная.
 - **ИБ** – блокировка информационной базы. Параметры:
 - источник блокировки (сеанс или соединение);
 - имя информационной базы;
 - разделяемая или исключительная.
 - **Конфигуратор** – исключительная блокировка конфигуратора. Параметры:
 - имя информационной базы.

- **Объект БД** – исключительная блокировка объекта «1С:Предприятия». Параметры:
 - имя информационной базы.
- Блокировки кластера:
 - **Менеджер кластера** – активность процесса менеджера кластера. Параметры:
 - имя сервера;
 - порты процесса менеджера кластера.
 - **Рабочий процесс** – активность рабочего процесса «1С:Предприятия». Параметры:
 - имя сервера;
 - порты рабочего процесса кластера.
 - **Соединение** – соединение с рабочим процессом кластера по TCP или регламентное задание. Параметры:
 - имя серверного контекста (может совпадать с именем информационной базы);
 - имя компьютера и идентификатор приложения, с которого установлено соединение;
 - имена информационной базы и номера соединений, если соединение ассоциировано с одной или несколькими информационными базами.

Инф. база

Имя информационной базы, к которой относится блокировка. Пустое, если блокировка не имеет отношения к информационной базе.

Соединение

Номер соединения с информационной базой. Может быть пустым, если:

- блокировка не относится к информационной базе;
- либо источником блокировки является сеанс, не назначенный ни одному соединению.

Сеанс

Номер сеанса, который установил блокировку. Может быть пустым, если:

- блокировка не относится к информационной базе;
- либо источником блокировки является соединение, которому не назначен сеанс.

Компьютер

Имя клиентского компьютера, с которого установлена блокировка. Пусто, если источником блокировки является серверный процесс.

Приложение

Имя клиентского приложения, установившего блокировку. Пусто, если источником блокировки является серверный процесс.

Сервер

Имя сервера рабочего процесса, ответственного за блокировку. Пусто, если источником блокировки является серверный процесс или если источником блокировки является сеанс, которому не назначено соединение.

Порт сервера

Сетевой порт рабочего процесса, ответственного за блокировку. Пусто, если источником блокировки является серверный процесс или если источником блокировки является сеанс, которому не назначено соединение.

Заблокировано в

Момент времени, когда блокировка была установлена.

5.2.13. Работа со списком профилей безопасности

Для отображения списка информационных баз, зарегистрированных в кластере, следует в дереве центральных серверов выбрать нужный сервер, выбрать нужный кластер, зарегистрированный на данном сервере, а затем выбрать и раскрыть ветку [Профили безопасности](#).

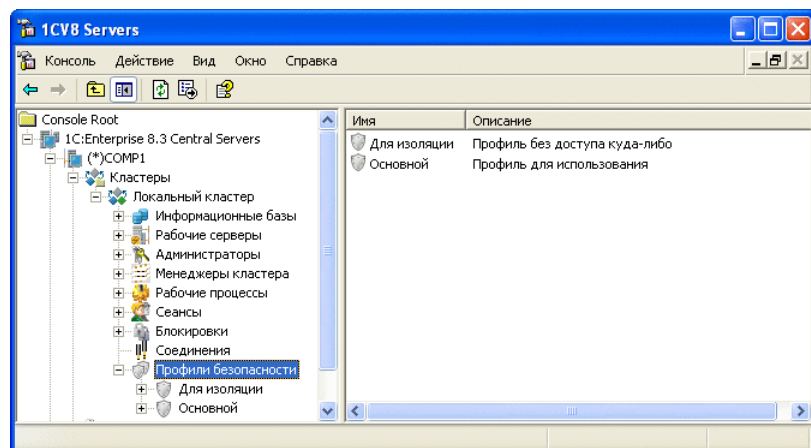


Рис. 72. Список профилей безопасности

Дерево центральных серверов содержит перечень профилей безопасности кластера. Каждый профиль безопасности идентифицируется именем. В поле свойств отображается список профилей безопасности выбранного кластера, содержащий имя профиля и его описание.

5.2.13.1. Добавление профиля

Для добавления нового профиля безопасности в кластер следует выбрать в дереве центральных серверов требуемый центральный сервер, выбрать требуемый кластер, зарегистрированный на этом сервере, выбрать ветку **Профили безопасности** и выполнить команду контекстного меню **Создать – Профиль безопасности** или аналогичную команду главного меню утилиты.

В результате выполнения команды на экране появится диалог свойств профиля безопасности.

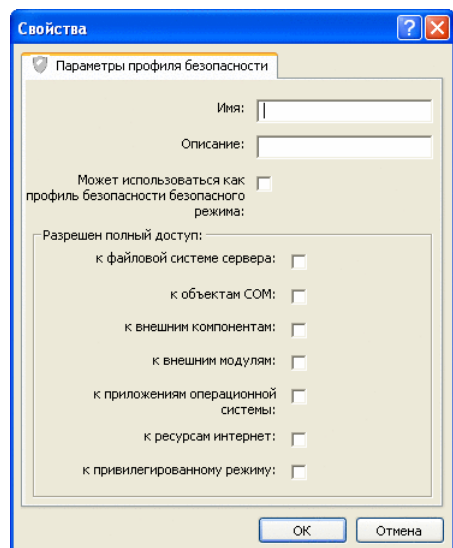


Рис. 73. Новый профиль безопасности

В поля диалога необходимо ввести следующие данные:

Имя

Имя профиля безопасности. Имя должно быть уникальным в пределах кластера.

Описание

Произвольное описание профиля безопасности.

Может использоваться как профиль безопасности безопасного режима

Указывает, что имя данного профиля может быть указано в качестве значения параметра `БезопасныйРежим` метода глобального контекста `УстановитьБезопасныйРежим()`, методов `Создать()` и `Подключить()` менеджера внешних обработок, методов `Создать()` и `Подключить()` менеджера внешних отчетов, а также получить в качестве результата работы функции глобального контекста `БезопасныйРежим()`.

к файловой системе сервера:

Определяет возможность доступа из прикладного решения к файловым ресурсам компьютера, на котором работает сервер «1С:Предприятия». Подробнее об этом параметре см. [здесь](#).

к объектам СОМ:

Определяет возможность взаимодействия прикладного решения с СОМ-объектами компьютера, на котором работает сервер «1С:Предприятие». Для серверов, работающих под управлением ОС Linux, данный параметр не имеет смысла. Подробнее об этом параметре см. [здесь](#).

к внешним компонентам:

Определяет возможность прикладного решения взаимодействовать с внешними компонентами на стороне сервера «1С:Предприятия». Подробнее об этом параметре см. [здесь](#).

к внешним модулям:

Определяет возможность использования в прикладном решении внешние модули (внешние отчеты, обработки и расширения конфигурации), а также оператору `Выполнить()` и функции `Вычислить()`. Подробнее об этом параметре см. [здесь](#).

к приложениям операционной системы:

Определяет доступность для прикладного решения приложений операционной системы на стороне сервера «1С:Предприятия». Перечень приложений является зависимым от операционной системы, под управлением которой работает сервер «1С:Предприятия». Подробнее об этом параметре см. [здесь](#).

к ресурсам интернет:

Определяет возможность взаимодействия кода прикладного решения, исполняемого на сервере «1С:Предприятия», с ресурсами сети Интернет. Подробнее об этом параметре см. [здесь](#).

к привилегированному режиму:

Определяет возможность управления привилегированным режимом из кода прикладного решения, работающего на стороне сервера «1С:Предприятия» в том случае, если данный профиль указан в качестве профиля безопасности безопасного режима.

5.2.13.2. Просмотр и изменения свойств профиля

Для просмотра и изменения свойств профиля безопасности следует выбрать требуемый профиль в списке профилей безопасности кластера и выполнить команду контекстного меню **Свойства** или аналогичную команду главного меню утилиты.

В результате выполнения команды на экране появится диалог свойств профиля безопасности.

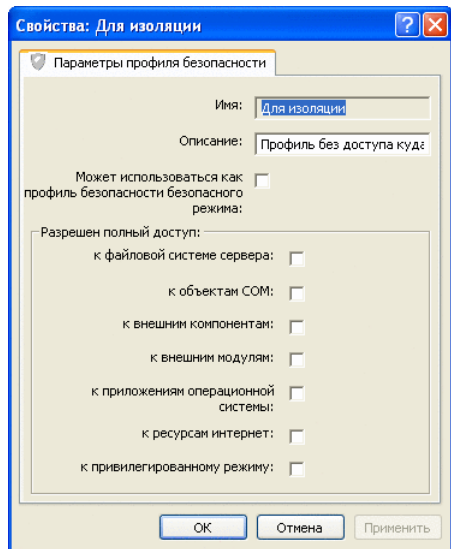


Рис. 74. Свойства профиля безопасности

Некоторые параметры профиля позволяют задавать исключения из общего правила запрета. Например, можно запретить использование всех каталогов файловой системы сервера, кроме некоторых, которые следует указать в списке исключений.

Для того, чтобы задать исключение из какого-либо ограничения профиля безопасности, следует встать на нужный элемент, подчиненный выбранному виртуальному каталогу, и в контекстном меню выбрать команду **Создать – Имя создаваемого элемента**, например **Создать – Виртуальный каталог**.

В открывшемся окне свойство **Описание** предназначено для описания создаваемого элемента, а остальные параметры описаны в соответствующем разделе общего описания профиля безопасности (см. [здесь](#)).

5.2.13.3. Удаление профиля

Для удаления профиля безопасности следует выбрать необходимый профиль в списке профилей безопасности и выполнить команду контекстного меню **Удалить** или аналогичную команду главного меню утилиты.

5.3. Программные средства администрирования кластера серверов

5.3.1. Доступ к кластеру серверов из встроенного языка

Программный интерфейс администрирования кластера серверов «1С:Предприятия» описан в синтакс-помощнике, в разделе **Средства интеграции и администрирования – COM-соединитель – Администрирование кластера серверов**.

Для администрирования кластера серверов используются два объекта: **Соединение с агентом сервера** и **Соединение с рабочим процессом**.

Соединение с агентом сервера может быть получено при помощи метода `ConnectAgent()` объекта COM-соединитель:

```
COMСоединитель = Новый СОМОбъект("V83.COMConnector");
СоединениеСАгентомСервера = COMСоединитель.ConnectAgent("TestSrv");
```

[Копировать в буфер обмена](#)

Соединение с агентом сервера позволяет выполнять следующие действия:

- аутентификация, добавление, удаление, получение списка администраторов центрального сервера и администраторов кластера;
- создание, удаление, получение списка кластеров;
- создание, удаление, получение списка серверов;
- создание, удаление, получение списка рабочих процессов кластера;
- получение списка сервисов кластера;
- получение списка сеансов информационной базы;
- получение списка соединений кластера;
- получение списка соединений информационной базы;
- получение списка информационных баз, зарегистрированных в кластере;
- получение списка блокировок кластера;
- получение другой информации.

Соединение с рабочим процессом может быть получено при помощи метода `ConnectWorkingProcess()` объекта COM-соединитель:

```
COMСоединитель = Новый СОМОбъект("V83.COMConnector");
СоединениеСРабочимПроцессом = COMСоединитель.ConnectWorkingProcess("TestSrv:1562");
```

[Копировать в буфер обмена](#)

Соединение с рабочим процессом позволяет выполнять следующие действия:

- аутентификация пользователей информационных баз;
- создание, удаление, получение списка информационных баз, зарегистрированных в кластере;
- получение списка соединений информационной базы;
- разрыв соединения информационной базы;
- соединение с информационной базой (COM-соединение);
- получение другой информации.

Подробнее про программное администрирование можно посмотреть во встроенной справке: **Встроенный язык – Средства интеграции и администрирования – COM-соединитель**

5.3.2. Внешнее управление сеансами

Для внешнего управления сеансами необходимо реализовать Web-сервис, который будет предоставлять определенный набор методов и логику управления. Этот Web-сервис может быть реализован как средствами системы «1С:Предприятие», так и сторонними средствами.

Далее будет дано описание Web-сервиса и приведен пример реализации средствами «1С:Предприятие».

5.3.2.1. Описание Web-сервиса

Имя Web-сервиса не специфицируется. Тайм-аут на выполнение методов равен 5 секунд (по умолчанию) или задается параметром `tout` при вызове Web-сервиса.

5.3.2.1.1. Версия 1

onStartSession

Описание:

Метод вызывается кластером серверов при начале сеанса (кроме сеанса фонового задания и сеанса WS-соединения). Метод должен определить, допустимо или нет создание нового сеанса с указанными параметрами. Решение сообщается кластеру серверов в виде кода возврата.

Параметры:

CallNo входной

Тип: **Число**. Порядковый номер вызова. При каждом вызове Web-сервиса кластером порядковый номер вызова увеличивается на 1. На основании значения `CallNo` Web-сервис может запросить синхронизацию данных о сеансах.

Номер вызова уникален в разрезе информационной базы.

ClusterID входной

Тип: **Строка**. Идентификатор кластера серверов. Содержит строковое представление уникального идентификатора (UUID).

ClusterName входной

Тип: **Строка**. Имя кластера серверов, в котором расположена информационная база.

InfoBaseName входной

Тип: **Строка**. Имя информационной базы, для работы с которой создается сеанс.

SessionID входной

Тип: **Строка**. Идентификатор сеанса. Содержит строковое представление уникального идентификатора (UUID).

UserID входной

Тип: **Строка**. Идентификатор пользователя. Содержит строковое представление уникального идентификатора (UUID).

UserName входной

Тип: **Строка**. Имя пользователя.

AppID входной

Тип: **Строка**. Имя приложения, пытающегося получить доступ к информационной базе. Подробную информацию о значениях этого параметра можно получить в описании функции глобального контекста `ПредставлениеПриложения()`.

Zone входной

Тип: **Строка**. Содержит начальные значения разделителей для создаваемого сеанса. Строка передается в формате, который указывается для ключа командной строки запуска приложения `/Z`.

LanguageCode входной

Тип: **Строка**. Код языка сообщений для создаваемого сеанса.

ErrorDescription выходной

Тип: **Строка**. Содержит описание причины запрета создания сеанса в виде, пригодном для чтения и понимания пользователем. Заполняется в случае, если метод Web-сервиса возвращает значение 1.

Возвращаемое значение:

Тип: **Число**. Возвращаемое значение может быть одним из следующих:

- 0 – сеанс может быть создан.
- 1 – сеанс не может быть создан. Пользователю, начинающему сеанс, выдается сообщение об ошибке входа с текстом исключения из параметра `ErrorDescription`, или с текстом **Начало сеанса запрещено сервисом внешнего управления сеансами**, если значение параметра `ErrorDescription` равно пустой строке.
- 2 – требуется выполнить синхронизация. В этом случае кластер серверов вызовет метод `synchronize` со списком сеансов, не включающим этот сеанс, после чего повторит вызов метода `onStartSession`. Вызовы внешнего управления сеансами не синхронизируются и определение необходимости синхронизации списка сеансов возложено на сервис внешнего управления сеансами. В качестве исходной информации может быть использован порядковый номер вызова (параметр `CallNo`). Пропуск некоторых номеров вызовов после некоторого ожидания может считаться условием, требующим синхронизации.

onFinishSession

Описание:

Метод вызывается при завершении работы сеанса.

Параметры:

CallNo входной

Тип: **Число**. Порядковый номер вызова.

Номер вызова уникален в разрезе информационной базы.

ClusterID входной

Тип: **Строка**. Идентификатор кластера серверов. Содержит строковое представление уникального идентификатора (UUID).

SessionID входной

Тип: **Строка**. Идентификатор сеанса. Содержит строковое представление уникального идентификатора (UUID).

Возвращаемое значение:

Отсутствует.

synchronize

Описание:

Предназначен для синхронизации данных о созданных сеансах между кластером серверов и Web-сервисом, реализующим механизм внешнего управления сеансами. Вызывает кластером серверов в случае, если метод `onStartSession()` вернул значение 2.

Параметры:

CallNo входной

Тип: **Число**. Порядковый номер вызова.

Номер вызова уникален в разрезе информационной базы.

ClusterID входной

Тип: **Строка**. Идентификатор кластера серверов. Содержит строковое представление уникального идентификатора (UUID).

ClusterName входной

Тип: **Строка**. Имя кластера серверов, в котором расположена информационная база.

InfoBaseName входной

Тип: **Строка**. Имя информационной базы, для которой выполняется синхронизация информации о количестве сеансов.

CurrentSessions входной

Тип: **Sessions**. Объект содержит данные обо всех сеансах, которые созданы для конкретной информационной базы с использованием механизма внешнего управления сеансами. Объект содержит свойство `Content`, который является коллекцией объектов `Session`, которые описывают один сеанс. Коллекция может быть пустой.

Описание объекта `Sessions` (в формате XSD):

[Копировать в буфер обмена](#)

```
<xs:schema xmlns:ns1="http://v8.lc.ru/8.1/data/core" xmlns:tns="http://v8.lc.ru/SessionManagement" xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://v8.lc.ru/8.1/data/core">
  <xs:import namespace="http://v8.lc.ru/8.1/data/core"/>
  <xs:complexType name="Session">
    <xs:sequence>
      <xs:element name="SessionID" type="xs:string"/>
      <xs:element name="UserID" type="xs:string"/>
      <xs:element name="UserName" type="xs:string"/>
      <xs:element name="AppID" type="xs:string"/>
      <xs:element name="Zone" type="xs:string"/>
      <xs:element name="LanguageCode" type="xs:string"/>
      <xs:element name="Hibernate" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Sessions">
    <xs:sequence>
      <xs:element name="Content" type="tns:Session" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Возвращаемое значение:

Отсутствует.

5.3.2.1.2. Версия 2

Дополнительно к возможностям, предоставляемых версией 1, версия 2 содержит два метода, которые позволяют учитывать переход сеанса в состояние **Спящий** и возврат его в рабочее состояние (подробнее см. [здесь](#)).

onHibernateSession

Описание:

Метод вызывается кластером серверов при переводе сеанса в спящее состояние. При переводе сеанса в спящее состояние сеанс перестает занимать лицензию (см. [здесь](#)), что необходимо учесть в данных Web-сервиса. При начале работы сеанса, который переводится в состояние **Спящий**, кластер серверов вызывал метод `onStartSession` данного Web-сервиса.

Параметры:

CallNo входной

Тип: **Число**. Порядковый номер вызова.

Номер вызова уникален в разрезе информационной базы.

ClusterID входной

Тип: **Строка**. Идентификатор кластера серверов. Содержит строковое представление уникального идентификатора (UUID).

SessionID входной

Тип: **Строка**. Идентификатор сеанса. Содержит строковое представление уникального идентификатора (UUID).

Возвращаемое значение:

Отсутствует.

Описание:

Параметры:

CallNo входной

Тип: **Число**. Порядковый номер вызова. При каждом вызове Web-сервиса кластером порядковый номер вызова увеличивается на 1. На основании значения *CallNo* Web-сервис может запросить синхронизацию данных о сеансах.

Номер вызова уникален в разрезе информационной базы.

ClusterID входной

Тип: **Строка**. Идентификатор кластера серверов. Содержит строковое представление уникального идентификатора (UUID).

SessionID входной

Тип: **Строка**. Идентификатор сеанса. Содержит строковое представление уникального идентификатора (UUID).

ErrorDescription выходной

Тип: **Строка**. Содержит описание причины запрета перевода сеанса в рабочее состояние в виде, пригодном для чтения и понимания пользователем. Заполняется в случае, если метод Web-сервиса возвращает значение 1.

Возвращаемое значение:

Тип: **Число**. Возвращаемое значение может быть одним из следующих:

- 0 – сеанс может быть переведен в рабочее состояние.
- 1 – сеанс не может быть переведен в рабочее состояние. Пользователю, чей сеанс пытается перейти в рабочее состояние, выдается сообщение об ошибке входа с текстом исключения из параметра *ErrorDescription*, или с текстом **Пробуждение сеанса запрещено сервисом внешнего управления сеансами**, если значение параметра *ErrorDescription* равно пустой строке.
- 2 – требуется выполнить синхронизация. В этом случае кластер серверов вызовет метод *synchronize* со списком сеансов, не включающим этот сеанс, после чего повторит вызов метода *onWakeUpSession*. Вызовы внешнего управления сеансами не синхронизируются и определение необходимости синхронизации списка сеансов возложено на сервис внешнего управления сеансами. В качестве исходной информации может быть использован порядковый номер вызова (параметр *CallNo*). Пропуск некоторых номеров вызовов после некоторого ожидания может считаться условием, требующим синхронизации.

5.3.2.2. Пример реализации

Рассмотрим пример Web-сервиса внешнего управления сеансами.

ПРИМЕЧАНИЕ. Пример, приведенный в данном разделе, не является законченным. Он предназначен для демонстрации работы механизма.

Данный пример будет реализовывать следующую схему работы:

- Администратор может ограничиваться количество одновременных сеансов для информационной базы. Для этого указывается название информационной базы и указывается количество одновременных сеансов.
- Если информационной базы нет в списке – количество сеансов с ней не ограничено.
- В процессе работы в списке можно посмотреть, какие сеансы существуют у данной информационной базы.

В качестве Web-сервиса используется простая конфигурация, которая содержит один справочник и один Web-сервис. Справочник содержит список информационных баз, ограничение по количеству сеансов и порядковый номер последнего вызова. Web-сервис выполняет действия по учету сеансов.

Справочник имеет следующие характеристики:

- Имя: *ДоступныеСеансы*.
- Длина кода: 0.
- Длина наименования: 50.
- Реквизиты:
 - Имя: *Количество*, тип *Число(2)*, неотрицательное.
 - Имя: *ПоследнийНомерВызова*, тип *Число(10)*, неотрицательное.
- Табличная часть:
 - Имя: *ТекущиеСеансы*.
 - Реквизиты:
 - Имя: *SessionID*, тип *Строка(40)*, переменной длины.
 - Имя: *UserID*, тип *Строка* неограниченной длины.
 - Имя: *UserName*, тип *Строка* неограниченной длины.
 - Имя: *AppID*, тип *Строка* неограниченной длины.
 - Имя: *Zone*, тип *Строка* неограниченной длины.
 - Имя: *LanguageCode*, тип *Строка* неограниченной длины.
 - Имя: *Hibernate*, тип *Булево*.

Данный справочник будет хранить имя информационной базы (стандартный реквизит *Наименование*), максимальное количество одновременных сеансов для этой базы (реквизит *Количество*) и порядковый номер последнего вызова Web-сервиса (реквизит *ПоследнийНомерВызова*). С помощью реквизита *ПоследнийНомерВызова* будет определяться необходимость синхронизации данных между кластером серверов «1С:Предприятия» и Web-сервисом. Табличная часть будет хранить информацию о тех сеансах, которые Web-сервис разрешил начать. В методе *onStartSession()* запись о сеансе будет создаваться, а в методе *onFinishSession()* запись о сеансе будет удаляться. Таким образом, сопоставляя максимальное количество сеансов, разрешенных для данной информационной базы, и количество запомненных сеансов, можно принять решение о возможности создания очередного сеанса. Если используется вторая версия интерфейса, то также будут фиксироваться переводы сеансов в спящий режим и пробуждение сеансов. При переводе сеанса в спящий режим, запись о сеансе не удаляется, а помечается соответствующим образом.

Также нужно создать Web-сервис, который описан выше (см. [здесь](#)), и поместить в модуль Web-сервиса нижеследующий программный код. Web-сервис должен содержать ссылку на пакет XDTO <http://v8.1c.ru/SessionManagement>, описывающий объект `Sessions`. Описание этого пакета см. [здесь](#).

Текст операций Web-сервиса:

Копировать в буфер обмена

```

Функция onStartSession(CallNo, ClusterID, ClusterName, InfoBaseName, SessionID, UserID, UserName, AppID, Zone, LanguageCode, ErrorDescription)
    ErrorDescription = "";
    Результат = Справочники.ДоступныеСеансы.НайтиПоНаименованию(InfoBaseName, Истина);
    Если Результат.Пустая() Тогда
        // Если в справочнике нет базы с таким именем - значит для нее нет ограничений
        Возврат 0;
    КонецЕсли;
    Объект = Результат.ПолучитьОбъект();
    Объект.ПоследнийНомерВызова = Объект.ПоследнийНомерВызова + 1;
    // Проверим - если нарушена последовательность вызовов, то потребуются синхронизация
    Если Объект.ПоследнийНомерВызова <> CallNo Тогда
        Возврат 2;
    КонецЕсли;
    Сеансы = Результат.ТекущиеСеансы.НайтиСтроки(Новый Структура("hibernate", Ложь));
    Если Сеансы.Количество()+1 > Результат.Количество Тогда
        // достигнуто максимальное количество сеансов, больше нельзя
        Объект.Записать();
        ErrorDescription = "Превышено максимальное количество сеансов, разрешенных сервисом внешнего управления сеансами";
        Возврат 1;
    Иначе
        // сеансы еще возможны
        Строка = Объект.ТекущиеСеансы.Добавить();
        Строка.SessionID = SessionID;
        Строка.UserID = UserID;
        Строка.UserName = UserName;
        Строка.AppID = AppID;
        Строка.Zone = Zone;
        Строка.LanguageCode = LanguageCode;
        Строка.Hibernate = Ложь;
        Объект.Записать();
        Возврат 0;
    КонецЕсли;
КонецФункции
Функция onFinishSession(CallNo, ClusterID, SessionID)
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        |     ДоступныеСеансы.Ссылка
        | ИЗ
        |     Справочник.ДоступныеСеансы КАК ДоступныеСеансы
        | ГДЕ
        |     ДоступныеСеансы.ТекущиеСеансы.SessionID = &SessionID";
    Запрос.УстановитьПараметр("SessionID", SessionID);
    РезультатЗапроса = Запрос.Выполнить();
    ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
    Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
        // удалим сеанс из базы сервиса
        Объект = ВыборкаДетальныеЗаписи.Ссылка.ПолучитьОбъект();
        Результат = Объект.ТекущиеСеансы.НайтиСтроки(Новый Структура("SessionID", SessionID));
        Для Каждого Строка Из Результат Цикл
            Объект.ТекущиеСеансы.Удалить(Объект.ТекущиеСеансы.Индекс(Строка));
        КонецЦикла;
        Объект.ПоследнийНомерВызова = CallNo;
        Объект.Записать();
    КонецЦикла;
    Возврат Неопределено;
КонецФункции
Функция Synchronize(CallNo, ClusterID, ClusterName, InfoBaseName, CurrentSessions)
    Результат = Справочники.ДоступныеСеансы.НайтиПоНаименованию(InfoBaseName, Истина);
    Если Результат.Пустая() Тогда
        // Если в справочнике нет базы с таким именем - значит для нее нет ограничений
        Возврат Неопределено;
    КонецЕсли;
    // сеансы еще возможны
    Объект = Результат.ПолучитьОбъект();
    Объект.ТекущиеСеансы.Очистить();
    Для каждого Сеанс Из CurrentSessions.Content Цикл
        Строка = Объект.ТекущиеСеансы.Добавить();
        Строка.SessionID = Сеанс.SessionID;
        Строка.UserID = Сеанс.UserID;
        Строка.UserName = Сеанс.UserName;
        Строка.AppID = Сеанс.AppID;
        Строка.Zone = Сеанс.Zone;
        Строка.LanguageCode = Сеанс.LanguageCode;
    КонецЦикла;
    Объект.ПоследнийНомерВызова = CallNo;
    Объект.Записать();
    Возврат Неопределено;
КонецФункции
Функция onHibernateSession(CallNo, ClusterID, SessionID)
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        |     ДоступныеСеансы.Ссылка
        | ИЗ
        |     Справочник.ДоступныеСеансы КАК ДоступныеСеансы
        | ГДЕ
        |     ДоступныеСеансы.ТекущиеСеансы.SessionID = &SessionID";
    Запрос.УстановитьПараметр("SessionID", SessionID);
    РезультатЗапроса = Запрос.Выполнить();
    ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
    Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
        // пометим сеанс в базе сервиса как спящий
        Объект = ВыборкаДетальныеЗаписи.Ссылка.ПолучитьОбъект();
        Результат = Объект.ТекущиеСеансы.НайтиСтроки(Новый Структура("SessionID", SessionID));
        Для Каждого Строка Из Результат Цикл
            Строка.Hibernate = Истина;
        КонецЦикла;
        Объект.ПоследнийНомерВызова = CallNo;
        Объект.Записать();
    КонецЦикла;
    Возврат Неопределено;
КонецФункции
Функция onWakeUpSession(CallNo, ClusterID, SessionID, ErrorDescription)
    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        |     ДоступныеСеансы.Ссылка
        | ИЗ
        |     Справочник.ДоступныеСеансы КАК ДоступныеСеансы
        | ГДЕ

```

```

ДоступныеСеансы.ТекущиеСеансы.SessionID = &SessionID";
Запрос.УстановитьПараметр("SessionID", SessionID);
РезультатЗапроса = Запрос.Выполнить();
Если РезультатЗапроса.Пустой() Тогда
    ErrorDescription = "Пробуждение невозможно! Старт сеанса с номером " + SessionID + " не зафиксирован сервисом";
    Возврат 1;
КонецЕсли;
ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    // Проверим – если нарушена последовательность вызовов, то потребуется синхронизация
    Если ВыборкаДетальныеЗаписи.Ссылка.ПоследнийНомерВызова+1 <> CallNo Тогда
        Возврат 2;
    КонецЕсли;
    // пометим сеанс в базе сервиса как спящий
    Объект = ВыборкаДетальныеЗаписи.Ссылка.ПолучитьОбъект();
    // получим количество неспящих сеансов
    Неспящих = Объект.ТекущиеСеансы.НайтиСтроки(Новый Структура("Hibernate", Истина)).Количество();
    // начнем просыпаться
    Результат = Объект.ТекущиеСеансы.НайтиСтроки(Новый Структура("SessionID", SessionID));
    Для Каждого Строка Из Результат Цикл
        Если Неспящих > Объект.Количество Тогда
            ErrorDescription = "Пробуждение невозможно! Превышено максимальное количество сеансов, разрешенных сервисом внешнего управления сеансами";
            Возврат 1;
        КонецЕсли;
        Неспящих = Неспящих+1;
        Строка.Hibernate = Ложь;
    КонецЦикла;
    Объект.ПоследнийНомерВызова = CallNo;
    Объект.Записать();
КонецЦикла;
Возврат 0;
КонецФункции

```

После создания конфигурации следует опубликовать Web-сервис на веб-сервере (описано в книге «1С:Предприятие 8.3. Руководство администратора»).

Предположим, что публикация называется *sc*, Web-сервис имеет имя *SessionControl*, пространство имен этого Web-сервиса <http://v8.1c.ru/SessionManagement>, и в состав пакетов XDTO должен входить пакет <http://v8.1c.ru/SessionManagement>. Публикация выполняется на компьютере *localhost*.

После публикации и проверки корректности опубликования Web-сервиса необходимо сформировать строку, указывающую кластеру серверов на созданный Web-сервис. Описание строки см. [здесь](#).

Строка будет иметь вид:

```
wsdl=http://localhost/sc/ws/SessionControl?wsdl;ns=http://v8.1c.ru/SessionManagement;srvc=SessionControl;port=SessionControlSoap;
```

[Копировать в буфер обмена](#)

Также необходимо создать клиент-серверную информационную базу, чьи сеансы будет управлять созданный Web-сервис. Зададим этой информационной базе имя *TestDB*. Это может быть любая информационная база (в том числе и пустая). После создания, с помощью консоли кластера следует указать в качестве значения свойства *Внешнее управление сеансами* созданную выше строку (*wsdl=...*), и установить флажок для свойства *Обязательное использование внешнего управления*.

Затем необходимо запустить информационную базу управления сеансами в режиме *1С:Предприятие* и в справочнике *ДоступныеСеансы* создать один элемент, у которого стандартный реквизит *Наименование* должен принимать значение *TestDB*, а реквизит *Количество* – значение 2. Таким образом, мы определили, что для информационной базы *TestDB* доступно не более 2 одновременных сеансов.

Теперь в информационную базу *TestDB* можно войти не более двух раз.

5.3.3. Сервер администрирования кластера серверов

5.3.3.1. Общая информация

Для администрирования кластера серверов можно использовать специальный сервер администрирования кластера. В состав сервера входит собственно сервер (*ras*) и утилита командной строки (*rac*), позволяющая управлять кластером серверов.

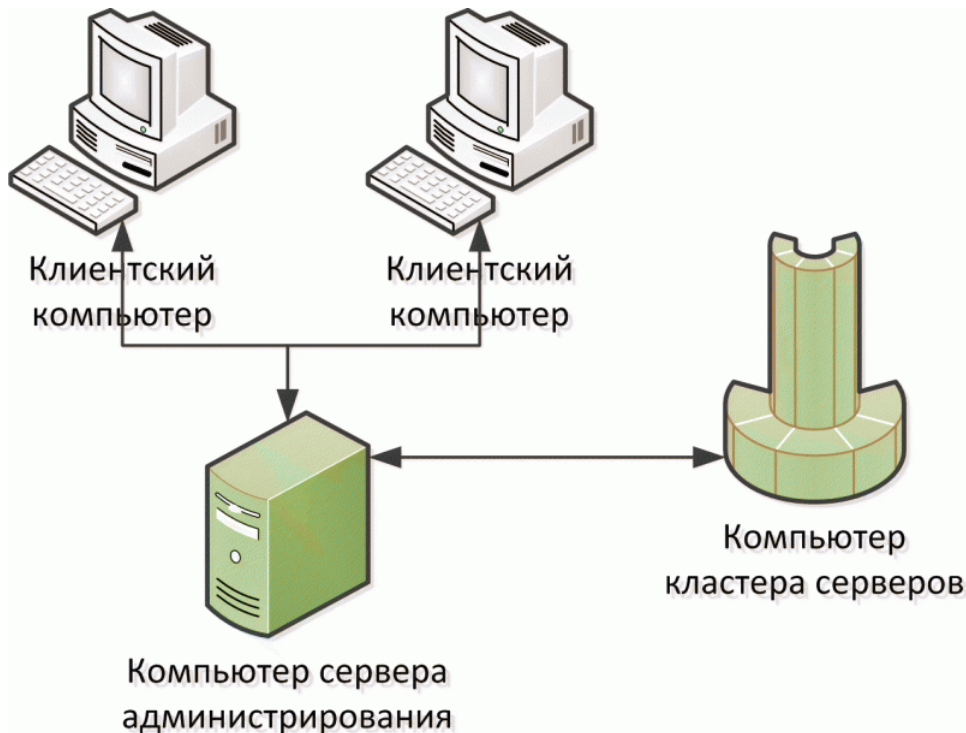


Рис. 75. Сервер администрирования

Версии кластера серверов и сервера администрирования (*ras*) должны совпадать, а при использовании утилиты командной строки (*rac*) следует учитывать следующие ограничения:

- Невозможно использование утилиты командной строки (*rac*) версий 8.3.1 и 8.3.2 с сервером администрирования (*ras*) версий 8.3.3 и старше.

- Невозможно использование утилиты командной строки (**rac**) версии 8.3.3 и старше с сервером администрирования (**ras**) версии 8.3.1 и 8.3.2.

И сервер администрирования и утилита командной строки могут работать в любой поддерживаемой ОС. Одновременно к одному кластеру серверов могут быть подключены несколько серверов администрирования. Один сервер администрирования может взаимодействовать с одним агентом сервера.

Сервер администрирования (**ras**) может выполняться как в режиме приложения, так и в режиме службы ОС Windows или демона ОС Linux. Общая схема работы выглядит следующим образом:

- Запускается сервер администрирования (как приложение или служба/демон).
- Утилита командной строки соединяется с сервером администрирования для выполнения необходимых действий.
- На время выполнения операций, сервер администрирования выполняет подключение к кластеру серверов и после выполнения операций – отключается от кластера. Следовательно, при регламентных операциях на кластере серверов, связанных с его (кластером) остановкой или перезапуском, останавливать сервер администрирования нет необходимости. Исключение составляет случай смены версии кластера серверов. В этом случае необходимо сменить версию сервера администрирования на версию, аналогичную версии кластера серверов.

Сервер администрирования и утилита администрирования входят в состав компонент сервера «1С:Предприятия» при установке системы (см. [здесь](#)).

Для взаимодействия сервера администрирования и утилиты администрирования используется сетевой порт **1545**, который может быть переопределен с помощью параметра `--port` командной строки запуска сервера администрирования (**ras**).

Утилита администрирования позволяет выполнить весь объем работ, необходимый для администрирования кластера серверов. Однако следующие возможности не поддерживаются утилитой:

- Аутентификация ОС для администраторов кластера серверов, рабочего сервера и информационной базы.

Утилита администрирования (**rac**) принимает все необходимые параметры в командной строке и осуществляет вывод информации в стандартный поток вывода (**stdout**). В случае успешного завершения код возврата утилиты будет равен значению 0. В противном случае код завершения будет отличен от 0 и сообщения об ошибке будет помещено в стандартный поток ошибок (**stderr**).

Результат работы утилиты представляет собой описание одного или нескольких объектов данных (например, перечень зарегистрированных в кластере серверов информационных баз) и представлено в виде таблицы:

```
<Имя параметра> : <Значение параметра>
```

Где каждый параметр выводится с новой строки и содержит пустую строку в качестве признака завершения описания объекта. При этом *<Имя параметра>* совпадает с именами ключей командной строки утилиты. Если параметр не может быть задан через командную строку (или является параметром, доступным только для чтения), то имя параметра образуется от имени свойства соответствующего COM-объекта по следующему правилу: все отдельные слова (или сокращения) в имени свойства записываются отдельными словами в нижнем регистре и разделяются символом "-". Так, свойство рабочего процесса `MemoryExcessTime`, будет преобразовано в имя `memory-excess-time`.

Команды создания элементов кластера (за исключением администраторов), в случае успешного завершения, выводят в поток идентификатор созданного элемента в описанном выше формате.

Строки, допускающие произвольные символы, выводятся в двойных кавычках, при этом двойные кавычки в самих строках дублируются.

Даты представляются в формате XML (<http://www.w3.org/TR/2012/REC-xmlschema11-2-20120405/#dateTime>).

Для получения более подробной информации о параметрах сервера администрирования (**ras**) или утилиты администрирования (**rac**), можно получить в командной строке, запустив соответствующий исполняемый файл с параметром `help`:

```
rac help
rac help
```

[Копировать в буфер обмена](#)

На диске ИТС также поставляется пакет Java-архивов, который позволяет взаимодействовать с сервером администрирования из программы на языке Java, без помощи консольной утилиты администрирования (<http://its.1c.ru/db/metod8dev#content:4985:hdoc>).

5.3.3.2. Запуск сервера администрирования

5.3.3.2.1. Для ОС Windows

В режиме приложения

Запуск сервера администрирования в режиме приложения выполняется с помощью командной строки следующего вида:

```
rac cluster --port=<port> <host[:port]>
```

[Копировать в буфер обмена](#)

В команде запуска могут использоваться следующие ключи:

```
cluster
```

Запуск сервера администрирования в режиме администрирования кластера серверов.

```
--port или -p
```

Указывает сетевой порт, по которому утилита администрирования будет взаимодействовать с сервером администрирования. Значение по умолчанию равно **1545**.

```
<host[:port]>
```

Указывается адрес агента сервиса того кластера серверов, для администрирования которого запускается сервер администрирования.

Если адрес агента кластера не задан явным образом, то по умолчанию используется адрес `localhost:1540`.

В режиме сервиса

Для запуска сервера администрирования в режиме сервиса, необходимо зарегистрировать сервер администрирование в качестве службы. Данная операция может быть выполнена с помощью утилиты `sc.exe`. Для выполнения регистрации необходимы права администратора.

В качестве примера рассмотрим командный файл, выполняющий регистрацию службы сервера.

Файл `register-ras.bat`:

```
@echo off
rem %1 - полный номер версии 1С:Предприятия
set SrvUserName=<имя пользователя>
set SrvUserPwd=<пароль пользователя>
set CtrlPort=1540
set AgentName=localhost
set RASPort=1545
set SrvName="1С:Enterprise 8.3 Remote Server"
set BinPath="%C:\Program Files\1cv8\%1\bin\ras.exe" cluster --service --port=%RASPort% %AgentName%:%CtrlPort%
set Description="Сервер администрирования 1С:Предприятия 8.3"
sc stop %SrvName%
```

[Копировать в буфер обмена](#)

```
sc delete %SvcName%
sc create %SvcName% binPath= %BinPath% start= auto obj= %SvcUserName% password= %SvcUserPwd% displayName= %Description%
```

Перед применением данного командного файла необходимо указать в нем данные реального пользователя (имя и пароль) от имени которого будет работать служба сервера администрирования (строки `set SvcUserName=` и `set SvcUserPwd=`). Данный командный файл выполняет регистрация сервера администрирования со следующими параметрами:

- Имя службы: **1C:Enterprise 8.3 Remote Server**;
- Отображаемое имя: **Сервер администрирования 1C:Предприятия 8.3**;
- Порт сервера администрирования: **1545**;
- Адрес кластера серверов «1C:Предприятия»: **localhost:1540**;
- Режим запуска службы: **Автоматический**.

Пример использования:

```
register-ras 8.3.3.100
```

[Копировать в буфер обмена](#)

5.3.3.2.2. Для ОС Linux

В режиме приложения

Запуск сервера администрирования в режиме приложения выполняется с помощью командной строки следующего вида:

```
./ras cluster --port=<port> <host[:port]>
```

[Копировать в буфер обмена](#)

В команде запуска могут использоваться следующие ключи:

```
cluster
```

Запуск сервера администрирования в режиме администрирования кластера серверов.

```
--port или -p
```

Указывает сетевой порт, по которому утилита администрирования будет взаимодействовать с сервером администрирования. Значение по умолчанию равно **1545**.

```
<host[:port]>
```

Указывается адрес агента сервиса того кластера серверов, для администрирования которого запускается сервер администрирования.

Если адрес агента кластера не задан явным образом, то по умолчанию используется адрес **localhost:1540**.

В режиме демона

Для запуска сервера администрирования (**ras**) в режиме демона, необходимо запустить сервер администрирования с использованием специального ключа командной строки:

```
./ras cluster --daemon --port=<port> <host[:port]>
```

[Копировать в буфер обмена](#)

Ключи командной строки запуска сервера администрирования (**ras**) в ОС Windows и ОС Linux идентичны.

5.4. Администрирование СУБД

При эксплуатации систем на базе «1С:Предприятие» имеется возможность выполнять следующие действия, которые не оказывают влияния на работоспособность прикладного решения и не затрагивают представления структуры данных, используемые при работе «1С:Предприятия» (включая возможность выполнения реструктуризации информационной базы и сохранение сделанных изменений после завершения реструктуризации):

1. Использование шифрования на уровне СУБД.

- доступно для СУБД:
 - Microsoft SQL Server 2008 и старше;
 - Oracle Database.
- документация по механизму:
 - Microsoft SQL Server 2008: [http://msdn.microsoft.com/ru-ru/library/bb510663\(v=sql.100\).aspx](http://msdn.microsoft.com/ru-ru/library/bb510663(v=sql.100).aspx);
 - Microsoft SQL Server 2008 R2: [http://msdn.microsoft.com/ru-ru/library/bb510663\(v=sql.105\).aspx](http://msdn.microsoft.com/ru-ru/library/bb510663(v=sql.105).aspx);
 - Microsoft SQL Server 2012: [http://msdn.microsoft.com/ru-ru/library/bb510663\(v=sql.110\).aspx](http://msdn.microsoft.com/ru-ru/library/bb510663(v=sql.110).aspx);
 - Microsoft SQL Server 2014: [http://msdn.microsoft.com/ru-ru/library/bb510663\(v=sql.120\).aspx](http://msdn.microsoft.com/ru-ru/library/bb510663(v=sql.120).aspx);
 - Oracle Database 10g R2: http://docs.oracle.com/cd/B19306_01/network.102/b14268/asotrans.htm (на английском языке);
 - Oracle Database 11g R1: http://docs.oracle.com/cd/B28359_01/network.111/b28530/asotrans.htm (на английском языке);
 - Oracle Database 11g R2: http://docs.oracle.com/cd/E25178_01/network.1111/e10746/asotrans.htm (на английском языке);

2. Возможность использования кластера серверов – группы [компьютеров](#), объединённых высокоскоростными каналами связи и представляющая с точки зрения пользователя единый аппаратный ресурс.

- доступно для СУБД:
 - IBM DB2;
 - Microsoft SQL Server;
 - Oracle Database;

- PostgreSQL.
- документация по механизму:
 - IBM DB2 v9.1:
<http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/c0006354.htm>;
 - IBM DB2 v9.5:
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.admin.ha.doc/doc/t0051382.html> (на английском языке);
 - IBM DB2 v9.7:
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.ha.doc/doc/t0051382.html> (на английском языке);
 - IBM DB2 v10.1:
<http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.admin.ha.doc/doc/t0051382.html> (на английском языке);
 - Microsoft SQL Server 2000:
[http://msdn.microsoft.com/ru-ru/library/aa196694\(v=sql.80\).aspx](http://msdn.microsoft.com/ru-ru/library/aa196694(v=sql.80).aspx);
 - Microsoft SQL Server 2005:
[http://msdn.microsoft.com/ru-ru/library/ms189134\(v=sql.90\).aspx](http://msdn.microsoft.com/ru-ru/library/ms189134(v=sql.90).aspx);
 - Microsoft SQL Server 2008:
[http://msdn.microsoft.com/ru-ru/library/ms189134\(v=sql.100\).aspx](http://msdn.microsoft.com/ru-ru/library/ms189134(v=sql.100).aspx);
 - Microsoft SQL Server 2008 R2:
[http://msdn.microsoft.com/ru-ru/library/ms189134\(v=sql.105\).aspx](http://msdn.microsoft.com/ru-ru/library/ms189134(v=sql.105).aspx);
 - Microsoft SQL Server 2012:
[http://msdn.microsoft.com/ru-ru/library/ms189134\(v=sql.110\).aspx](http://msdn.microsoft.com/ru-ru/library/ms189134(v=sql.110).aspx);
 - Microsoft SQL Server 2014:
[http://msdn.microsoft.com/ru-ru/library/ms189134\(v=sql.120\).aspx](http://msdn.microsoft.com/ru-ru/library/ms189134(v=sql.120).aspx);
 - Oracle Database 10g R2:
http://docs.oracle.com/cd/B19306_01/server.102/b14210/hafeatures.htm#CJACEEIA (на английском языке);
 - Oracle Database 11g R1:
http://docs.oracle.com/cd/B28359_01/server.111/b28281/architectures.htm (на английском языке);
 - Oracle Database 11g R2:
http://docs.oracle.com/cd/E15586_01/server.1111/e17157/architectures.htm (на английском языке);
 - PostgreSQL 8.2:
<http://www.postgresql.org/docs/8.2/interactive/high-availability.html> (на английском языке);
 - PostgreSQL 8.3:
<http://www.postgresql.org/docs/8.3/interactive/high-availability.html> (на английском языке);
 - PostgreSQL 8.4:
<http://www.postgresql.org/docs/8.4/interactive/high-availability.html> (на английском языке);
 - PostgreSQL 9.0:
<http://www.postgresql.org/docs/9.0/interactive/high-availability.html> (на английском языке);
 - PostgreSQL 9.1:
<http://www.postgresql.org/docs/9.1/interactive/high-availability.html> (на английском языке);
 - PostgreSQL 9.2:
<http://www.postgresql.org/docs/9.2/interactive/high-availability.html> (на английском языке).
- 3. Использование сжатия данных на уровне СУБД.
 - доступно для СУБД:
 - Oracle Database (только через сжатое табличное пространство).
 - документация по механизму:
 - Oracle Database 11g R2:
http://docs.oracle.com/cd/E11882_01/server.112/e25494/tables.htm#CJAGFBFG (на английском языке).
- 4. Возможность изменения положения predeterminedных табличных пространств.
 - доступно для СУБД:
 - IBM DB2;
 - Oracle Database;
 - PostgreSQL.
 - predeterminedные табличные пространства:
 - IBM DB2:
 - для индексов – `V81C_INDEXSPACE`;
 - для данных – `V81C_LARGESPACE`;
 - для LOB – `V81C_LOBSPACE`;
 - пользовательское временное табличное пространство – `V81C_USERTEMP`;
 - системное временное табличное пространство – `V81C_SYSTEMPBP`.
 - Oracle Database:
 - для индексов – `V81C_INDEX`;
 - для данных – `V81C_DATA`;
 - для LOB – `V81C_LOB`;

- временное табличное пространство – [V81C_TEMP](#).

- PostgreSQL:

- для индексов – [V81C_INDEX](#);
- для данных – [V81C_DATA](#);
- временное табличное пространство – произвольное наименование, которое задается через конфигурационный файл [postgresql.conf](#).

5. Возможность изменения положения файла с базой данных.

- доступно для СУБД:

- Microsoft SQL Server.

6. Изменение местоположения журнала транзакций.

- доступно для СУБД:

- IBM DB2;
- Microsoft SQL Server;
- Oracle Database;
- PostgreSQL.

7. Выполнение административных задач, не изменяющих структур базы данных, но служащих для обеспечения работоспособности адекватной производительности.

- задачи:

- проверка целостности;
- реиндексация;
- дефрагментация;
- реорганизация;
- создание резервной копии (см. [здесь](#));
- очистка процедурного кеша;
- сбор статистики.

- доступно для СУБД:

- IBM DB2;
- Microsoft SQL Server;
- Oracle Database;
- PostgreSQL.

Глава 6. Удаление системы

6.1. Удаление кластера серверов под Windows

Удаление сервера «1С:Предприятие» выполняется специальной программой, которая удаляет компоненты системы с жесткого диска компьютера, производит изменения в меню [Пуск](#) и системной информации Microsoft Windows.

Перед удалением необходимо завершить работу сервера «1С:Предприятия».

Для удаления системы «1С:Предприятие» необходимо выполнить следующие действия:

1. Запустить панель управления операционной системы Microsoft Windows и щелкнуть мышью пиктограмму [Установка и удаление программ](#) ([Программы и компоненты](#) для Windows Vista и выше).
2. Если необходимо, в выданном на экран диалоге щелкнуть мышью на пиктограмме [Замена или удаление](#).
3. В списке установленных программ выбрать строку [1С Предприятие 8.3 \(8.3.X.YYY\)](#) и нажать кнопку [Удалить](#).

Будет задан вопрос о необходимости выполнения удаления. В случае утвердительного ответа будет запущено удаление выбранной версии с компьютера и выполнение необходимых изменений в системной информации.

6.2. Удаление кластера серверов под Linux

Для удаления сервера «1С:Предприятия» с компьютера следует воспользоваться программой менеджера пакетов используемой операционной системы. В данном разделе приведен пример использования менеджера пакетов [rpm](#).

ВНИМАНИЕ! Как и установка, удаление должно выполняться от лица пользователя [root](#).

Перед удалением необходимо завершить работу кластера серверов, для чего следует выполнить команду:

```
/etc/init.d/srv1cv83 stop
```

[Копировать в буфер обмена](#)

Для удаления следует выполнить команду:

```
rpm -e <Имя_пакета_rpm>
```

[Копировать в буфер обмена](#)

[<Имя_пакета_rpm>](#) в данном случае – это имя установленного rpm-пакета «1С:Предприятия». Имя установленного rpm-пакета соответствует названию файла пакета, без суффикса «.i386.rpm», например, [1C_Enterprise83-common-8.3.3-100](#). Таким образом, для удаления пакета [1C_Enterprise83-common-8.3.3-100.i386.rpm](#) следует ввести команду:

```
rpm -e 1C_Enterprise83-common-8.3.3-100
```

[Копировать в буфер обмена](#)

Удаление rpm-пакетов должно выполняться в порядке, обратном установке, чтобы зависимый пакет удалялся до того пакета, от которого он зависит. В применении к «1С:Предприятию» это означает, что rpm-пакет [1C_Enterprise83-common](#) должен быть удален в последнюю очередь.

Также можно выполнить удаление rpm-пакетов командой:

```
rpm -e `rpm -qa | grep 1C_`
```

[Копировать в буфер обмена](#)

Данная команда удалит все установленные пакеты, которые начинаются с префикса «1C_». Зависимости будут отслежены автоматически.

Для других пакетных менеджеров существуют аналогичные команды.

Приложение 1. Формат строки описания лицензии

Информация о лицензии может отображаться в полном или кратком виде. Информация о лицензии в кратком виде используется для отображения в колонке списка рабочих процессов или сеансов. Информация о лицензии в полном виде используется в окне свойств рабочего процесса или сеанса.

Информация о лицензии имеет следующий формат:

Кем получено, PID, Сервер, Порт, Описание лицензии, Имя файла с лицензией [Копировать в буфер обмена](#)

В зависимости от типа лицензии (HASP или программная), а также краткой или полной информации о лицензии – будет присутствовать тот или иной набор полей.

Кем получено

Сообщает о том, кем получена лицензия:

- **Клиент** – лицензия получена клиентским приложением.
- **Сервер** – лицензия получена сервером «1С:Предприятия».

PID

Идентификатор процесса ОС, который получил клиентскую лицензию.

Не используется в кратком представлении лицензии.

Сервер

Имя сервера «1С:Предприятия», который получил клиентскую лицензию.

Не используется в кратком представлении лицензии.

Не используется в случае, если лицензия получена клиентским приложением.

Порт

Номер сетевого порта процесса сервера (**rphost** или **rmngr**), который получил лицензию.

Не используется в кратком представлении лицензии.

Не используется в случае, если лицензия получена клиентским приложением.

Описание лицензии

Содержит описание источника лицензии.

Если лицензия получена из аппаратного ключа защиты, то описание лицензии имеет следующий вид:

<Серия ключа> <Тип ключа> <Количество лицензий> [Копировать в буфер обмена](#)

Где:

- **Серия ключа** – серия ключа аппаратной защиты, подробное описание которых приведено в книге «1С:Предприятие 8.3. Руководство администратора».
- **Тип ключа** – указывает тип аппаратного ключа:
 - **Локальный** – ключ подключен к тому компьютеру, который выполнил получение лицензии.
 - **Сетевой** – ключ доступен по сети через HASP License Manager.
- **Количество лицензий** – максимальное количество лицензий, доступных в данном ключе.

Если клиентская лицензия предоставляется с помощью системы программного лицензирования, то описание лицензии имеет следующий вид:

<Номер комплекта> <Тип лицензии> <Количество лицензий> [Копировать в буфер обмена](#)

Где:

- **Номер комплекта** – регистрационный номер комплекта с программными лицензиями.
- **Тип лицензии** – указывает активированный тип лицензии. Для комбинированных комплектов может принимать значение **1** или значение, равное полю **Количество лицензий**. Для однопользовательских, **многопользовательских** или серверных лицензий значение всегда равно **1**.
- **Количество лицензий** – максимальное количество лицензий, доступных в данном комплекте лицензий.

Имя файла с лицензией

Полное имя файла с программной лицензией, «чья» лицензия была получена рабочим процессом или сеансом.

Не используется в кратком представлении лицензии.

Не используется для лицензии, полученной из аппаратного ключа защиты.

Рассмотрим примеры строк представления клиентских лицензий:

Клиент, 4648, ORGL8 Локальный 1 [Копировать в буфер обмена](#)

Лицензия получена клиентским приложением, имеющим идентификатор процесса ОС равный **4648**. Лицензия получена из локального ключа аппаратной защиты.

Клиент, 3840, ORG8A Сетевой 300 [Копировать в буфер обмена](#)

Лицензия получена клиентским приложением, имеющим идентификатор процесса ОС равный **3840**. Лицензия получена из ключа аппаратной защиты, доступного по сети.

Сервер, 1648, SERVER-1C, 1560, ORG8B Сетевой 500 [Копировать в буфер обмена](#)

Клиентская лицензия получена через сервер **SERVER-1C**, работающего через порт **1560**. Процесс, получивший лицензию, имеет идентификатор процесса ОС равный **1648**, а лицензия получена из аппаратного ключа защиты на **500** лицензий, доступного по сети.

[Копировать в буфер обмена](#)

Сервер, 1648, SERVER-1C, 1560, ORGL8 Локальный 1

Клиентская лицензия получена через сервер **SERVER-1C**, работающего через порт **1560**. Процесс, получивший лицензию, имеет идентификатор процесса ОС равный **1648**, а лицензия получена из локального, однопользовательского, аппаратного ключа защиты.

[Копировать в буфер обмена](#)

Сервер, 1970, ENSR8 Локальный 1

Серверная лицензия (для 32-разрядного сервера) получена рабочим процессом, имеющим идентификатор процесса ОС равный **1970**. Ключ расположен на том же компьютере, на котором работает рабочий процесс.

[Копировать в буфер обмена](#)

Сервер, 4524, SERVER-1C, 1560, 8000314159 1 1, <file:///C:/Users/USR1CV8/AppData/Local/1C/1Cv8/conf/20100521112156.lic>

Серверная лицензия получена рабочим процессом сервера **SERVER-1C**, работающего через порт **1560**. Процесс, получивший лицензию, имеет идентификатор процесса ОС равный **4524**. Получена программная серверная лицензия, имеющая номер комплекта **8000314159**. Файл с лицензией имеет имя **20100521112156.lic** и расположен в каталоге **C:/Users/USR1CV8/AppData/Local/1C/1Cv8/conf**.

[Копировать в буфер обмена](#)

Сервер, 896, SERVER-1C, 1564, 8000453822 20 20, <file:///C:/Documents and Settings/USR1CV8/Local Settings/Application Data/1C/1Cv8/conf/20110812100013.lic>

Клиентская лицензия получена через сервер **SERVER-1C**, работающего через порт **1560**. Процесс, получивший лицензию, имеет идентификатор процесса ОС равный **4524**. Получена программная лицензия из комбинированного комплекта на 20 лицензий, который активирован как многопользовательская лицензия. Номер комплекта **8000453822**. Файл с лицензией имеет имя **20110812100013.lic** и расположен в каталоге **C:/Documents and Settings/USR1CV8/Local Settings/Application Data/1C/1Cv8/conf**.

1С:Предприятие 8.3.6

Руководство пользователя Интерфейс «Такси»

83.000.03 от 05-02-2015

Руководство пользователя. Интерфейс «Такси» (02)

Введение (02)

Глава 1. Интерфейс «Такси» (11)

Глава 2. Работа в формах (22)

Глава 3. Списки (19)

Глава 4. Работа с данными различных видов (08)

Глава 5. Отчеты (17)

Глава 6. Изменение варианта отчета (20)

Глава 7. Сервисные возможности (16)

Глава 8. Настройка программы (16)

Глава 9. Получение справочной информации (07)

Глава 10. Формат поисковых выражений (02)

Руководство пользователя. Интерфейс «Такси»

1С:ПРЕДПРИЯТИЕ 8.3

Руководство

пользователя.

Интерфейс «Такси»

Москва

Фирма «1С»

2015

ПРАВО ТИРАЖИРОВАНИЯ
ПРОГРАММНЫХ СРЕДСТВ И ДОКУМЕНТАЦИИ ПРИНАДЛЕЖИТ ФИРМЕ «1С»

Приобретая систему «1С:Предприятие»,
вы тем самым даете согласие не допускать
копирования программ и документации
без письменного разрешения фирмы «1С»

© ООО «1С», 2014

Фирма «1С», Москва, 123056, а/я 64.

Отдел продаж: Селезневская ул., 21,

телефон: +7 (495) 737-92-57,

факс: +7 (495) 681-44-07.

email: 1c@1c.ru

URL: www.1c.ru, www.v8.1c.ru

Группа разработки программ – А. Алексеев, А. Безбородов, Д. Бескоровайнов, П. Василец, А. Виноградов, А. Волков, И. Гольштейн, Е. Горностаев, Г. Дамье, А. Даровских, О. Дерут, Н. Евграфов, Б. Евтифеев, Д. Зарецкий, Д. Ивашов, С. Копиенко, И. Коваленко, Н. Корсаков, С. Кравченко, В. Кудрявцев, П. Кукушкин, А. Кунченко, А. Лакутин, М. Лейбович, Г. Леонтьев, А. Лехан, А. Макеев, А. Медведев, Е. Митрошкин, С. Мурзин, С. Нуралиев, М. Отставнов, Д. Павленко, А. Плякин, А. Рукин, Д. Русанов, М. Саблин, Е. Силин, Д. Службин, А. Смирнов, В. Соколов, П. Солодкий, В. Сосновский, Д. Сысоенков, А. Топорков, В. Тунегов, А. Трубкин, В. Филиппов, А. Цилябин, В. Черемисинов, П. Чиков, А. Чичерин, А. Шевченко, А. Щербинин.

Документация – В. Байдаков, В. Дранищев, Е. Королькова, А. Краюшкин, И. Кузнецов, М. Лавров, А. Моничев, А. Плякин, М. Радченко.

Группа тестирования – Е. Аникеева, К. Афанасьев, Я. Баджиева, Н. Бондарчук, Т. Булыгина, К. Доровских, С. Егоров, А. Егорова, С. Карасев, А. Коробова, Е. Литвиненко, И. Макарова, М. Панкратов, О. Ридер, Е. Ситосенко, Е. Смирнова, М. Снопко, А. Соколов, С. Харькова, Н. Шаргунова, Д. Шведова.

Группа консультационной поддержки – С. Алексеева, О. Багрова, О. Баклушина, М. Белоконь, Э. Гарифуллина, В. Давыдова, О. Дмитренко, Л. Ермакова, М. Ершова, Ю. Жестков, С. Жолудев, М. Жолудева, О. Завальская, Н. Заявлиная, О. Колоскова, Г. Коробка, Ю. Лаврова, С. Лепешкина, С. Мазурин, С. Марков, Ю. Мисан, А. Павликов, И. Панин, О. Пехтерева, С. Постнова, А. Прокуровский, Г. Степаненко, Н. Степанов, Т. Токарева, Г. Ястребова.

Корректурa – Е. Семененко.

Наименование книги:	1С:Предприятие 8.3. Руководство пользователя
Номер издания:	83.010.03
Дата выхода:	5 февраля 2015 г.

ЛИНИЯ КОНСУЛЬТАЦИЙ

Для пользователей системы «1С:Предприятие 8» услуги линии консультаций предоставляются по линии информационно-технологического сопровождения (ИТС).

Получение консультаций возможно только после регистрации программного продукта (для чего необходимо заполнить регистрационную анкету и выслать ее в фирму «1С») и оформления подписки на ИТС.

В стоимость комплекта поставки программного продукта включено обслуживание по линии ИТС в течение периода, указанного в вашей регистрационной анкете. Купон на бесплатное обслуживание по линии ИТС, а также конверт с маркой и адресом входят в состав приобретенного вами комплекта программ.

По истечении периода бесплатного обслуживания получение консультаций возможно только по договору на ИТС. Для заключения договора обращайтесь к партнерам фирмы «1С».

Телефон и адрес электронной почты для консультаций указаны на диске ИТС.

Линия консультаций работает с 9:30 до 17:30 по московскому времени, кроме суббот, воскресений и праздничных дней (по распорядку московских предприятий).

Обращаясь к линии консультаций, следует находиться рядом со своим компьютером, иметь под рукой настоящее руководство и свою половину регистрационной анкеты. Желательно заранее уточнить типы используемого компьютера и принтера.

Набрав телефон линии консультаций, вы должны услышать ответ дежурного сотрудника. После этого сообщите наименование вашей организации, номер вашего экземпляра системы (он обозначен на CD-ROM комплекта поставки и на вашей половине регистрационной анкеты) и, возможно, другую регистрационную информацию по запросу сотрудника линии консультаций. Названная вами информация будет проверена по данным, указанным в отосланной в фирму «1С» половине регистрационной анкеты.

Отвечая на возникшие у вас вопросы, дежурный сотрудник может воспроизвести возникшую ситуацию на своем компьютере. Он даст вам рекомендации сразу же или после обсуждения с разработчиками. Вам не нужно просить к телефону конкретных специалистов: мы отвечаем за работу всего персонала. Работа линии консультаций регистрируется, поэтому при повторных обращениях по сложным проблемам вы можете сослаться на дату и время предыдущего разговора.

МЫ ВСЕГДА РАДЫ ВАМ ПОМОЧЬ!

Введение

Настоящая книга является Руководством по использованию системы «1С:Предприятие» в интерфейсе [Такси](#) (далее – Руководство).

Данное Руководство предназначено для пользователей «1С:Предприятия».

Так как работа системы во многом определяется используемой конфигурацией, для ознакомления с особенностями работы конкретных документов, отчетов и других элементов конфигурации необходимо изучить описание конфигурации, которая используется при работе с «1С:Предприятием».

Специалистам, занимающимся конфигурированием и администрированием системы, необходимо обращаться к книгам «1С:Предприятие 8.3. Руководство разработчика» и «1С:Предприятие 8.3. Руководство администратора».

Структура Руководства

В главе 1 описываются приемы навигации по программе в интерфейсе [Такси](#).

В главе 2 содержатся сведения о работе с формами.

В главе 3 приводится описание порядка работы со списками в системе «1С:Предприятие».

В главе 4 содержится описание типовых приемов работы с различными видами данных.

В главах 5 и 6 излагаются приемы работы с отчетами и вариантами отчетов.

В главе 7 описываются сервисные возможности системы: использование ссылок и избранного, работа с историей, использование встроенного калькулятора и т. д.

В главе 8 рассматриваются возможности пользовательской настройки интерфейса, форм и параметров системы.

В главе 9 содержится информация по работе со справочной системой.

В главе 10 описывается формат поисковых выражений, используемый при поиске в программе.

Сведения о системе, об установке и обновлении системы, запуске программы и работе со списком информационных баз, а также информация по работе со встроенными редакторами приведены в книге «1С:Предприятие 8.3. Руководство пользователя».

Примечания

В тексте Руководства встречаются примечания нескольких видов:

- Совет – содержит альтернативный метод действия;
- Примечание – содержит дополнительную информацию;
- Внимание! – содержит информацию, необходимую для правильной работы системы.

Обозначения

Обозначения клавиш. Клавиши, например [Enter](#), [Esc](#), [Delete](#), выделяются в тексте, как показано выше.

Для ссылок на клавиши управления курсором (клавиши со стрелками) будет использоваться фраза [клавиши управления курсором](#). Если необходимо упомянуть эти клавиши по отдельности, будут использоваться словосочетания [Стрелка вверх](#), [Стрелка вниз](#), [Стрелка вправо](#) и [Стрелка влево](#).

Комбинации клавиш. Когда для выполнения какого-либо действия необходимо нажать комбинацию клавиш, в тексте это обозначается, например, как [Ctrl + F3](#).

Обозначения кнопок. Наименования кнопок в диалогах и формах, закладках, ввода-редактирования данных будут даваться их названиями без кавычек, например **ОК**, **Отмена**, **Удалить** и так далее.

Описание действий. В Руководстве описание действия (открытие журнала документов, ввод документа, построение отчета и т. д.) производится выбором пункта меню (основного или активного окна, а также контекстного меню). Аналогичные действия в большинстве случаев можно выполнить с помощью команд командной панели. При выборе пункта меню следует обращать внимание на пиктограмму, располагающуюся слева от наименования пункта. Эта пиктограмма также размещена на кнопке командной панели с аналогичным действием. Изображения пиктограмм в тексте Руководства обычно не приводятся. Помимо пиктограммы следует обращать внимание на сочетание клавиш, с помощью которых также выбирается данное действие. Сочетание указывается справа от наименования пункта меню.

Выбор элемента обычно осуществляется двойным или одинарным нажатием левой кнопки мыши.

Что вы должны знать

Характер изложения данного Руководства предполагает, что вы знакомы с операционной системой компьютера, на котором работает система «1С:Предприятие» (ОС семейства Microsoft Windows и ОС семейства Linux), и владеете базовыми навыками работы в ней.

Вам должны быть знакомы следующие понятия и навыки (список приведен для операционной системы Windows):

- использование меню **Пуск (Start)** для вызова программ,
- приемы работы с окнами,
- работа с меню,
- использование управляющих элементов диалогов,
- стандартные диалоги,
- понятие буфера обмена операционной системы и приемы работы с ним,
- настройка операционной системы с помощью панели управления (**Control Panel**).

Если вы недостаточно хорошо владеете перечисленными выше понятиями и навыками, рекомендуем обратиться к документации по операционной системе.

При описании мобильного приложения предполагается, что вы знакомы с операционной системой мобильного устройства, на котором работает система «1С:Предприятие» (iOS, Android, далее – операционная система или ОС), и владеете базовыми навыками работы в ней.

Вам должны быть знакомы следующие понятия и навыки:

- установка и обновление приложения с помощью магазина приложений ОС,
- удаление приложения с мобильного устройства,
- приемы работы с пользовательским интерфейсом ОС,
- способы резервного копирования данных пользователя.

Глава 1. Интерфейс «Такси»

В платформе «1С:Предприятие» реализованы три вида (режима) интерфейса:

- [Такси](#),
- [Формы в закладках](#),
- [Формы в отдельных окнах](#).

В данной книге содержатся сведения об интерфейсе [Такси](#) и описываются приемы навигации по программе.

Описание интерфейсов [Формы в закладках](#) и [Формы в отдельных окнах](#) см. в книге «1С:Предприятие 8.3. Руководство пользователя».

В «1С:Предприятии» пользователь работает с системой окон. Существует два вида окон: **основное** и **вспомогательные**.

Основное окно приложения предназначено для навигации по приложению, вызова различных команд, для работы с конкретными объектами информационной базы (например, с документами или элементами списков). Во вспомогательных окнах открываются системные функции, такие как календарь, калькулятор и т. д.

Чтобы открыть ту или иную форму, следует найти в интерфейсе нужную команду (открытие списка, ввода документа и т. д.) и выполнить ее (выбрать с помощью мыши или клавиши [Enter](#)).

Вид интерфейса может быть установлен разработчиком или же выбран пользователем в системных параметрах, если его использование предусмотрено разработчиком. Подробнее о переключении вариантов интерфейса см. раздел «[Системные параметры](#)», [здесь](#).

При запуске системы открывается основное окно программы. В нем пользователю видна вся структура прикладного решения. Основное разделение функциональности представляется в виде панели разделов.

В общем случае основное окно приложения имеет следующий вид:

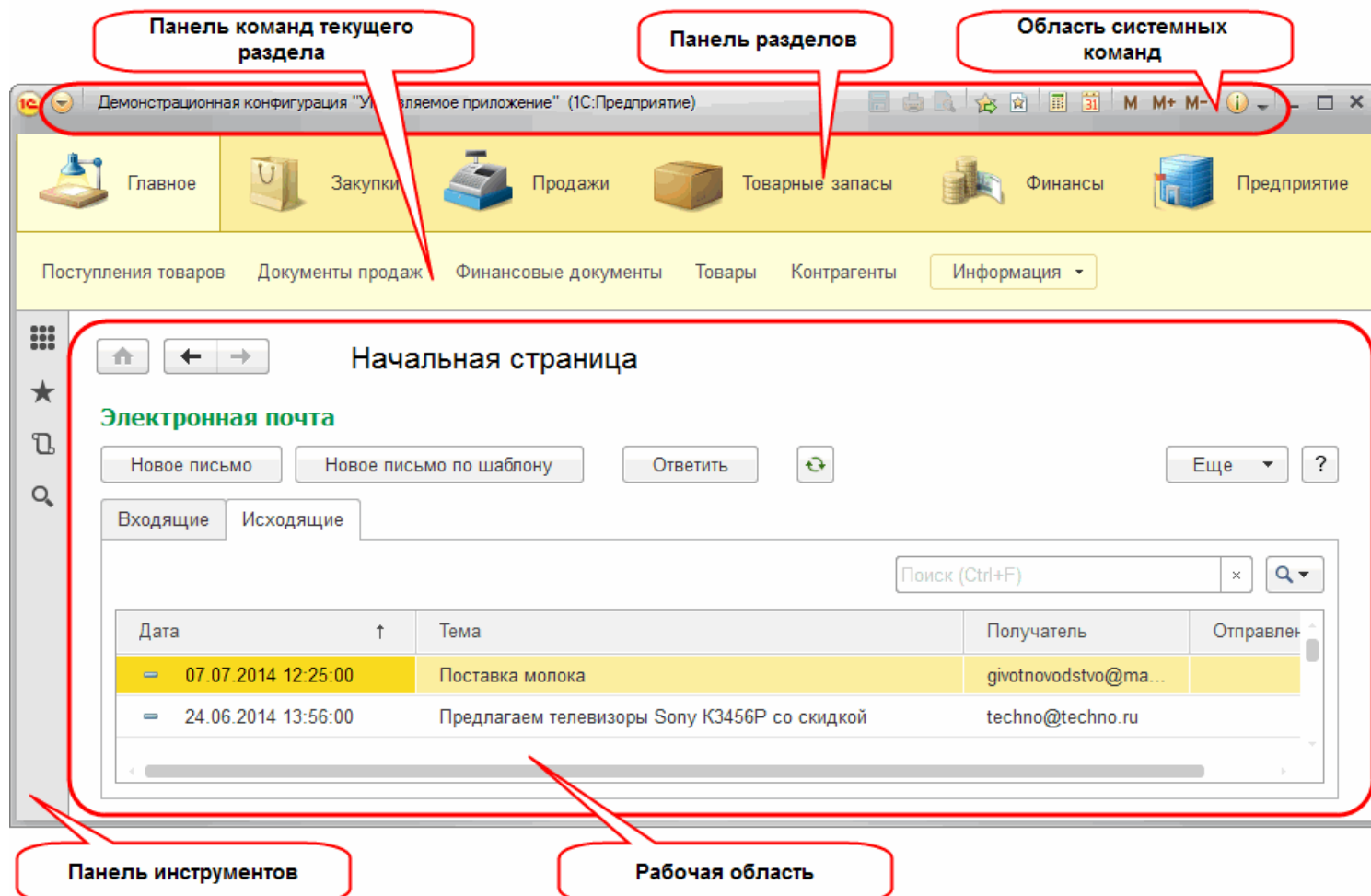


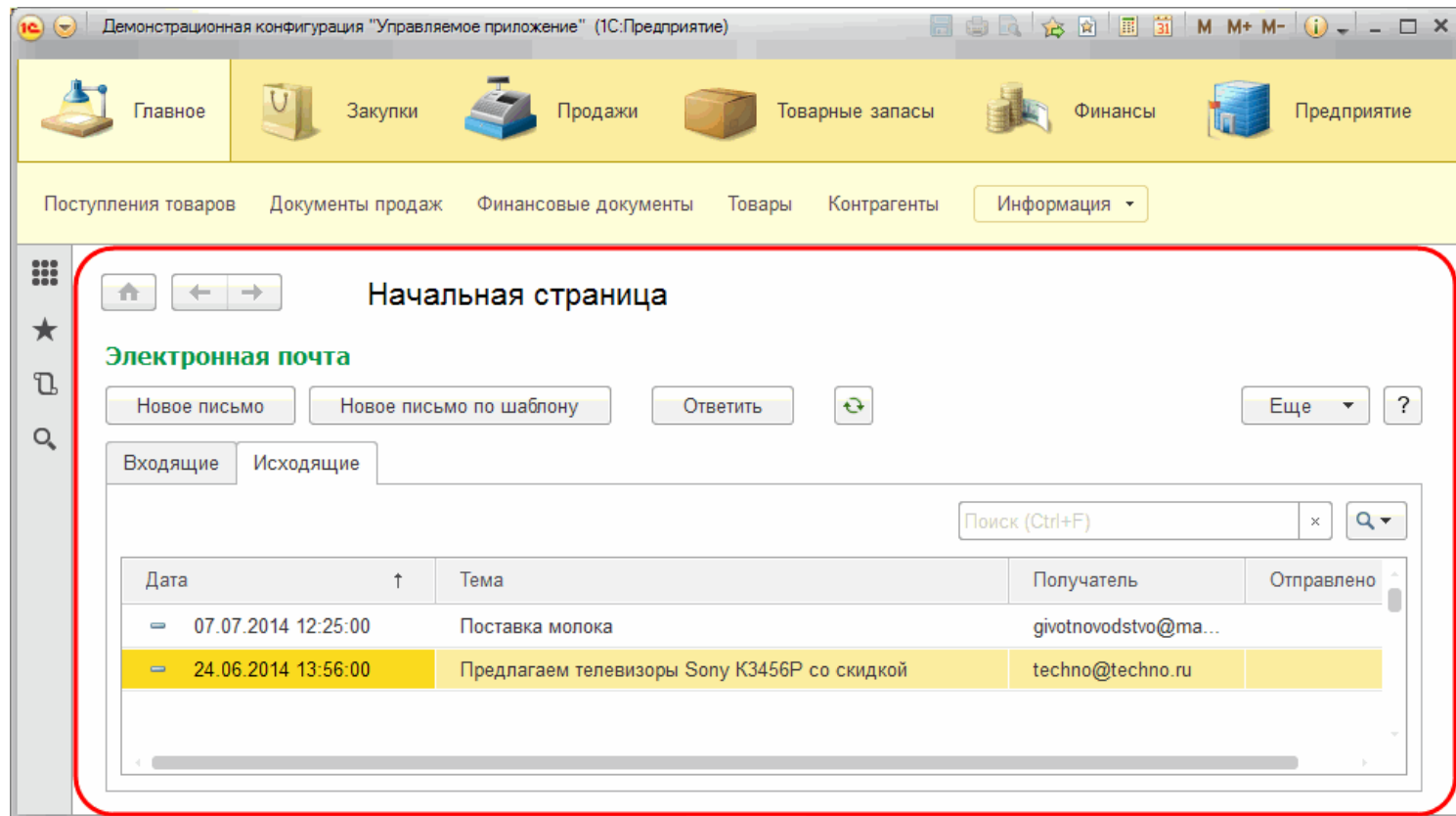
Рис. 1. Основное окно программы

Окно устроено таким образом, что позволяет достаточно легко ориентироваться в программе, перемещаться между различными областями, переключаться между открытыми формами и окнами.

Существует возможность изменять размеры основного окна с помощью мыши.

1.1. Начальная страница

При запуске программы отображается начальная страница (если предусмотрены формы для нее).

**Рис. 2. Начальная страница**

Существует возможность пользовательской настройки начальной страницы (если предусмотрена на стадии разработки). Для этого в главном меню следует выбрать **Вид – Настройка начальной страницы**. Информацию по настройке начальной страницы см. [здесь](#).

Для переключения между формами начальной страницы используется клавиша **F6**.

Чтобы вернуться на начальную страницу из любой другой открытой формы, следует нажать кнопку в виде домика в заголовке формы.

**Рис. 3. Кнопка перехода на стартовую страницу в заголовке формы**

Если в программе нет содержимого для начальной страницы, то кнопка перехода на начальную страницу будет недоступна.

1.2. Панель разделов

Панель разделов содержит список всех разделов, из которых состоит программа.

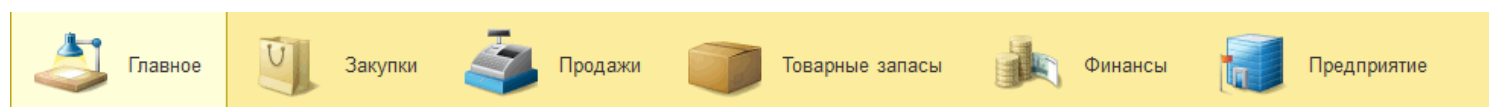


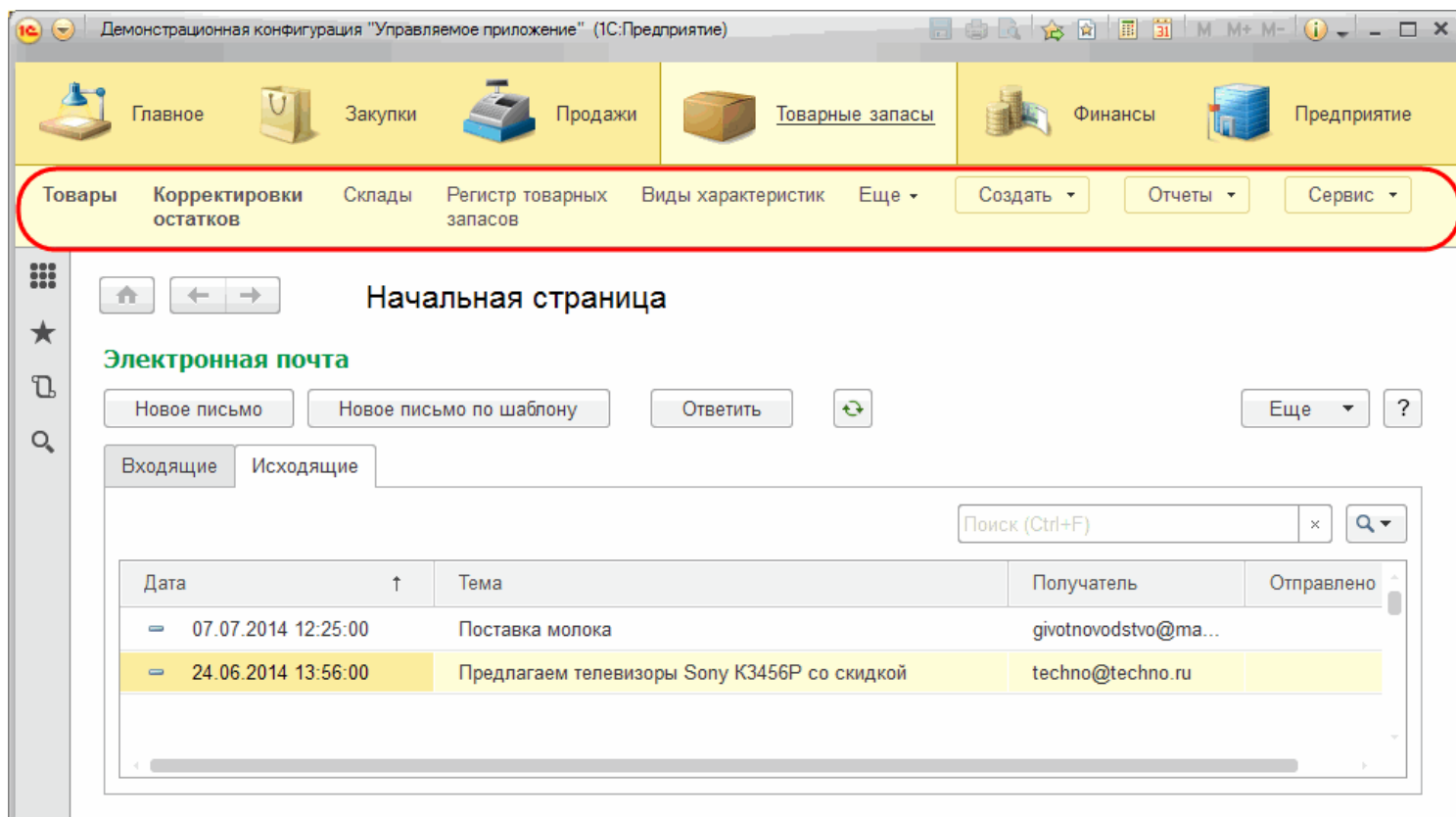
Рис. 4. Пример панели разделов

Если разделы не умещаются в списке по ширине окна, то появится кнопка прокрутки:

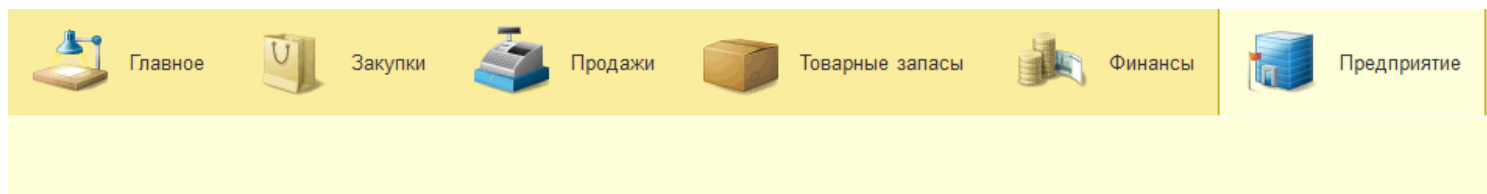
**Рис. 5. Пример кнопки прокрутки на панели разделов**

Существует возможность настраивать панель разделов. Диалог настройки панели разделов открывается командой главного меню **Вид – Настройка панели разделов**. Настройка подробно описана в разделе «[Панель разделов](#)» [здесь](#).

Если панель команд текущего раздела (см [здесь](#)) отображается, то при выборе раздела в панели команд отображаются команды, доступные в текущем разделе.

**Рис. 6. Панель команд раздела «Товарные запасы»**

Панель команд текущего раздела может не отображать команды, если у пользователя недостаточно прав на их использование или они были отключены пользователем.

**Рис. 7. Панель команд раздела не содержит команды**

При повторном выборе раздела откроется «меню функций» – список всех команд выбранного раздела.

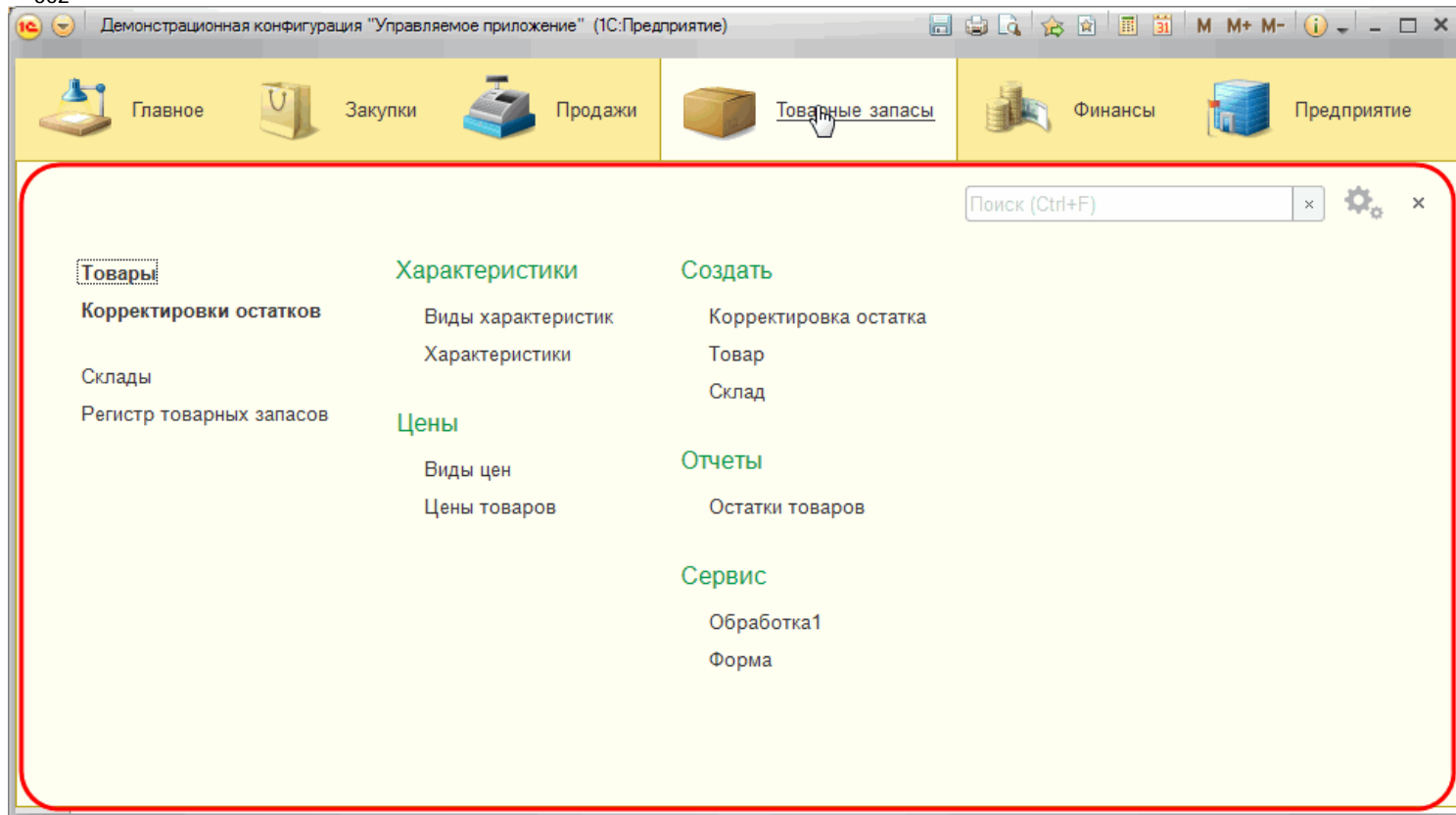


Рис. 8. Список всех команд выбранного раздела

Команды в меню функций делятся на команды навигации и команды действий (разделы [Создать](#), [Отчеты](#) и [Сервис](#)). Подробнее см. раздел «[Команды навигации](#)» [здесь](#) и раздел «[Команды действий](#)», [здесь](#).

При нажатии на гиперссылку выполнится соответствующая команда (открытие формы списка, создание нового объекта, запуск отчета, и т. д.).

Существует возможность поиска по списку команд. Чтобы найти команду, следует ввести ее название (или часть) в поле [Поиск](#). Поиск выполняется по всем разделам, а не только по командам текущего раздела. В меню функций отобразятся только команды, в названиях которых есть искомое выражение.

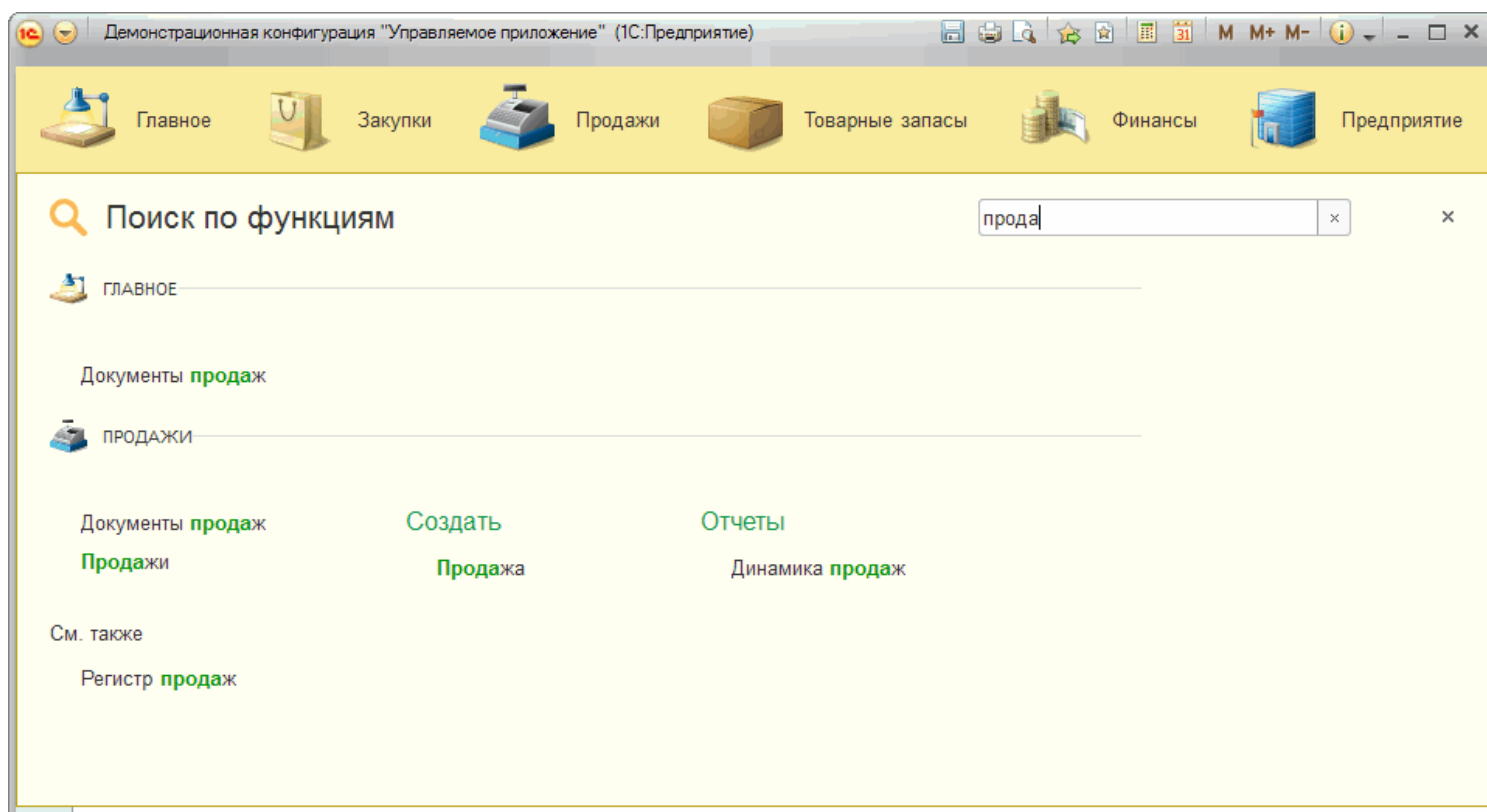


Рис. 9. Пример поиска по функциям

Чтобы отменить поиск, следует нажать клавишу **Esc**.

Если панель команд текущего раздела скрыта, то при выборе раздела откроется меню функций.

ПРИМЕЧАНИЕ. Также открыть меню функций можно нажатием клавиши **F10**.

Если панель разделов не отображается (скрыта настройками программы или пользователем в редакторе панелей), то над меню функций будет отображен список разделов, и при выборе раздела отобразятся команды этого раздела.

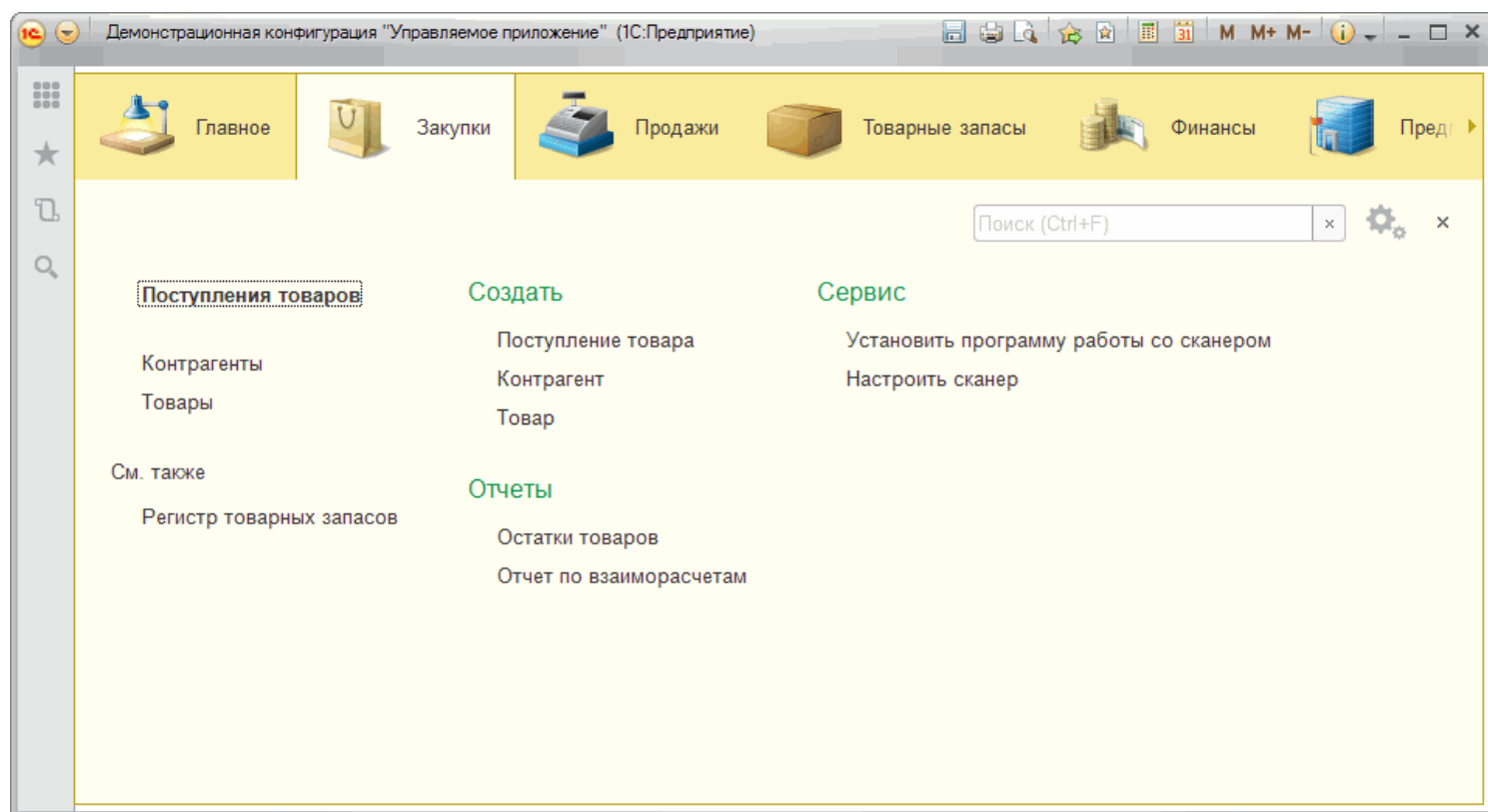


Рис. 10. Панель разделов в меню действий

При нажатии на иконке **Настройки** откроется подменю, содержащее команды **Настройка навигации** и **Настройка действий**, при вызове которых откроется соответствующий диалог настроек команд текущего раздела. Подробнее о настройке списка команд навигации см. [здесь](#). Подробнее о настройке списка команд действий раздела см. [здесь](#).

Существует возможность быстро добавлять команды из списка меню функций в **Избранное** для последующего оперативного вызова. Для этого следует нажать на звездочку, расположенную слева от команды.



Рис. 11. Добавление команды в Избранное

1.2.1. Команды навигации

Список команд навигации в меню функций отражает структуру текущего раздела. Если у раздела имеются подчиненные разделы, то они будут отображаться как отдельные группы.

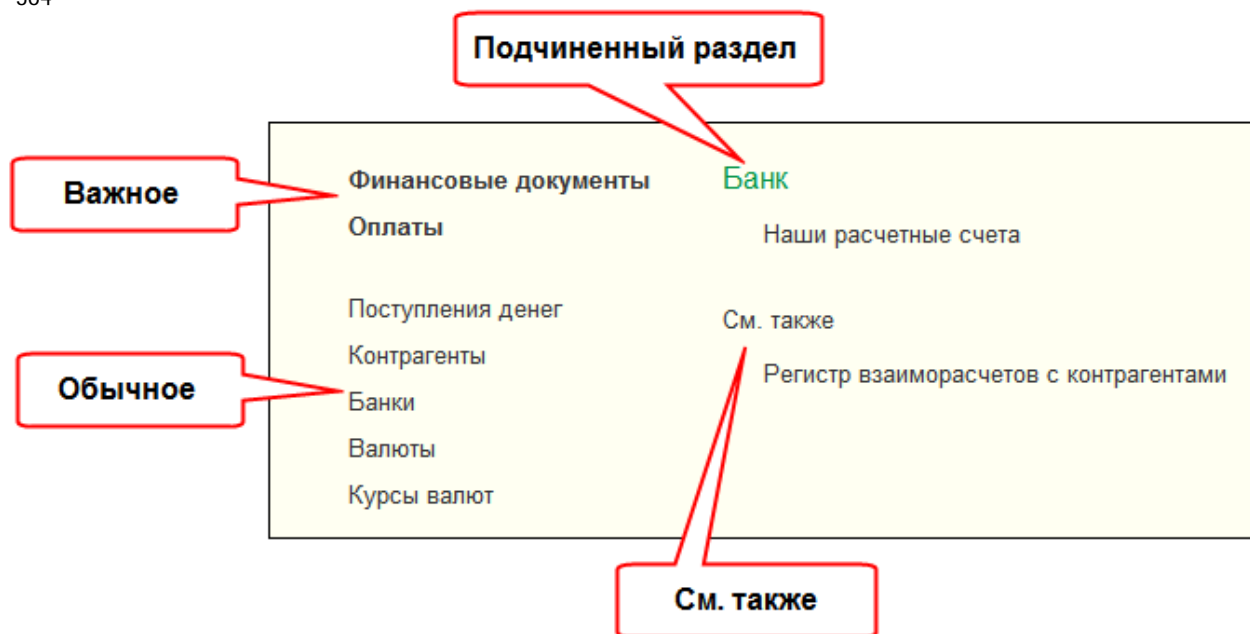


Рис. 12. Команды навигации

Команды навигации подразделяются на три группы:

- **Важное** – команды перехода к важным областям данных в контексте текущего раздела;
- **Обычное** – команды перехода к данным текущего раздела;
- **См. также** – ссылки на дополнительную информацию, которая может не относиться к данным текущего раздела напрямую, но может быть востребована в некоторых случаях.

Состав команд и принадлежность команды к группе устанавливаются разработчиком.

СОВЕТ. При работе с клавиатуры для перехода на панель навигации текущего раздела нужно нажать **Alt + 2**.

Существует возможность настройки списка команд навигации. Диалог настройки списка открывается командой **Настройка навигации** в меню функций раздела. О настройке списка команд навигации см. раздел [«Панель навигации»](#) [здесь](#).

Команды действий

На панели команд раздела отображаются списки команд-действий, которые доступны в текущем разделе.

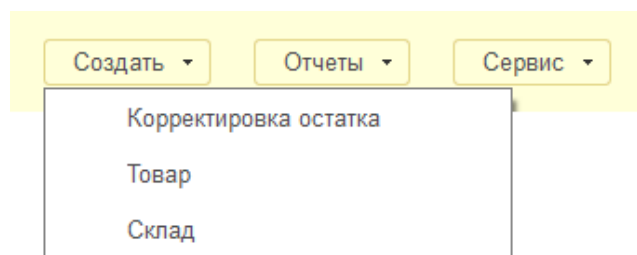


Рис. 13. Команды действий на панели команд текущего раздела

Эти команды объединены в группы: **Создать**, **Отчеты**, **Сервис**. Также могут существовать иные группы, созданные в конфигурации.

В группе **Создать** размещаются команды создания новых объектов информационной базы, например, документов или элементов списков.

Группа **Отчеты** содержит команды открытия форм отчетов.

В группе **Сервис** отображаются команды вызова служебных инструментов. Если в группе нет ни одной команды, то она не отображается. Высота и ширина панели регулируются.

При наведении курсора на название команды отображается подсказка, которая может содержать

дополнительную информацию о действии или ссылку на нее.

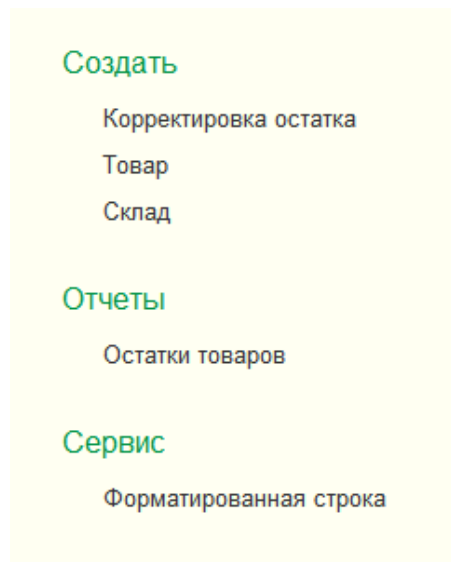


Рис. 14. Команды действий

СОВЕТ. При работе с клавиатуры для перехода на панель действий текущего раздела нужно нажать **Alt + 3**.

Существует возможность изменять состав и порядок команд действий в группах в меню функций в форме настройки (подробнее см. раздел «[Панель действий](#)» [здесь](#)). Диалог настройки списка команд действий открывается командой [Настройка действий](#) меню функций.

1.3. Панель команд текущего раздела

В данной панели отображаются команды выбранного раздела. Например, для раздела [Товарные запасы](#) в данной панели отобразятся команды [Товары](#) (команда открытия списка товаров) и [Корректировки остатков](#).

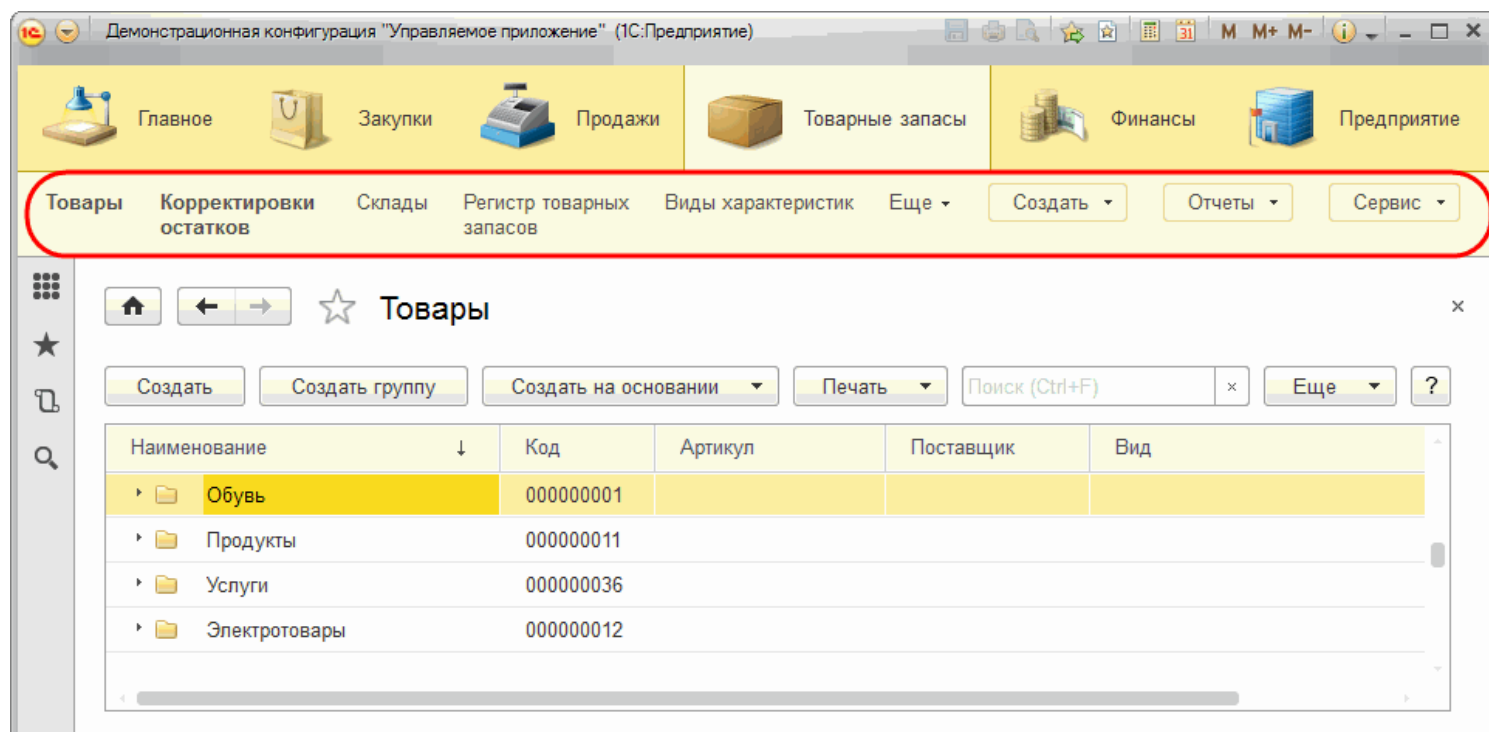


Рис. 15. Панель команд текущего раздела

При нажатии на гиперссылку элемента панели команд выполнится соответствующая команда (например, откроется форма списка товаров).

1.4. Панель открытых

Панель содержит список открытых форм.

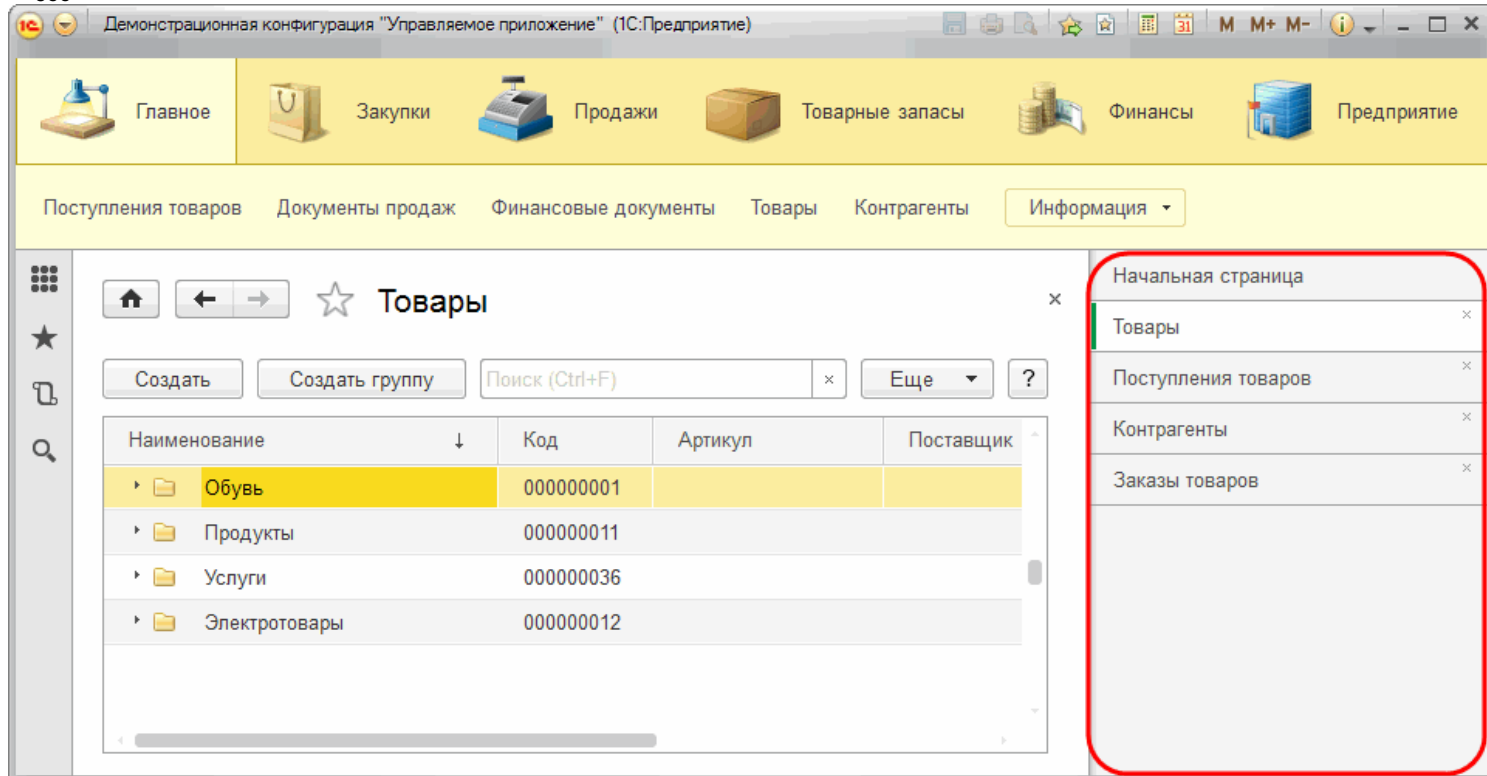


Рис. 16. Панель открытых

Если на начальной странице отображаются формы, то в списке присутствует пункт [Начальная страница](#). Строки списка упорядочены в порядке открытия (новые попадают в конец списка). При выборе элемента активизируется соответствующая форма.

Чтобы закрыть форму из панели, нужно нажать крестик в строке списка.

1.5. Панель инструментов

Панель инструментов по умолчанию располагается слева от основной рабочей области и выглядит следующим образом:



Рис. 17. Панель инструментов

На панели инструментов содержатся команды открытия следующих форм:

- меню функций для текущего раздела ([Действия](#)), см. [здесь](#);
- форма избранного ([Избранное](#)), см. раздел «[Избранное](#)» [здесь](#);
- форма истории ([История](#)), см. раздел «[История](#)» [здесь](#);
- форма поиска ([Поиск](#)), см. раздел «[Поиск в данных](#)» [здесь](#).

1.6. Область системных команд

Заголовок приложения является составным и содержит область системных команд.

В левой части **заголовка** расположено главное меню.



Рис. 18. Командная панель окна приложения

Главное меню содержит только общие команды – это команды для работы с файлами, команды управления окнами, сервисные команды и др.

СОВЕТ. При работе с клавиатуры для перехода в главное меню текущего окна нужно нажать клавишу **F10**.

В центре располагается заголовок приложения.



Рис. 19. Правая часть области системных команд

В правой части области системных команд по умолчанию расположены команды работы с файлами, работы со ссылками и стандартные функции, такие как вызов календаря, калькулятора и т. д.

Закрытие основного окна и выход из программы осуществляются при вызове команды **Файл – Выход** главного меню или при нажатии кнопки **Заккрыть** основного окна. Поддерживается возможность навигации по программе при помощи клавиатуры. Таблицы сочетаний клавиш для навигации содержатся во встроенной справке.

1.7. Форма объекта

В заголовке формы отображаются кнопки перехода к начальной странице перехода между открытыми формами и т. д.

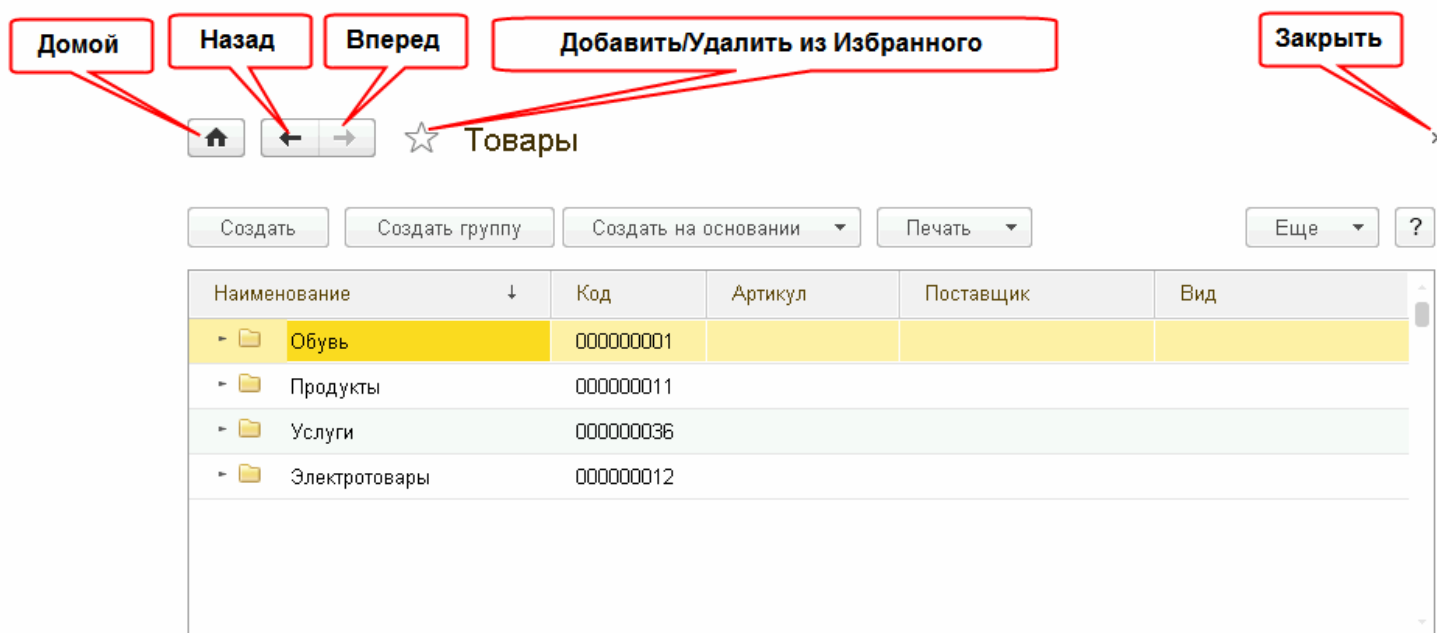


Рис. 20. Форма объекта

Чтобы активизировать начальную страницу, нужно нажать кнопку **Домой**. Данная кнопка активна, если в программе заданы формы для начальной страницы.

Для возврата к предыдущей активной форме нажать кнопку **Назад**. Для перехода к форме, которая была активной после текущей, нужно нажать кнопку **Вперед**. Если форма была активизирована последней, кнопка перехода недоступна.

Чтобы добавить открытую форму в список избранных или удалить ее из списка, следует использовать переключатель в форме звездочки. Если звездочка активна, это означает, что ссылка на элемент находится в списке избранных (о работе с избранным см. раздел «Избранное», [здесь](#)).

Для закрытия формы нажать кнопку **Заккрыть**.

Панель навигации формы

Если для формы реализована панель навигации, то она отображается горизонтально под заголовком формы.

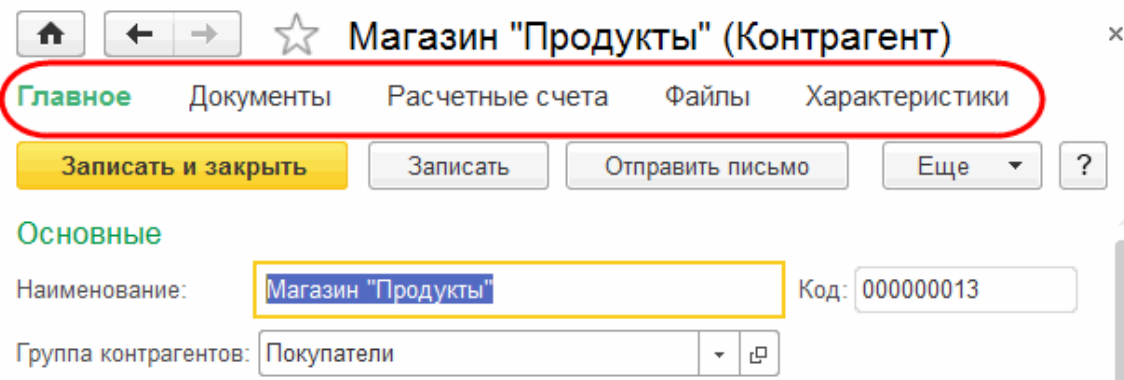


Рис. 21. Панель навигации формы

Панель позволяет переходить к просмотру различных сведений, логически связанных с данными, которые отображает основная форма объекта.

Команда текущей формы в панели выделяется цветом. При нажатии на пункт **Главное** будет активирована основная форма текущего объекта. При нажатии ссылки в панели навигации формы активируется вспомогательная форма объекта (см. рис. ниже).

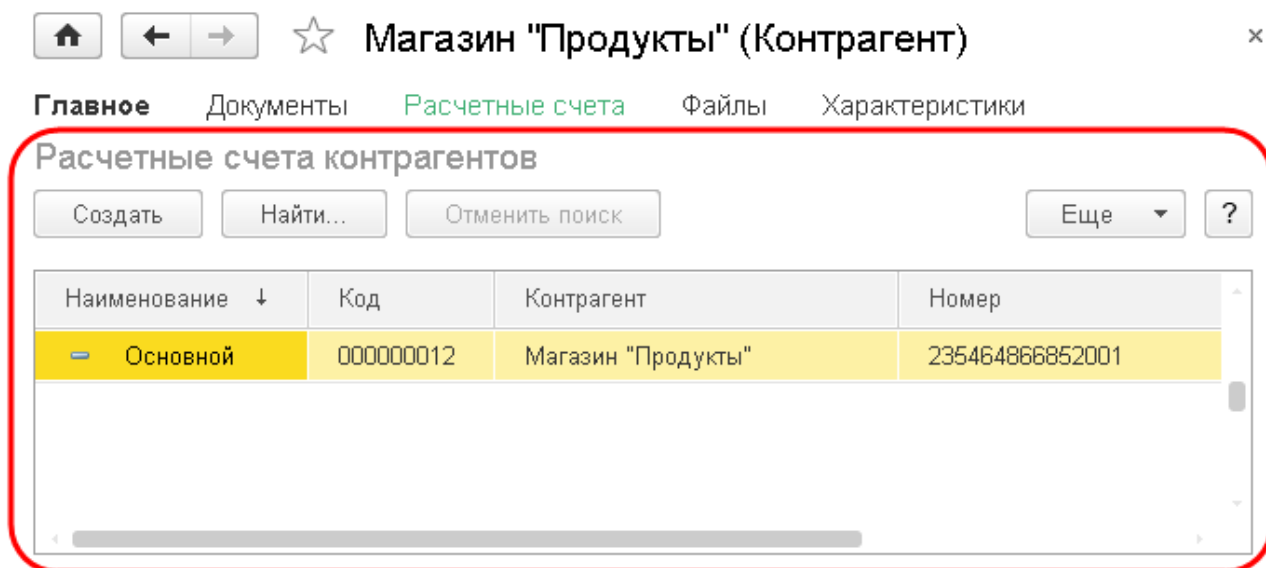


Рис. 22. Форма списка счетов контрагента (вспомогательная)

В панели навигации могут располагаться команды из групп **Перейти** и **См. также**. В зависимости от программы на панели навигации могут отображаться и другие команды.

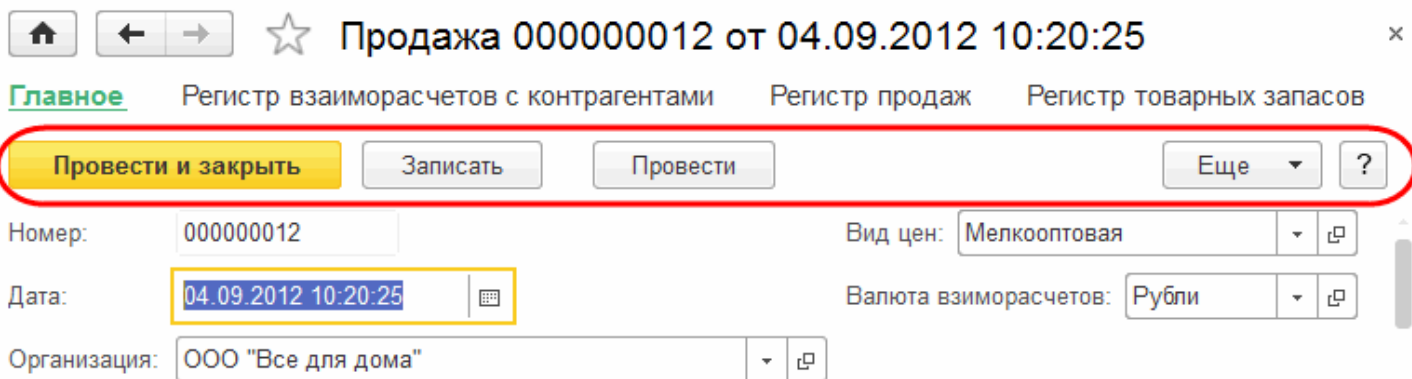
Если переход по ссылке осуществляется для еще не сохраненного объекта, система выводит предупреждение: **Данные еще не записаны. Переход к """" возможен только после записи данных. Данные будут записаны.**

При нажатии кнопки **OK** текущие данные будут сохранены и будет выполнен переход к выбранной области данных. При нажатии кнопки **Отмена** сохранение и переход не выполняются, и можно продолжить редактирование текущих данных.

Панели навигации формы настраиваются командой главного меню **Вид – Настройка панели навигации формы**. Процесс настройки списка команд навигации аналогичен настройке списка команд навигации раздела. Подробнее см. раздел [«Панель навигации»](#), [здесь](#).

1.7.1. Командная панель формы

Командная панель формы содержит команды, непосредственно связанные с объектом, который отображается в основной форме. Команды отображаются на панели в виде кнопок.



Продажа 000000012 от 04.09.2012 10:20:25

[Главное](#) Регистр взаиморасчетов с контрагентами Регистр продаж Регистр товарных запасов

Провести и закрыть Записать Провести Еще ?

Номер: 000000012 Вид цен: Мелкооптовая

Дата: 04.09.2012 10:20:25 Валюта взаиморасчетов: Рубли

Организация: ООО "Все для дома"

Рис. 23. Командная панель формы

Кнопка, используемая по умолчанию (и при нажатии **Enter**), выделяется на панели полужирным начертанием текста и цветом. На рисунке выше кнопкой по умолчанию является кнопка [Провести и закрыть](#).

Меню [Еще](#) содержит список всех доступных команд формы.

Команда [Еще - Показать в списке](#) открывает команды перехода к спискам, содержащим объект, при этом фокус будет установлен на строке, соответствующей данному объекту.

Глава 2. Работа в формах

В «1С:Предприятии» пользователь работает с формами. Форма может содержать различные элементы, с помощью которых информация отображается и может быть изменена пользователем: поля ввода, командные панели, кнопки, флажки, закладки, таблицы, различные диаграммы, поля текстового, табличного и HTML-документов, поля географической и графической схем. Данная глава содержит описание общих принципов работы с формами и элементами форм.

2.1. Общие принципы работы с формами

Существует несколько общих элементов, которые имеет каждая форма. Обычно в заголовке формы отображается название объекта или списка объектов, которые она отображает. Название формы задается в процессе настройки конфигурации и не может быть изменено пользователем.

Для **выбора объекта** следует нажать левую кнопку мыши, установив указатель мыши на объект, при работе с клавиатуры использовать клавиши управления курсором.

Если поле объекта имеет затененный фон, это означает, что объект недоступен для редактирования.

Для **перехода к редактированию объекта** дважды следует нажать левую кнопку мыши или клавишу **Enter**.

Элементы формы могут иметь подсказку, в которой при настройке конфигурации был описан их смысл. Чтобы получить подсказку по конкретному реквизиту формы, следует поместить над ним указатель мыши. Надпись, поясняющая назначение реквизита, появится возле указателя через 1–2 секунды (если такая надпись была создана при настройке формы).

Перемещение по элементам формы. Для перемещения по элементам формы можно использовать клавиши **Tab** и **Shift + Tab** или просто щелкать мышью на соответствующем элементе диалога.

Некоторые элементы могут быть настроены таким образом, что для перехода по **Tab** и **Shift + Tab** они недоступны (например, командная панель). Для перехода к таким элементам следует использовать мышь или комбинацию клавиш **Alt + F10**.

Также можно завершать ввод очередного реквизита нажатием клавиши **Enter**, тогда курсор будет автоматически переходить на следующий элемент диалога. Последовательность перехода определяется при разработке конфигурации.

Если это обусловлено конфигурацией, форма может проверять корректность и полноту введенных данных. В случае неправильного ввода могут появляться соответствующие сообщения.

Кроме того, после завершения ввода реквизита системой могут выполняться некоторые автоматические действия, например, заполнение значений других реквизитов формы.

Сохранение параметров формы. Если форма может изменять размер, то при закрытии и ее повторном открытии форма будет иметь те же размеры и располагаться в том же месте экрана, что и при ее закрытии. Элементы управления при этом сохраняют размеры, какими они были при закрытии формы.

Внешний вид формы определяется конфигурацией и может быть изменен пользователем, если доступна возможность настроить форму (пункт меню **Еще – Изменить форму**). Порядок настройки форм излагается в разделе [«Настройка формы»](#) [здесь](#).

2.2. Поля

Поля предназначены для просмотра, ввода и редактирования данных различных типов. Чтобы ввести значение в поле, нужно привести курсор на область поля, нажать левую кнопку мыши и ввести значение – обычно с помощью клавиатуры или выбором из списка.

2.2.1. Поле ввода

Поле ввода предназначено для просмотра, ввода и редактирования данных различных типов. Чтобы ввести значение в поле, нужно привести курсор на область поля, нажать левую кнопку мыши и ввести значение – обычно с помощью клавиатуры или выбором из списка.

Приемы работы с полем ввода зависят от типа данных, редактируемых или просматриваемых в элементе формы, а также от настроек, заданных в конфигурации.

Рис. 24. Пример формы с полями ввода

Обычно для того чтобы ввести данные в поле, достаточно перейти к нужному полю ввода, нажав левую кнопку мыши или клавишу **Tab**. При этом доступны возможности редактирования символьных строк, предоставляемые операционной системой, например, работа с буфером обмена.

Красное подчеркивание означает, что данное поле необходимо заполнить.

Более бледная рамка у поля означает, что данное поле заполняется автоматически или не рекомендуется к заполнению (например, поле **Код** на [рис. 24](#) заполняется при записи автоматически). При попытке изменить значение в таком поле вручную система выдаст соответствующее предупреждение.

В случае ввода некорректных данных в выпадающем списке отобразится информация о том, что значение не найдено.

Нажатие клавиши **F4** открывает список элементов для выбора (см. [рис. 25](#)) или всплывающий список (если предусмотрено в программе).

Рис. 25. Выбор группы контрагентов из списка

Нажатие сочетания клавиш **Ctrl + Shift + F4** открывает форму выбранного элемента.

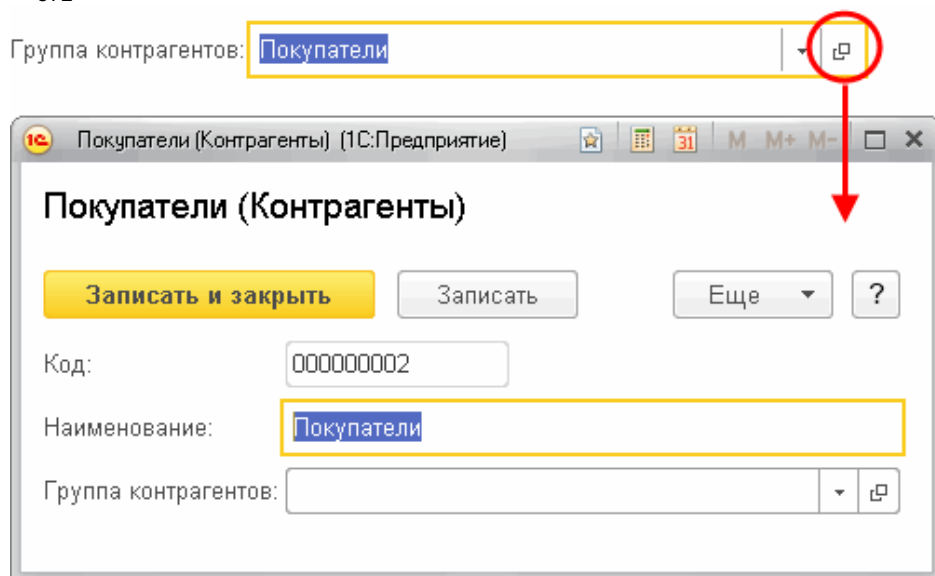


Рис. 26. Открытие формы выбранного элемента из поля ввода

Осветленная рамка поля ввода (как у поля **Код** на [рис. 26](#)) означает, что редактировать значение в этом поле не следует.

Если для поля ввода установлен **многострочный режим**, то при нажатии клавиши **Enter** управление не переходит к следующему элементу формы, а в данном поле ввода начинается новая строка.

Для многострочного поля ввода может быть установлен расширенный режим ввода, при котором допускается использование символа табуляции, а также стандартный поиск текста. Следует учитывать, что для данного режима недоступен переход к следующему элементу формы по клавише **Tab**.

Если поле ввода имеет ограниченный размер данных, то при вставке в поле ввода данных из буфера обмена будет взята такая часть исходного текста, чтобы заполнить остаток места (или выделенный текст).

2.2.1.1. Строка

Если поле ввода связано со строковыми данными, то оно будет иметь вид:

Наименование:

Рис. 27. Пример поля ввода типа «Строка»

В незаполненном поле ввода, связанном со строковыми данными, в зависимости от конфигурации может отображаться подсказка ввода. Подсказка отображается в пустом поле ввода, скрывается при вводе текста и не влияет на вводимое значение.

Наименование:

Рис. 28. Подсказка в поле ввода

2.2.1.2. Число

Если поле ввода связано с числовыми данными (например, количество, сумма), то кнопка выбора будет иметь форму калькулятора, и при нажатии на нее число будет редактироваться с помощью отобразившегося калькулятора.



Рис. 29. Ввод числа в поле ввода при помощи калькулятора

Также для поля может быть настроено управление значением с помощью регулятора.



Рис. 30. Изменение значения в поле ввода с помощью регулятора «Количество»

2.2.1.3. Дата

Если поле ввода предназначено для ввода даты, то в зависимости от настроек программы пустое поле даты будет отображать разделительные точки и двоеточия.

При вводе даты существует возможность использовать встроенный календарь:

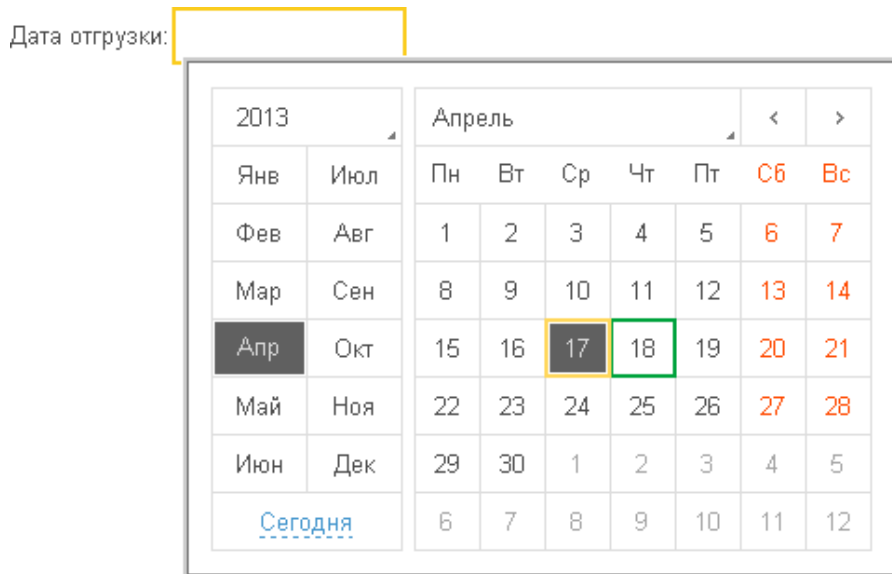


Рис. 31. Ввод даты в поле ввода с помощью календаря

Если поле ввода содержит дату и время, а отображается только дата или дата и часть времени, то при редактировании даты в поле ввода неотображаемая часть времени остается такой же, какой была до изменения.

2.2.1.4. Данные различных типов

Помимо рассмотренных выше типов значения реквизита, в форме существует возможность вводить значения, которые могут относиться к одному из нескольких типов. В поле ввода такого реквизита присутствует кнопка **Выбрать**. При нажатии этой кнопки на экран будет выведен список типов, из которого требуется выбрать необходимый.

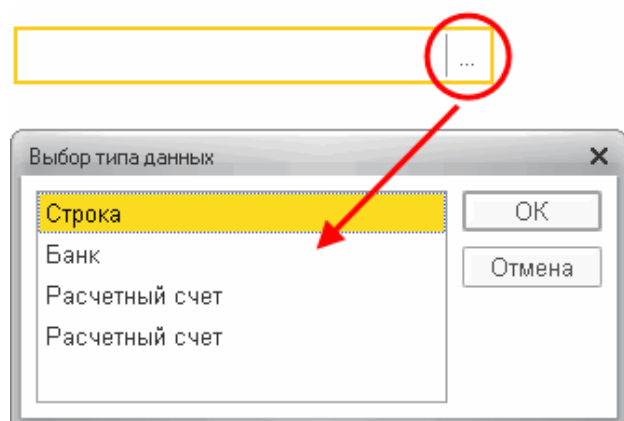


Рис. 32. Ввод данных выбранного типа

В этом окне необходимо выбрать наименование требуемого типа. Далее следует в открывшемся списке выбрать требуемое значение.

2.2.1.5. Выбор значения объекта

Существует возможность выбирать значение объекта без открытия отдельного окна – в списке, который всплывает при позиционировании курсора в поле ввода или при нажатии кнопки выбора в поле (или клавиши [Стрелка вниз](#), кроме многострочных полей).

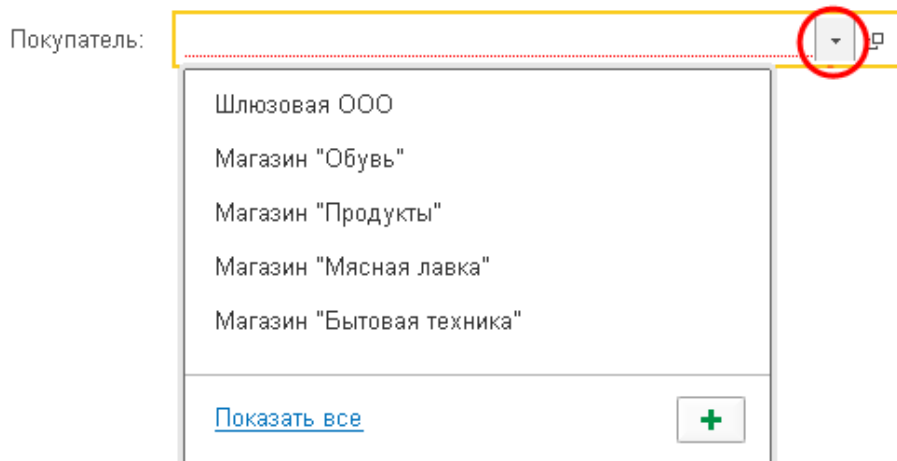


Рис. 33. Поле ввода со всплывающим списком

СОВЕТ. При работе с клавиатуры для выбора элемента также используется [F4](#), а сочетание клавиш [Ctrl + Shift + F4](#) открывает форму выбранного элемента.

Необходимая строка списка выбирается нажатием левой кнопки мыши.

Всплывающий список также может хранить и отображать историю выбранных значений. При позиционировании в поле ввода во всплывающем списке отображаются последние выбранные значения данных, что позволяет ускорить выбор данных, когда необходимо выбрать ранее использованные данные.

Поиск значения в поле ввода. При позиционировании в поле ввода может быть предложено ввести строку для поиска, отобразить целиком список данных (команда [Показать все](#) или клавиша [F4](#)) и/или создать новый объект (команда [Создать](#) или клавиша [F8](#)).

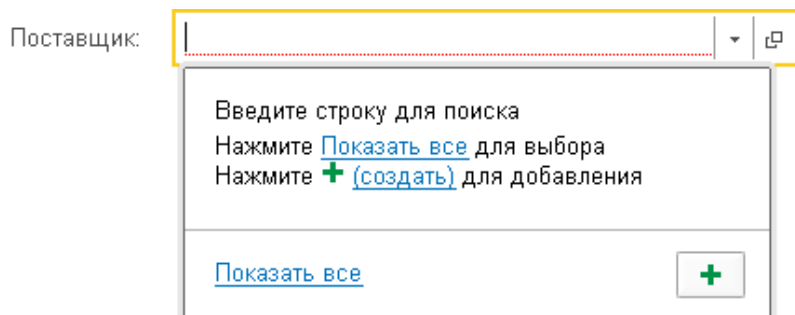


Рис. 34. Всплывающий список в поле ввода

Если введенного значения в списке нет, программа предложит отобразить все значения или создать новый объект.

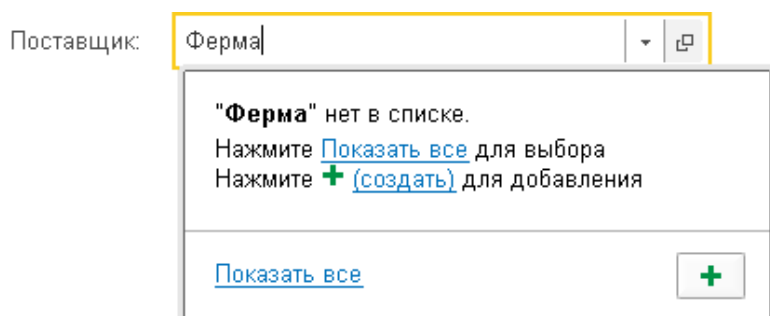


Рис. 35. Значение в списке не найдено

При начале ввода текста программа находит данные, отвечающие набранному тексту, и выводит их в виде выпадающего списка. В зависимости от настроек программы поиск может выполняться по началу строки или по любой части строки. Например:

Поставщик: маг

Магазин "Бытовая техника" (000000014)
Магазин "Мясная лавка" (000000016)
Магазин "Обувь" (000000015)
Магазин "Продукты" (000000013)

[Показать все](#)

Рис. 36. Поиск по началу строки в поле ввода

Если для данных в поле установлена возможность поиска по любой части, достаточно ввести любой фрагмент текста, как показано на рисунке:

Товар	Артикул
оло	


Молоко (000000026)
Молоко 1,5% жирности (000000041)
Молоко 2,5% жирности (000000042)
Молоко 3,5% жирности (000000043)

[Показать все](#)

Рис. 37. Поиск по любой части строки в поле ввода

Если данных достаточно много и их обработка требует значительного количества времени, то во время поиска будет отображено соответствующее сообщение.

Покупатель:

 Идет поиск данных...

[Показать все](#)

Рис. 38. Поиск данных в фоне

Если в поле выбраны данные, помеченные на удаление, будет выведено предупреждение.

Если для набранного текста не найдено соответствий, то система предложит выбрать одно из действий – выбрать из списка, показать все или создать новый объект (если такая возможность установлена в программе).

При выборе команды [Создать](#) (или при нажатии клавиши **F8**) откроется форма создания нового объекта.

Во время выбора в выпадающем списке поля ввода при щелчке мышью на другом элементе формы выпадающий список закроется, а значение, установленное в нем, не изменится.

ПРИМЕЧАНИЕ. Для реквизитов, тип которых не определен (например, выбран составной тип), программа не позволит ввести текст до тех пор, пока не будет указан конкретный тип данных.

Полнотекстовый поиск данных при вводе. В зависимости от настроек конфигурации для поиска данных в поле ввода может быть использован полнотекстовый поиск. При этом дается возможность ввести одно или несколько слов. Поиск будет выполнен по любому из введенных слов.

N	Товар
1	мол фер

Молоко "Продукты с фермы" 1.5% ...

Молоко "Продукты с фермы" 3.2% ...

Молоко "Продукты с фермы" 3.2% топленое...

[Показать все](#) +

Рис. 39. Полнотекстовый поиск в поле ввода

Во время полнотекстового поиска будут найдены строки, содержащие искомые слова во всех сочетаниях. Будут найдены как целые слова, так и строки, в которых вводимые строки являются частью целых слов. (аналогично действию оператора * полнотекстового поиска).

Отказ от выбранного значения. Значение, которое вводится путем выбора из списка, очищается с помощью кнопки очистки, расположенной справа от поля реквизита (если такая кнопка определена при конфигурировании) или клавишами **Shift + F4**. При этом в поле будет установлено значение по умолчанию для данного поля.

Значение:

Рис. 40. Поле ввода с кнопкой очистки

Ввод и возврат к предыдущему значению отменяются клавишей **Esc**.

В многострочном поле ввода при нажатии клавиши **Esc** на экране отобразится вопрос о сохранении изменений. Чтобы сохранить выполненные изменения, нужно нажать кнопку **Да**. Чтобы отказаться от изменения, нажать **Нет**. Чтобы вернуться к редактированию поля, нажать кнопку **Отмена**.

2.2.2. Флажок

Элемент формы **Флажок** предназначен для включения или выключения некоторого значения.

Используются флажки, имеющие два либо три состояния.

Учет по складам:

Рис. 41. Пример флажка на форме

Если флажок, расположенный в колонке таблицы или на форме, недоступен для редактирования, то он отображается особым способом.

Для переключения значения флажка следует щелкнуть по нему мышью или (при использовании клавиатуры) активизировать его клавишами **Tab** и **Shift + Tab** и нажать **Пробел**.

Для флажка с двумя состояниями также поддерживается переключение клавишами **+** (плюс) и **-** (минус).

2.2.3. Переключатель

Элемент формы **Переключатель** предназначен для выбора одного из нескольких возможных значений. Требуемое значение выбирается с помощью мыши. При использовании клавиатуры следует активизировать текущее выбранное значение переключателя клавишами **Tab** и **Shift + Tab**, клавишами управления курсором выбрать требуемое значение и нажать клавишу **Enter**. Например, в форме товара предлагается выбор вида товара:

Вид: Товар Услуга

Рис. 42. Переключатель вида товара

2.3. Тумблер

Элемент формы **Тумблер** предназначен для переключения значений (как переключатель) или для включения значения (аналогично флажку). В самом простом виде тумблер выглядит, как показано на рисунке:

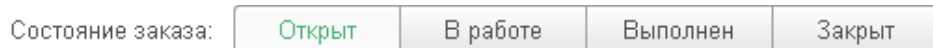


Рис. 43. Тумблер состояния заказа

Для выбора значения нужно нажать соответствующую кнопку тумблера. У нажатой кнопки активизируется зеленый индикатор, и она будет отображена ярче.

2.4. Индикатор

Индикатор используется только для отображения текущего состояния реквизита формы, с которым он связан (определяется при проектировании конфигурации).

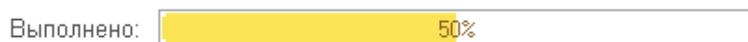


Рис. 44. Индикатор

2.5. Полоса регулирования

Элемент формы **Полоса регулирования** предназначен для быстрого и наглядного ввода числовых значений. По внешнему виду он напоминает обычный линейный регулятор (на рисунке под полем ввода, с которым он связан), расположенный вертикально или горизонтально.

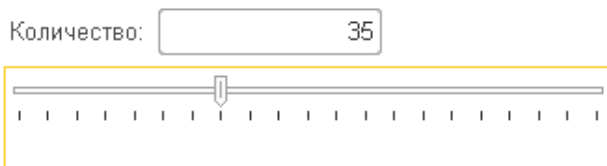


Рис. 45. Регулятор значения

Значение реквизита, связанного с этим элементом, определяется положением бегунка и значениями минимума и максимума, установленными при проектировании конфигурации.

Положение бегунка может быть изменено клавишами управления курсором, но с помощью мыши удобнее.

2.6. Гиперссылка

В форме могут располагаться элементы управления, представляющие собой текст, при наведении указателя мыши на который указатель принимает вид руки (см. [рис. 46](#)) и появляется подчеркивание. Такой элемент называется **Гиперссылка**. Он предназначен для перехода к некоторому объекту (например, открытия других форм). Для выбора гиперссылки следует щелкнуть по ней мышью или сделать ее активной клавишами **Tab** и **Shift + Tab** и нажать клавишу **Enter**. Настройка отработки нажатия гиперссылки определяется при проектировании конфигурации. Гиперссылку могут иметь и картинки, размещенные в форме. При поднесении указателя мыши к картинке с гиперссылкой форма указателя также принимает вид, как на [рис. 46](#).



Рис. 46. Гиперссылка на форме

2.7. Кнопка

Кнопки в форме используются для выполнения некоторых действий, предусмотренных алгоритмом формы. Для выбора кнопки следует щелкнуть по ней мышью или сделать ее активной клавишами **Tab** и **Shift + Tab** и нажать клавишу **Enter**.

Одна из кнопок в форме может быть предопределенной, и ее нажатие может выполняться не только мышью, но и с помощью клавиш **Ctrl + Enter**.

Добавить

Рис. 47. Кнопка на форме

Для некоторых кнопок могут быть предусмотрены другие комбинации клавиш, которые позволяют нажать эту кнопку без использования мыши и без перехода к ней последовательным обходом элементов формы.

Для некоторых кнопок может быть задан режим отображения нажатия. Такие кнопки используются при установке или отмене какого-либо режима.

Если действие, связанное с кнопкой, недоступно, кнопка также будет отображаться как недоступная.

Добавить

Рис. 48. Кнопка не доступна

2.8. Таблица

Форма может содержать таблицу, предназначенную для просмотра списка с несколькими колонками. Состав колонок определяется назначением конкретной формы, а строки являются собственно данными, которые просматриваются в списке и в некоторых случаях могут редактироваться. В качестве примера приведем список номенклатуры в документах [Продажа](#).

Продажа 000000009 от 13.08.2012 14:15:30
×

[Главное](#)
[Регистр взаиморасчетов с ...](#)
[Регистр продаж](#)
[Регистр товарных запасов](#)

Номер:
 Вид цен:

Дата:

Организация:

Товары (2)

N	Товар	Цена	Количество	Сумма
1	Bosch1234	2 500,00	2,00	5 000,00
2	Bosch15	3 200,00	3,00	9 600,00

Количество (итог): **5,00**
Сумма (итог): **14 600,00**

Рис. 49. Список товаров в документе «Продажа»

В некоторых формах колонки располагаются таким образом, что значения колонок в каждой строке таблицы выводятся в несколько уровней (два или более). Такой способ позволяет одновременно выводить в строке большое количество информации без прокручивания таблицы по горизонтали.

Если высота ячейки таблицы равна одной строке, а в ячейке при этом содержится несколько строк текста, то отображается только первая строка текста, которая заканчивается многоточием.

Добавление, копирование, удаление, изменение данных, сортировка, отбор и другие действия с данными, размещаемыми в списке, обычно осуществляются с помощью кнопок командной панели списка или с помощью команд контекстного меню, вызываемых в области списка, если выполнение подобных действий предусмотрено при конфигурировании.

Приемы работы с таблицей зависят от типа данных, редактируемых или показываемых в элементе формы, а также от настроек, заданных в конфигурации или определенных средствами встроенного языка, и в основном совпадают с правилами работы с полем ввода. Для начала редактирования ячейки нужно нажать клавишу **Enter** или выбрать ячейку и начать ее редактировать. Чтобы завершить редактирование строки таблицы, нужно нажать клавиши **Shift + F2**.

Просмотр таблицы. При просмотре таблицы существует возможность использовать мышь и клавиатуру.

Для удобства просмотра содержимого таблицы ширина колонок может быть изменена. Для этого следует подвести указатель мыши к границе колонок, пока он не изменит вид, и нажать левую кнопку мыши. Перемещая мышь, установить нужную ширину колонок, затем отпустить кнопку мыши. Установленные размеры колонок запоминаются. Если при изменении ширины колонки нажать клавишу **Ctrl**, то будет изменяться ширина только данной колонки. Данный прием можно использовать, если для каких-либо системных списков не предусмотрены заголовки.

Если ширина колонки не позволяет увидеть всего содержимого поля, то следует поместить указатель мыши над нужной колонкой в нужной строке, и возле указателя появится содержимое поля полностью.

Существует возможность изменять порядок следования колонок при просмотре таблицы. Если необходимо установить порядок следования колонок, следует воспользоваться настройкой формы, описанной [здесь](#).

В некоторых таблицах допускается использование множественного выбора строк списка. Удерживая нажатой клавишу **Ctrl**, можно выбрать необходимые строки с помощью мыши или клавиш управления курсором.

Для листания таблицы постранично используются клавиши **PgUp** и **PgDown**.

Для перехода на первую строку таблицы служат клавиши **Ctrl + Home**; на последнюю – **Ctrl + End**.

Корректировка ячейки таблицы. Для корректировки реквизита таблицы следует выбрать ячейку, содержащую этот реквизит, и начать редактирование. Если для ячейки предусмотрен список значений, следует нажать **Ctrl + Стрелка вниз**, чтобы открыть его и выбрать значение из списка. Для окончания редактирования ячейки – нажать клавишу **Enter**. Если для редактирования данных в ячейке используется поле ввода, то доступны приемы работы для поля ввода. Подробнее см. [здесь](#).

Ввод строки. Для ввода в таблицу новой строки следует нажать клавишу **Insert**. Новая ячейка строки автоматически переключается в режим редактирования. Нажатие клавиши **Enter** после указания значения реквизита переводит курсор в следующую ячейку.

Возможность перехода между ячейками строки во время редактирования с помощью повторного нажатия **Enter** или клавиши **Tab** устанавливается на этапе конфигурирования.

Также новая строка может быть создана автоматически при нажатии клавиши **Стрелка вниз** в последней строке таблицы, если такая возможность была установлена в конфигурации. Если при редактировании новой строки не была введена информация ни в одну ячейку, то новая строка будет удалена.

ПРИМЕЧАНИЕ. Если при вводе строки в таблицу курсор после нажатия клавиши **Enter** перескакивает через ячейки, значит, в данной конфигурации установлен пропуск этих колонок таблицы.

Удаление строки. Предусмотрена возможность удаления строк таблицы.

ВНИМАНИЕ! Удаление строки производится без дополнительного предупреждения.

Для удаления строки таблицы используются клавиши управления курсором, или можно мышью выбрать строку и нажать клавишу **Delete**.

Копирование строки. Чтобы скопировать строку, нужно выбрать клавишами управления курсором или мышью строку, которая будет служить образцом, и нажать клавишу **F9**. В таблицу будет добавлена новая строка, данные которой будут заполнены значениями, скопированными из данных строки-образца.

Номера строк. Крайняя левая колонка таблицы содержит номера строк. При вводе новой строки в таблицу ей автоматически присваивается очередной порядковый номер.

На форме может быть определен порядок следования строк таблицы. При этом существует возможность передвигать строки вверх и вниз. Для перемещения строк используются команды [Переместить вверх](#) и [Переместить вниз](#). При перемещении строк их номера автоматически изменяются.

Сортировка строк таблицы. Строки в таблице могут быть упорядочены произвольным образом, если это обусловлено конфигурацией системы. Чтобы установить сортировку строк по данным любой колонки, нужно выбрать колонку и выполнить команду меню [Еще – Сортировать по убыванию](#) или [Сортировать по возрастанию](#). Также существует возможность перетаскивать строки таблицы с помощью мыши.

Если необходимо установить более сложное упорядочивание элементов таблицы, следует воспользоваться настройкой формы, описанной [здесь](#).

2.9. Группы

Система предоставляет возможность объединять различные элементы формы в группы. Это могут быть группы полей, колонок таблицы или списка, страницы, группы команд. Также колонки списка могут быть объединены в вертикальные или горизонтальные группы. В качестве примера вертикальная группа колонок списка на форме будет выглядеть следующим образом:

N	Товар	Цена
	Артикул	
1	Молоко	30,00
	МоI34	

Рис. 50. Группа колонок таблицы

Элементы на форме могут визуально распределяться по группам. Группа может быть обособлена чертой или выделена рамкой. Например, как показано на рисунке.

Данные покупателя

Номер:	<input type="text" value="000000016"/>
Дата:	<input type="text" value="10.09.2012 14:21:40"/> <input type="button" value="📅"/>
Организация:	<input type="text" value="ООО " товары"=""/> <input type="button" value="▼"/> <input type="button" value="📄"/>
Покупатель:	<input type="text" value="Магазин " продукты"=""/> <input type="button" value="▼"/> <input type="button" value="📄"/>
Склад:	<input type="text" value="Малый"/> <input type="button" value="▼"/> <input type="button" value="📄"/>

Рис. 51. Группа полей на форме

При настройке формы существует возможность перемещать элементы формы между группами. При этом системой определяется допустимость такого переноса. Если перенос требует изменения каких-либо свойств элемента, изменение осуществляется автоматически. Например, при переносе группы элементов в группу с видом [Страницы](#) ее свойство [Вид](#) изменится с [Обычная группа](#) на [Страница](#). Если в результате изменились требования к подчиненным элементам, они либо автоматически изменяются, либо удаляются.

На форме могут располагаться группы, которые можно сворачивать. Такие группы отображаются особым образом.



Адреса и контакты

Регион:	<input type="text"/> <input type="button" value="▼"/> <input type="button" value="📄"/>
Страна:	<input type="text"/>
Город:	<input type="text"/>

Рис. 52. Сворачиваемая группа

При нажатии кнопки в заголовке такой группы ее содержимое скрывается.

Чтобы раскрыть содержимое группы, следует снова нажать кнопку.

Подробнее о настройке форм см. раздел «[Настройка формы](#)» [здесь](#).

2.9.1. Командная панель

Форма может содержать группы элементов вида [Командная панель](#). Каждая командная панель представляет собой набор кнопок (команд), действие которых распространяется только на данную форму или на отдельную часть формы.

При работе с клавиатуры командная панель участвует в обходе элементов формы по клавише **Tab** или **Shift + Tab**. Обходятся все кнопки, а при нажатии кнопки панели (клавишей **Enter**) выполняется действие.

2.9.2. Страницы

Форма может содержать страницы, если это обусловлено конфигурацией. Закладки страницы могут быть расположены вертикально (сверху, снизу) или горизонтально (слева, справа) от содержимого страницы. Для выбора определенной страницы необходимо щелкнуть на ней левой кнопкой мыши. Также между страницами можно переключаться, используя сочетания клавиш **Ctrl + PgUp** (вперед) или **Ctrl + PgDn**.

Каждая такая страница может содержать какие-либо «свои» элементы формы. Например, форма документа [Расходная накладная](#) может содержать закладку [Товары](#), на которой расположен список товаров, и закладку [Прочее](#) с полем для дополнительного описания отгрузки (как показано на [рис. 53](#)).

N	Товар	Цена	Количество	Сумма
1	Sony K3456P	10 000,00	1,00	10 000,00
	H657			
2	VekoNT02	7 000,00	1,00	7 000,00
	NT02			
Количество (итог): 2,00		Сумма (итог): 17 000,00		

Рис. 53. Страницы в документе «Расходная накладная»

2.10. Диаграмма

Диаграмма может быть встроена в форму и в табличный документ.

Существует возможность управлять диаграммой, встроенной в форму. Настройка производится с помощью реквизитов, размещенных в форме (состав определяется при конфигурировании). Если позволяют исходные данные, допускается выбор другого типа диаграммы. О работе с диаграммой в табличном документе см. приложение «Редактор табличных документов», поставляемое в электронной версии документации.

2.11. Поле текстового документа

[Поле текстового документа](#) предназначено для работы с текстами. В нем размещается окно текстового редактора, предназначенное для ввода и сохранения строк текста. Подробнее о работе с редактором см. приложение «Редактор текстового документа», доступное в электронной версии документации.

2.12. Поле HTML-документа

Поле HTML-документа предназначено для просмотра HTML-документа на форме. При просмотре информации, размещенной в поле, доступен переход по ссылкам (если такие предусмотрены), поиск с помощью команд главного меню **Правка: Найти, Найти следующий, Найти предыдущий**, переход между страницами с помощью команд контекстного меню **Вперед, Назад**.

ПРИМЕЧАНИЕ. Если на поле отображается внешний ресурс, то поиск выполняется средствами браузера.

Для данного поля доступны команды меню **Файл: Печать, Сохранить как**.

2.13. Поле графической схемы

Элемент формы **Поле графической схемы** предназначен для отображения и редактирования графической схемы.

Редактирование графических схем описано в приложении 4 «Редактор графической схемы», доступном в электронной версии документации.

Порядок обхода элементов схемы определяется при конфигурировании.

Команда **Графическая схема – Масштаб** предназначена для выбора масштаба отображения графической схемы.

Аналогичные действия по изменению масштаба можно выполнять с помощью мыши, у которой имеется колесо прокрутки, при нажатой клавише **Ctrl**.

В этом поле могут отображаться общие карты маршрута бизнес-процессов или карты маршрута конкретных экземпляров бизнес-процессов с учетом пройденных и активных точек маршрута.

В качестве примера отображения карты маршрута в графической схеме приведем карту маршрута бизнес-процесса продажи.

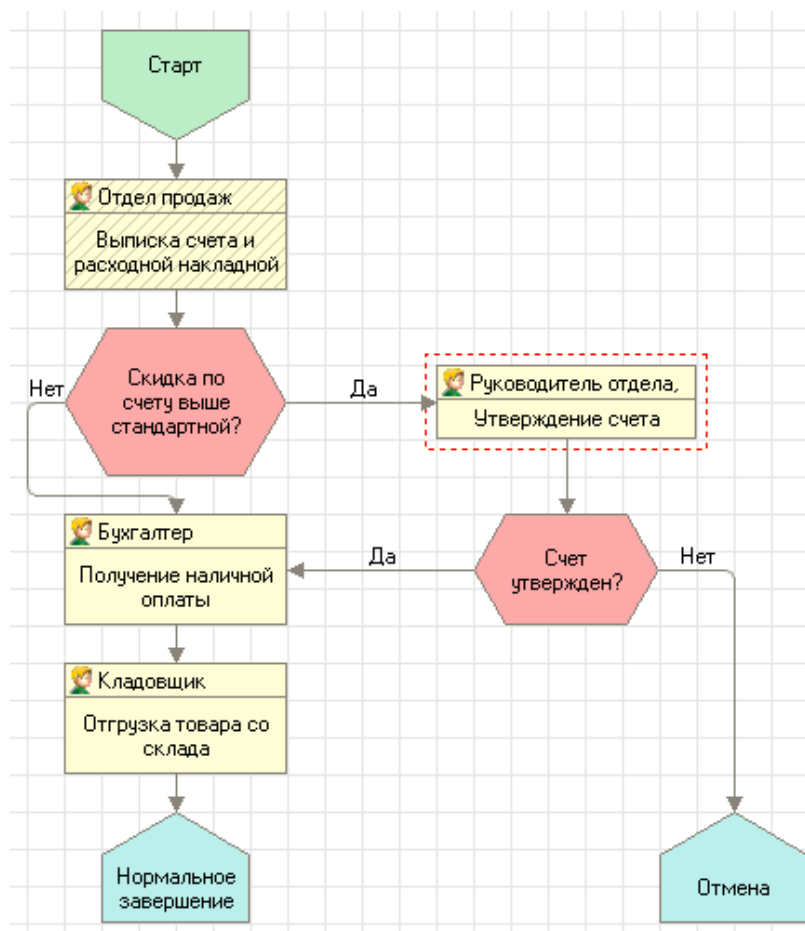


Рис. 54. Графическая схема карты маршрута

Пройденные точки маршрута отображаются заштрихованным фоном (на рисунке – **Выписка счета**). Точка маршрута считается пройденной, если все связанные с ней задачи выполнены.

Активные точки обводятся красной пунктирной линией. Точка считается активной, если с ней связана одна или более невыполненных задач (на рисунке – **Утверждение счета**).

2.14. Поле картинки

Элемент [Поле картинки](#) предназначен для размещения в форме различных картинок. Этот элемент также может содержать ссылку (см. описание в разделе «[Гиперссылка](#)»).

В некоторых случаях, когда изображение не помещается в поле элемента, при выборе элемента появляются линейки прокрутки, с помощью которых можно увидеть изображение полностью (если такая возможность была установлена в конфигурации). При наличии линейки прокрутки изображение можно двигать также с помощью клавиш управления курсором, [Home](#) (показать верхний левый угол), [End](#) (показать нижний правый угол), [PgUp](#) (перейти вверх на размер окна по вертикали, а при нажатой клавише [Alt](#) – по горизонтали), [PgDn](#) (перейти вниз на размер окна по вертикали, а при нажатой клавише [Alt](#) – по горизонтали).

Если мышь имеет колесо прокрутки, то вертикальное перемещение изображения выполняется прокруткой колеса от себя (вверх) и к себе (вниз). Если при прокрутке нажать клавишу [Shift](#), то выполняется горизонтальная прокрутка картинки.

Для элемента формы [Поле картинки](#) добавляется контекстное меню, состоящее из пунктов [Копировать](#), [Сохранить как](#), [Просмотреть картинку](#).

Чтобы скопировать изображение в буфер обмена, нужно выбрать пункт меню [Копировать](#). После этого его можно вставить в другое приложение.

Меню [Сохранить как](#) доступно для картинок, загруженных из файла. С его помощью можно сохранить картинку в файл на диске.

Если выбран пункт [Просмотреть картинку](#), то на экран выводится окно просмотра картинки, в котором есть кнопки увеличения и уменьшения масштаба, а также установки реального размера. Существует возможность изменять размеры окна. Для выбора нужного масштаба используются пункты меню [Увеличить](#), [Уменьшить](#) и [Оригинальный размер](#). Для изменения масштаба можно также использовать клавиши, расположенные на цифровой клавиатуре. Нажатие клавиши [+](#) (плюс) увеличивает картинку, клавиши [-](#) (минус) – уменьшает. Команда [Оригинальный размер](#) устанавливает реальный размер картинки.

Следует отметить, что для действий, выполняемых с помощью мыши, картинка должна быть «в фокусе», т. е. по ней следует предварительно щелкнуть мышью. Если щелкнуть одну из кнопок, то фокус будет на кнопке, и дальнейшие действия мышью (прокрутка и масштаб) не будут выполняться до тех пор, пока фокус не будет снова установлен на картинке.

Аналогичные действия по изменению масштаба можно выполнить с помощью мыши, у которой имеется колесо прокрутки, при нажатой клавише [Ctrl](#). Для увеличения масштаба необходимо крутить колесо от себя, для уменьшения – на себя. Изменение масштаба также можно выполнять с клавиатуры, используя клавиши [+](#) (плюс) и [-](#) (минус), расположенные справа от цифровой клавиатуры.

2.15. Поле табличного документа

Форма может содержать встроенный табличный документ (элемент формы [Поле табличного документа](#)). О работе с элементом формы [Поле табличного документа](#) см. в приложении 3, поставляемом в электронной версии документации.

Перемещение по таблице. Окно табличного документа в любой момент времени показывает только часть таблицы, расположенной в этом окне. Для выбора ячейки таблицы используется мышь. Выбранная ячейка становится активной. Для вывода на экран областей, не поместившихся в границах окна, используются линейки прокрутки.

СОВЕТ. Чтобы прокрутить документ по горизонтали, нужно нажать и удерживать клавишу [Shift](#), прокрутить колесо мыши.

Для удобства прокрутки большого документа можно нажать колесо мыши и выбрать курсором направление прокрутки. Скорость прокрутки зависит от расстояния между текущим положением курсора и тем местом, где было нажато колесо (отображается специальным образом).

Управление видом таблицы. При помощи пунктов подменю [Таблица – Вид](#) можно управлять отображением различных областей и компонентов табличного документа: заголовков строк и колонок, сеткой таблицы и т. д.

Кратко поясним назначение пунктов меню [Вид](#). Каждый пункт работает как переключатель: выбор (нажатие) вызывает или прекращает отображение соответствующей области.

--	--

Пункт меню	Действие
Зафиксировать таблицу	Фиксирует верхние строки и левые колонки табличного документа таким образом, чтобы при перелистывании табличного документа они постоянно присутствовали на экране
Отображать сетку	Включает и отключает показ сетки табличного документа – тонких линий, разделяющих ячейки таблицы
Отображать заголовки	Управляет показом заголовков строк и колонок табличного документа
Отображать группы	Включает и отключает показ групп – областей слева от заголовков строк и сверху от заголовков колонок
Редактирование	Включает и отключает запрет на любые изменения в табличном документе
Отображать примечания	Включает и отключает режим показа примечаний
Черно-белый просмотр	Отключает и включает показ цветов, заданных для табличного документа. При включении этого пункта (в меню напротив него появляется галочка) все цвета будут определяться установками операционной системы Microsoft Windows
Режим просмотра страниц	Отключает и включает показ минимального количества строк (определяется введенной информацией), области печати, разрывов страниц
Масштаб	Дискретно масштабирует изображение. При выборе этого пункта открывается подменю, в котором можно выбрать желаемый масштаб изображения табличного документа

СОВЕТ. Для масштабирования можно использовать мышь с колесом прокрутки. Нужно нажать клавишу **Ctrl** и, не отпуская ее, повернуть колесо. Движением колеса от себя масштаб увеличивается, к себе – уменьшается.

Фиксация таблицы (шапки и боковой части). Редактор табличных документов системы «1С:Предприятие» позволяет зафиксировать верхние строки и левые колонки табличного документа таким образом, чтобы при прокрутке табличного документа они постоянно присутствовали на экране.

Чтобы зафиксировать верхние строки табличного документа, следует выделить строку табличного документа целиком и выбрать пункт **Таблица – Вид – Зафиксировать таблицу**. Зафиксированы будут все строки, расположенные выше выделенной строки.

Чтобы зафиксировать колонки табличного документа, следует выделить колонку табличного документа целиком. Зафиксированы будут колонки, расположенные слева от выделенной колонки.

Чтобы зафиксировать одновременно строки и колонки табличного документа, следует выделить одну ячейку табличного документа. Зафиксированы будут строки и колонки, расположенные соответственно выше и левее выделенной ячейки.

Для отмены фиксации следует повторно выбрать пункт **Зафиксировать таблицу**.

Защита от редактирования. Для запрета редактирования табличного документа используется пункт **Таблица – Вид – Редактирование**, который должен быть отключен.

Если ячейка содержит текст, который полностью не помещается в видимой области, то при подведении указателя мыши текст показывается в виде подсказки. Если размер текста достаточно большой, чтобы отобразить его в виде подсказки, показывается только часть текста.

2.16. Поле форматированного документа

На форме может быть расположен элемент вида **Поле форматированного документа**, предназначенный для хранения текстового документа, включающего элементы форматирования: выделение цветом, изменение шрифта текста, гиперссылки, картинки и т. д.

Такое поле может использоваться, например, при создании сообщения электронной почты или для редактирования простейшего html-документа.

Если поле форматированного документа находится в режиме **Только просмотр**, то пользователь может переходить по ссылкам, которые содержит документ. Переход выполняется в новом окне браузера.

С помощью команд главного меню **Файл – Сохранить** и **Сохранить как** форматированный документ может быть сохранен как текстовый файл или html-документ.

Форматирование документа выполняется с помощью команд командной панели форматированного документа.

Кратко перечислим предоставляемые возможности форматирования.

Изменение шрифта. Шрифт документа можно изменить двумя способами:

- выбрать команду **Изменение Шрифта** и в диалоге **Выбор шрифта** выбрать шрифт, размер шрифта, стиль начертания (**Жирный**, **Подчеркнутый**, **Наклонный**, **Зачеркнутый**) и нажать кнопку **ОК**;
- выделить текст, шрифт которого нужно изменить, и с помощью соответствующих команд панели инструментов увеличить или уменьшить размер шрифта пошагово, установить стиль шрифта (полужирный, курсив или подчеркнутый).

Выбор цвета фона и текста. Система «1С:Предприятие» обеспечивает возможность выбрать цвет фона и цвет текста. Для этого нужно выделить текст, на панели инструментов выбрать команду **Изменение цвета фона** (чтобы изменить цвет фона текста) или **Изменение цвета текста** (чтобы изменить цвет текста). В открывшемся диалоге **Выбор цвета** выбрать необходимый цвет и нажать кнопку **ОК**.

Вставка гиперссылки. Текст форматированного документа может содержать ссылки. Нужно выделить объект (текст или картинку), выбрать пункт **Вставить гиперссылку**. На экран выводится диалог формирования. В поле **Ссылка** указать адрес гиперссылки и нажать кнопку **ОК**.

Выделенный объект форматируется в соответствии со стилем гиперссылки. Обычно это синий цвет текста с подчеркиванием.

Вставка картинки. В форматированный документ можно вставлять картинки. Для вставки картинки выбрать пункт **Вставить картинку** и выбрать файл в стандартном диалоге выбора файла.

Вставка символов. Для добавления в текст символов из списка используется команда **Вставить символ**.

Вставка списка. Чтобы начать маркированный список, используется команда **Маркированный список**; чтобы начать нумерованный – **Нумерованный список**. Затем нужно ввести текст.

Чтобы добавить следующий элемент списка, следует нажать клавишу **Enter**.

Для завершения списка нужно нажать клавишу **Enter** дважды или нажать клавишу **Backspace**, чтобы удалить последний маркер или цифру списка.

Настройка выравнивания. Текст документа может быть выровнен по левому краю, по правому краю или по ширине документа.

Нужно выделить текст, который необходимо выровнять, и нажать кнопку **Влево/Вправо/По центру/По ширине**.

Текст с отступами. Можно увеличить или уменьшить отступ от края поля для одного или нескольких абзацев текста.

Чтобы увеличить отступ, нужно выделить текст и нажать кнопку **Увеличить отступ**. Если необходимо его уменьшить, нажать **Уменьшить отступ**.

Междустрочный интервал. Чтобы настроить междустрочный интервал, нужно выделить текст и выбрать команду **Междустрочный интервал**. В открывшемся окне ввести размер интервала с клавиатуры или с помощью кнопок и нажать **ОК**.

2.17. Поле планировщика

Элемент формы **Поле планировщика** предназначен для отображения и редактирования различных данных планирования. В зависимости от конфигурации, данный элемент можно использовать для планирования задач, расписаний, событий, встреч и т.д.

В поле планировщика всегда содержится одно измерение – шкала времени.

12 августа 2014 г.	
08:00	
09:00	✉ 09:17 - 09:37 Проверка почты
10:00	⚠ 10:00 - 11:00 Важное совещание
11:00	
12:00	
13:00	☕ 13:00 - 14:00 Обед
14:00	
15:00	
16:00	
17:00	

Рис. 55. Поле планировщика с периодом отображения "День"

В зависимости от настроек конфигурации, в поле могут использоваться несколько измерений.

Период отображения поля планировщика также может зависеть от конфигурации.

Существует возможность с помощью команд главного меню, расположенных в подменю **Планировщик** изменять период, отображаемый в поле планировщика:

- **К началу** - перейти к началу периода отображения.
- **Назад** - перейти к предыдущему периоду отображения. Например, если выбран период отображения **День** и отображается день 12 августа 2014 г., будет выполнен переход к 11 августа 2014 г.
- **Сегодня** - перейти к текущему дню.
- **Вперед** - перейти к следующему периоду отображения.
- **К концу** - перейти к концу периода отображения.

2.17.1. Добавление элемента

В зависимости от настроек конфигурации, существует несколько способов добавления элементов на поле.

Чтобы создать элемент, достаточно выбрать нажатием правой клавиши мыши поле напротив требуемого интервала на оси времени.

Если установлено в настройках конфигурации, при выборе времени появится окно быстрого редактирования элемента.

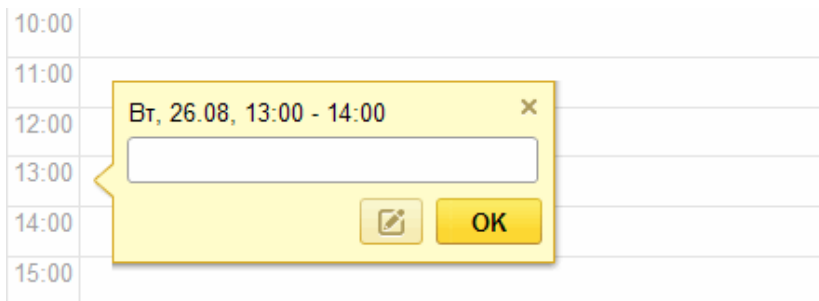


Рис. 56. Создание нового элемента планировщика

Чтобы сохранить информацию, нажмите кнопку **OK**.

2.17.2. Редактирование элемента

Чтобы просмотреть или изменить данные элемента, в контекстном меню или в окне быстрого редактирования следует нажать кнопку **Редактировать**. При этом откроется диалог редактирования элемента.

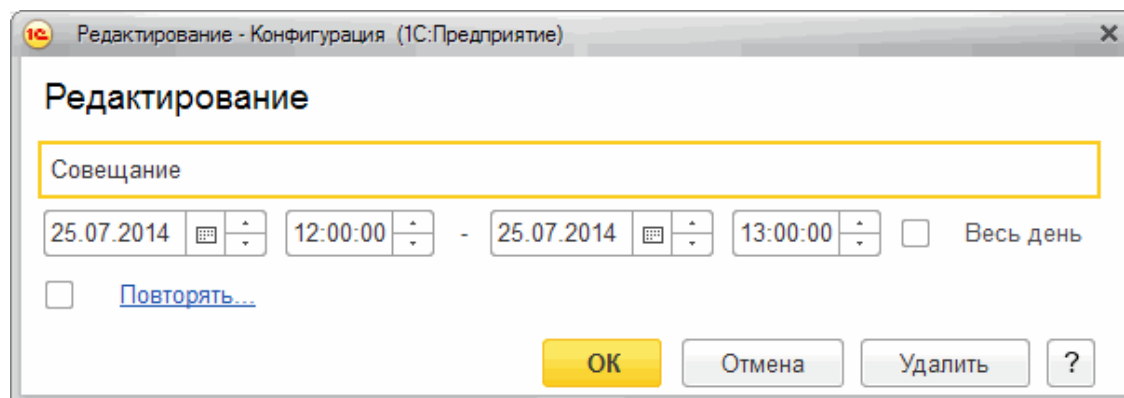


Рис. 57. Редактирование элемента планировщика

В диалоге устанавливается следующая информация об элементе планировщика:

- **Наименование** – наименование элемента планировщика;
- **Период проведения** – Если установлен флажок **Весь день**, время проведения события автоматически устанавливается **00:00 - 23:59** для выбранного дня;
- **Период повтора** -Чтобы установить период повтора события, нажмите ссылку **Повторять...** или установите флажок слева от нее. Расписание повторов события редактируется в отдельном окне.

Для завершения редактирования элемента планировщика нажмите **OK**.

Чтобы отменить внесенные изменения, нажмите **Отмена**.

Чтобы удалить событие, нажмите кнопку **Удалить**.

2.17.2.1. Перемещение элемента

Изменить интервал времени, занятый элементом, можно несколькими способами:

- Можно перетащить элемент на новые дату и время с помощью мыши.
- Изменить интервал времени в диалоге редактирования элемента.

2.17.2.2. Создание повторяющегося элемента

В планировщике существует возможность настроить повторение элемента с заданным интервалом, например,

Для этого установите флажок **Повторять...** или щелкните по ссылке в окне редактирования элемента планировщика.

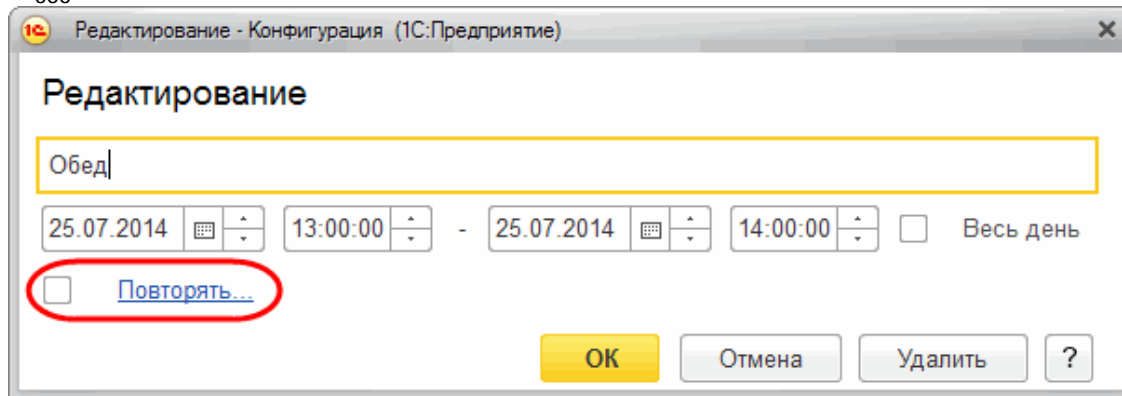


Рис. 58. Создание повторяющегося элемент

В открывшемся окне установите требуемые параметры.

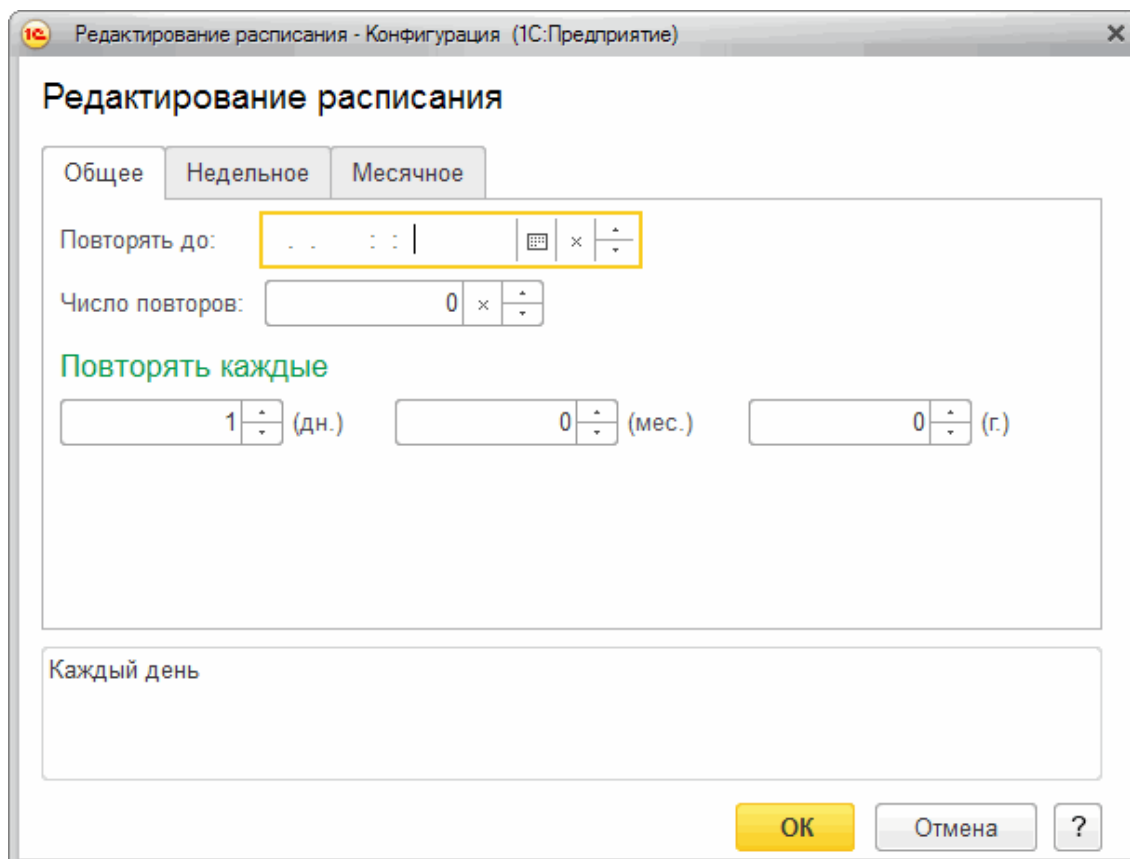


Рис. 59. Редактирование расписания элемента планировщика

На закладке [Общее](#) указывается дата завершения повторов (в поле [Повторять до:](#)), число повторов (поле [Число повторов](#)), интервал повтора в днях, месяцах, годах ([Повторять каждые](#))

Проверить правильность настройки можно в нижней части окна редактирования расписания.

На закладке [Недельное](#) указывается недельное расписание элемента:

Редактирование расписания - Конфигурация (1С:Предприятие)

Редактирование расписания

Общее Недельное **Месячное**

Повторять каждые: × (нед.)

По дням недели

Пн Вт Ср Чт Пт Сб Вс

Каждый день

OK Отмена ?

Рис. 60. Редактирование недельного расписания элемента планировщика

В поле **Повторять каждые** устанавливается интервал повтора элемента планировщика в неделях.

В группе **По дням недели** следует установить флажки по тем дням недели, в которых элемент будет повторяться.

На закладке **Месячное** указывается месячное расписание элемента планировщика.

Редактирование расписания - Конфигурация (1С:Предприятие)

Редактирование расписания *

Общее Недельное **Месячное**

Повторять в

× день месяца

× день недели в месяце

По месяцам

Январь Февраль Март Апрель

Май Июнь Июль Август

Сентябрь Октябрь Ноябрь Декабрь

1-го числа месяца

OK Отмена ?

Рис. 61. Редактирование месячного расписания

В группе **Повторять в** следует выбрать номер дня месяца и/или дня недели в месяце и указать порядок отсчета

(с начала или конца месяца/недели).

В группе [По месяцам](#) следует установить флажки по тем месяцам, в которых требуется повторение элемента.

В нижней части окна редактирования расписания представлены сводные данные установленного расписания повторов.

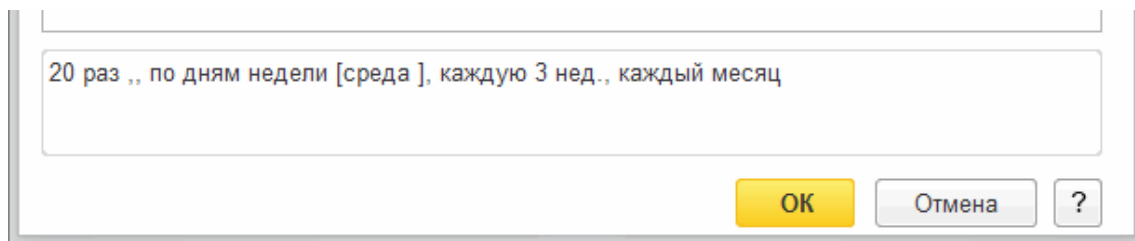


Рис. 62. Сводные данные расписания повторов

2.17.3. Удаление элемента

Данное действие доступно только для элементов планировщика, которые пользователь может редактировать. Чтобы удалить элемент, выберите его правой кнопкой мыши и в контекстном меню выберите команду [Удалить](#).

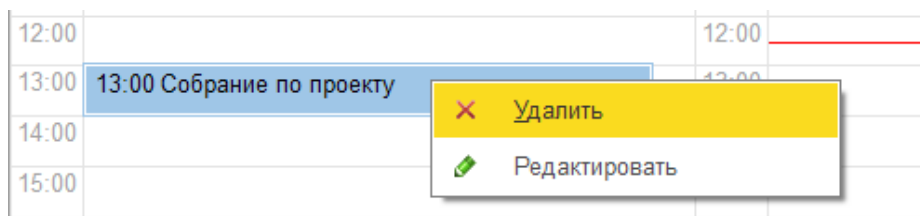


Рис. 63. Удаление элемента планировщика

2.18. Работа формы с сохранением настроек

Когда на форме расположено много реквизитов, то в конфигурации может быть предусмотрена возможность сохранять наборы установленных реквизитов и использовать их при необходимости. Для этого на форме располагаются команды [Сохранить параметры](#) и [Восстановить параметры](#).

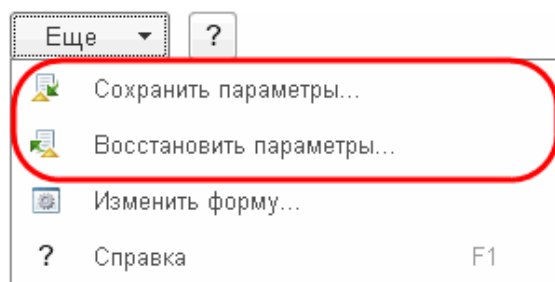


Рис. 64. Команды для работы с параметрами формы

Следует установить нужные значения в полях формы и выбрать пункт [Сохранить параметры...](#) В появившемся окне [Сохранение параметров](#) ввести имя для набора сохраняемых параметров и нажать кнопку [Сохранить](#).

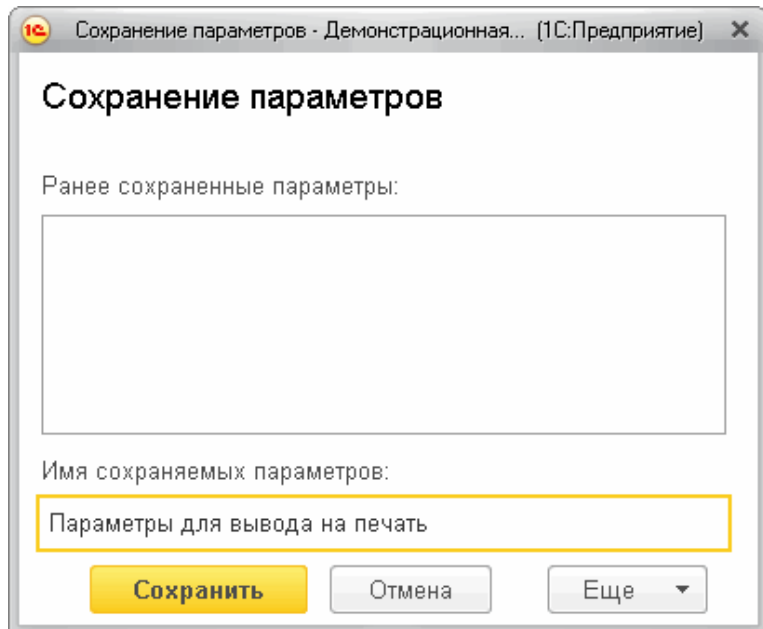


Рис. 65. Сохранение параметров формы

Для использования сохраненных ранее параметров формы служит пункт [Восстановить параметры](#). В появившемся окне нужно выбрать набор параметров и нажать кнопку [Выбрать](#).

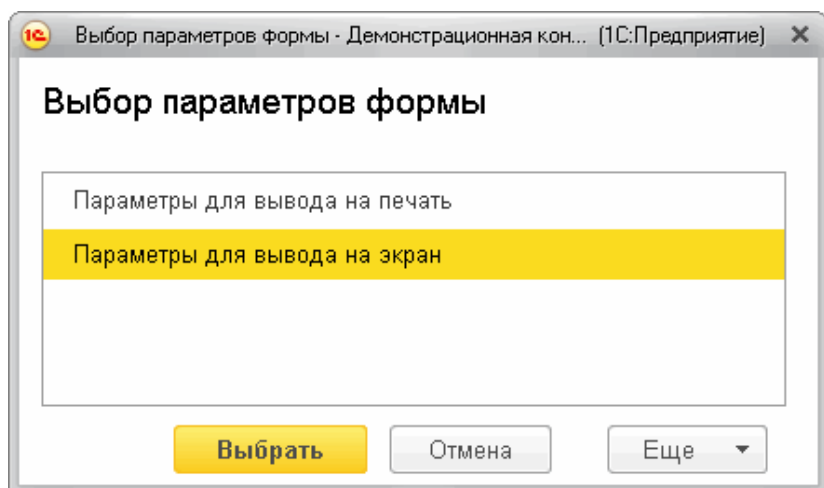


Рис. 66. Выбор сохраненных параметров формы

Параметры из выбранного набора будут использоваться в качестве параметров формы.

Глава 3. Списки

В системе «1С:Предприятие» для отображения информации чаще всего используются различные **списки**.

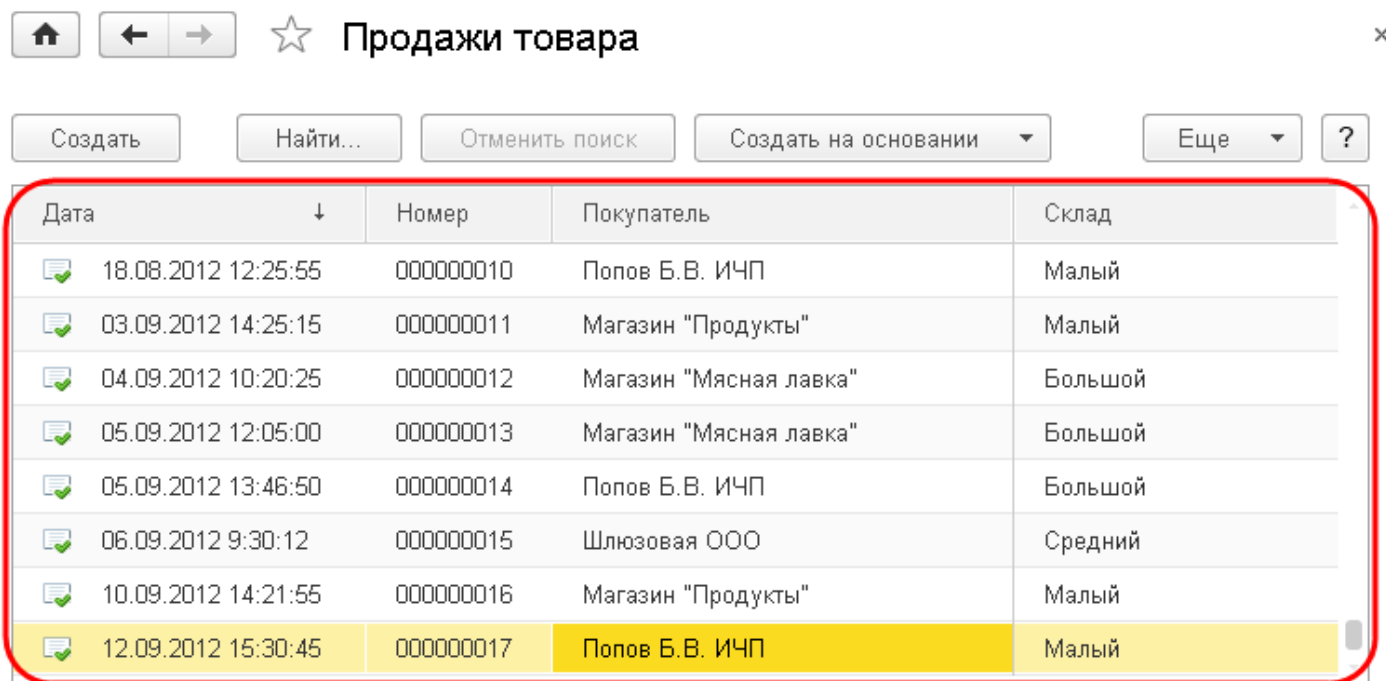


Рис. 67. Список документов «Продажи товара» и командная панель

Если это не запрещено в конкретной конфигурации (для конкретного пользователя), список может редактироваться в процессе работы – могут вводиться новые записи, редактироваться или удаляться существующие.

Работа со списками может осуществляться в двух формах: **форма списка** и **форма выбора элемента**.

Чтобы перейти к форме списка объектов, обычно следует выбрать соответствующую ссылку на панели навигации.

В форме выбора список открывается только при вводе значения реквизита формы нажатием кнопки выбора или клавиши **F4**. При этом в конфигурации могут быть определены различные формы для выбора элементов и групп.

Обычно в форме выбора можно установить курсор на требуемый элемент списка и по нажатию клавиши **Enter** выбрать текущий элемент как значение редактируемого реквизита, а также создать новый объект. В форме списка по нажатию клавиши **Enter** для текущего элемента открывается форма редактирования. Поведение программы для отработки нажатия клавиши **Enter** может быть изменено на этапе конфигурирования.

3.1. Просмотр списка

Для просмотра списка используются общие приемы работы с таблицами. Подробно эти приемы описаны в разделе «[Таблица](#)» [здесь](#).

Наименование	Код	Артикул	Поставщик	Вид
Колбаса	000000030	Kol67	Животноводство О...	Товар
Кроссовки	000000024	Kros001	Корнет ЗАО	Товар
Масло	000000021	ПРД-0002	Животноводство О...	Товар
Молоко	000000026	Mol34	Животноводство О...	Товар
Обувь	000000001			
Пинетки	000000025	Pin23	Пантера АО	Товар
Продукты	000000011			
Пылесосы	000000014			

Рис. 68. Форма списка «Товары»

Форма списка в общем случае представляет собой таблицу, в которой выводится список элементов. Набор колонок таблицы определяется конкретной конфигурацией.

Состав и взаимное расположение колонок, отображаемых в таблице, могут различаться. То есть конфигурация может предлагать несколько визуальных представлений одного и того же списка.

ПРИМЕЧАНИЕ. В ячейке таблицы текст, содержащий более 20 000 символов, будет отображен не полностью.

При нажатии левой кнопки мыши на колонке списка система отсортирует информацию в списке по значению колонки.

Пользователь может настроить состав отображаемых колонок с помощью пункта меню [Еще – Изменить форму...](#) Методика настройки форм приведена в разделе [«Настройка формы»](#) [здесь](#).

Если список содержит много элементов (строк) и колонок, то в форме отображается только определенная его часть, а внизу и справа появляются линейки прокрутки.

Как элементы, так и группы списка могут быть помечены к удалению, что может отображаться зачеркнутыми пиктограммами в крайней левой колонке списка.

Изменение ширины колонок в списках отслеживается. Размер сохраняется и восстанавливается при следующем открытии окна. Если список открывается в отдельном окне, то ширина колонок сохраняется вместе с размерами и положением окна.

3.1.1. Иерархические списки

Платформа «1С:Предприятие» предоставляет возможность работы с иерархическими списками с неограниченным числом уровней вложенности (ограничение может задаваться на этапе конфигурирования).

Иерархические списки состоят из элементов различных уровней, при этом элементы нижних уровней подчинены элементам верхних уровней. Существует два вида иерархии списков: иерархия групп и элементов и иерархия элементов. Вид устанавливается конфигурацией.

В списке с иерархией групп и элементов содержатся два вида элементов – группы и собственно элементы. Группа обозначает узел, в который входят другие (подчиненные) группы и элементы, а элемент является конкретным объектом.

Для списков с иерархией элементов любой из элементов может быть как узлом, так и отдельным объектом. Примером может служить список подразделений. Каждое подразделение может содержать в своем составе другие подразделения, но набор свойств у всех подразделений будет одинаков.

Иерархический список имеет следующие режимы отображения: иерархический список, список, дерево. Режимы переключаются с помощью пункта меню [Еще – Режим просмотра](#).

Выбранный режим просмотра отображается «нажатой пиктограммой» (такая пиктограмма имеет рамку напротив пункта меню).

Режим «Список». Если в качестве режима просмотра выбран [Список](#), то в форме показываются все элементы списка. При этом группы и элементы показываются неупорядоченно.

Наименование	↓	Код	Артикул	Поставщик	Вид
☰ Колбаса		000000030	Kol67	Животноводство О...	Товар
☰ Кроссовки		000000024	Kros001	Корнет ЗАО	Товар
☰ Масло		000000021	ПРД-0002	Животноводство О...	Товар
☰ Молоко		000000026	Mol34	Животноводство О...	Товар
📁 Обувь		000000001			
☰ Пинетки		000000025	Pin23	Пантера АО	Товар
📁 Продукты		000000011			
📁 Пылесосы		000000014			

Рис. 69. Режим просмотра «Список»

Режим «Иерархический список». Для списков с иерархией групп и элементов при включенном режиме [Иерархический список](#) показываются только элементы верхнего уровня, а после перехода в некоторую группу – только элементы, подчиненные этой группе.

Наименование	↓	Код
▶ 📁 Покупатели		000000002
▶ 📁 Поставщики		000000001

Наименование	↓	Код
▼ 📁 Покупатели		000000002
☰ Магазин "Бытовая техника"		000000014
☰ Магазин "Мясная лавка"		000000016
☰ Магазин "Обувь"		000000015
☰ Магазин "Продукты"		000000013
☰ Попов Б.В. ИЧП		000000007
☰ Шлюзовая ООО		000000004

Рис. 70. Список контрагентов в режиме «Иерархический список»

- [Группа скрыта](#) – строка соответствует группе элементов. Нажатием левой кнопки мыши на этом значке можно раскрыть группу и перейти к работе с элементами, входящими в группу;
- [Группа открыта](#) – строка содержит название группы, открытой для просмотра. Такая строка всегда помещается в верхней части таблицы, под названиями колонок списка. Нажатием левой кнопки мыши на пиктограмме в строке можно закрыть группу;
- [Элемент списка](#) – строка является элементом списка.

Раскрыть группу в списке можно одним из следующих способов:

- дважды щелкнуть мышью на строке списка с наименованием нужной группы;
- нажать левую кнопку мыши на символе **[+]** в строке с наименованием нужной группы;

- установить курсор на строке с наименованием нужной группы и нажать клавиши **Ctrl + Стрелка вниз**;
- установить курсор на строку с наименованием группы и нажать кнопку **+** на цифровой клавиатуре;
- установить курсор на строку с наименованием нужной группы и выбрать пункт **Еще – Перейти на уровень ниже**.

Для возврата на предыдущий уровень можно нажать клавиши **Ctrl + Стрелка вверх**, находясь в любой строке группы, или выбрать пункт **Еще – Перейти на уровень выше**.

Режим «Дерево». Если в качестве режима просмотра выбран режим **Дерево**, то элементы списка отображаются в виде дерева.









Наименование	↓	Код
⊖  Товары		
⊕  Обувь		000000001
⊕  Продукты		000000011
⊕  Услуги		000000036
⊖  Электротовары		000000012
⊕  Пылесосы		000000014
⊕  Телевизоры		000000039
⊕  Чайники		000000013

Рис. 71. Иерархический список в режиме «Дерево»

Для удобства работы дерево имеет возможность раскрываться и закрываться. Знак **+** (плюс) в узле ветви указывает, что ветвь можно раскрыть. При нажатии левой кнопки мыши на этом знаке ветвь откроет для просмотра следующий уровень, а знак **+** (плюс) изменится на **–** (минус). Свернуть ветвь дерева можно нажатием мыши на знаке **–** (минус).

Если дерево раскрыто, то для перехода к нужной группе используются клавиши **Стрелка Вверх** и **Стрелка Вниз**.

Для одновременного раскрытия и перехода на группу нижнего уровня следует использовать сочетание клавиш **Ctrl + Стрелка Вниз**. Для перехода к родительской группе – **Ctrl + Стрелка Вверх**.

Чтобы свернуть узел дерева и все подчиненные, используются клавиши **Shift + Alt + Num-**. Чтобы свернуть все узлы дерева, используются клавиши **Ctrl + Shift + Alt + Num-**. Чтобы развернуть узел дерева и все подчиненные, используются клавиши **Shift + Alt + Num+**. Чтобы развернуть все узлы дерева, используются клавиши **Ctrl + Shift + Alt + Num+**.

Также при просмотре в любом из режимов «дерево» или «список» развернуть узел дерева можно нажатием клавиши **Num+**, а свернуть – нажатием клавиши **Num-**.

Для быстрого перехода в начало или конец списка можно использовать соответственно клавиши **Home** и **End**.

Для списков с иерархией элементов в режиме **Дерево** просмотр осуществляется так же, как и для списков с иерархией групп и элементов. При этом пиктограммы строк для списков с иерархией элементов не различаются.

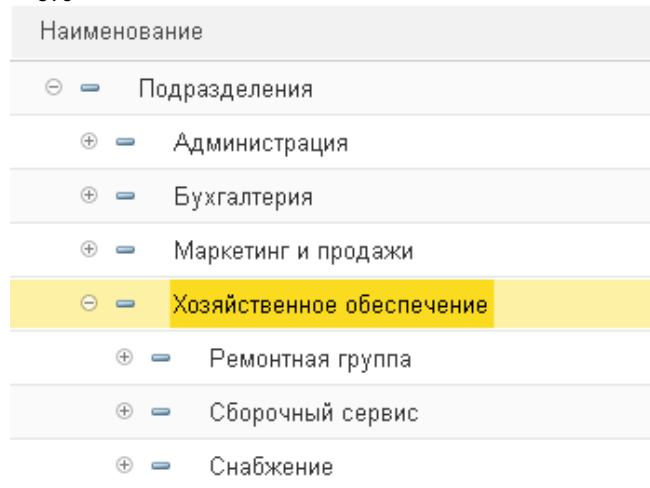


Рис. 72. Список с иерархией элементов в режиме просмотра «Дерево»

3.2. Создание элемента списка

Для создания элемента нужно нажать кнопку [Создать](#) формы списка или клавишу [Insert](#).

Если необходимо, чтобы элемент, создаваемый в иерархическом списке, сразу попал в определенную группу этого списка, то перед созданием следует перейти в эту группу (должен быть включен режим [Иерархический список](#) или [Дерево](#)).

ПРИМЕЧАНИЕ. Если для иерархического списка установлен режим просмотра [Список](#), то новый элемент будет создан в корневой группе.

Если в форме элемента предусмотрен реквизит [Группа](#), то независимо от режима и текущей группы для нового элемента можно указать принадлежность к любой другой группе.

Редактирование данных при вводе нового элемента списка осуществляется в выводимой на экран форме редактирования (окне) элемента.

Внешний вид окна, его свойства, поведение управляющих элементов определяются конфигурацией.

При редактировании реквизитов элемента списка используются общие приемы работы с формой. Подробно эти приемы описаны в главе [«Общие принципы работы с формами»](#) [здесь](#).

При работе с формой создания элемента списка могут проверяться корректность и полнота введенных данных. В случае неправильного ввода могут появляться соответствующие сообщения. После заполнения реквизитов для записи элемента в список нужно нажать кнопку [Записать и закрыть](#) или подобную ей кнопку, предназначенную для сохранения результатов редактирования, или закрыть форму кнопкой закрытия окна. После того как на экран будет выдан запрос о сохранении изменений, следует ответить [Да](#) для записи в список нового элемента или [Нет](#) – для отказа от записи.

Для отказа от ввода в список нового элемента следует нажать клавишу [Esc](#).

СОВЕТ. Для создания элемента списка используется команда группы [Создать](#) панели действий.

3.2.1. Создание группы списка

Для создания новой группы нужно выбрать пункт меню [Еще – Создать группу](#) формы списка.

Если необходимо, чтобы новая группа, вводимая в список, сразу попала в определенную группу списка, то перед вводом следует перейти в эту группу (должен быть включен режим [Иерархический список](#) или [Дерево](#)).

Если в форме ввода группы предусмотрен реквизит [Группа](#), то для новой группы можно указать принадлежность к любой другой группе независимо от режима и текущей группы.

Также как и при создании элемента списка, при создании новой группы редактирование реквизитов группы осуществляется в выводимой на экран форме группы.

Система не проводит различий между кодами элементов списка и групп элементов, т. е. коды группы и элемента

не могут совпадать.

В остальном приемы работы по созданию и редактированию группы списка совпадают с приемами работы по вводу элемента.

3.2.2. Копирование элемента списка

Новый элемент или группа могут быть добавлены в список копированием уже существующего элемента или группы. Для этого нужно поместить курсор на строку с элементом или группой, которые будут использоваться в качестве образца, и выбрать пункт [Еще – Скопировать](#).

На экране отобразится форма создания нового элемента, в котором все реквизиты будут скопированы из реквизитов элемента-образца. Если для списка не задано автоматическое присвоение кодов, код нового элемента копируется из кода элемента-образца.

Корректировка реквизитов производится по общим правилам, изложенным в разделе [«Редактирование элемента списка»](#) [здесь](#).

3.3. Редактирование элемента списка

Для редактирования элемента необходимо установить курсор в любую ячейку строки, реквизиты которой требуется отредактировать, нажать клавишу [Enter](#), или дважды щелкнуть мышью в любом месте этой строки, или нажать клавишу [F2](#). На экран будет вызвана форма редактирования элемента списка, в котором следует отредактировать требуемые реквизиты.

Когда открыта форма элемента списка (или несколько форм), форма списка по-прежнему остается доступна для работы в окне приложения: можно вводить в нее новые элементы и группы, удалять их, переносить элементы из группы в группу, то есть использовать все возможности по работе со списками, предоставляемые программой.

Если в списке редактирование запрещено, то информация в открываемой форме будет недоступна для редактирования.

3.3.1. Реорганизация списка

При работе с иерархическими списками может возникать необходимость реорганизации элементов списка, то есть переноса существующих элементов или групп элементов из одной группы в другую.

Для переноса элемента или группы в другую группу списка нужно выполнить следующие действия:

1. В списке выбрать строку с тем элементом или группой, которую необходимо перенести в другое место списка.
2. Выбрать пункт [Еще – Переместить в группу](#).
3. В открывшемся диалоге выбрать группу, куда предполагается перенести выбранный элемент.

СОВЕТ. Также элементы списка можно перемещать между группами с помощью мыши. При этом следует учитывать, что в иерархическом списке вручную перетащить элемент в родительскую группу нельзя.

Если в конфигурации задана поддержка уникальности кодов элементов списка, то при переносе код переносимого элемента не должен совпадать с кодами элементов, уже существующих в группе. Если коды совпадут, будет выдано сообщение: [Значение поля <Имя поля> не уникально](#). При этом необходимо будет исправить код у переносимого элемента.

3.3.2. Выбор значения из списка

Список может быть открыт для выбора элемента. В этом варианте список открывается нажатием кнопки выбора или клавиши [F4](#) при вводе реквизита из списка.

Когда список открыт для выбора значения, окно списка может иметь несколько другой вид, чем когда он просто открыт, – иной состав и расположение колонок списка, отсутствие или иное положение дерева групп и так далее. В процессе настройки программы для различных режимов работы со списком можно создать различные формы. Как правило, форма выбора содержит минимум необходимой информации, зачастую – только колонки наименования и кода.

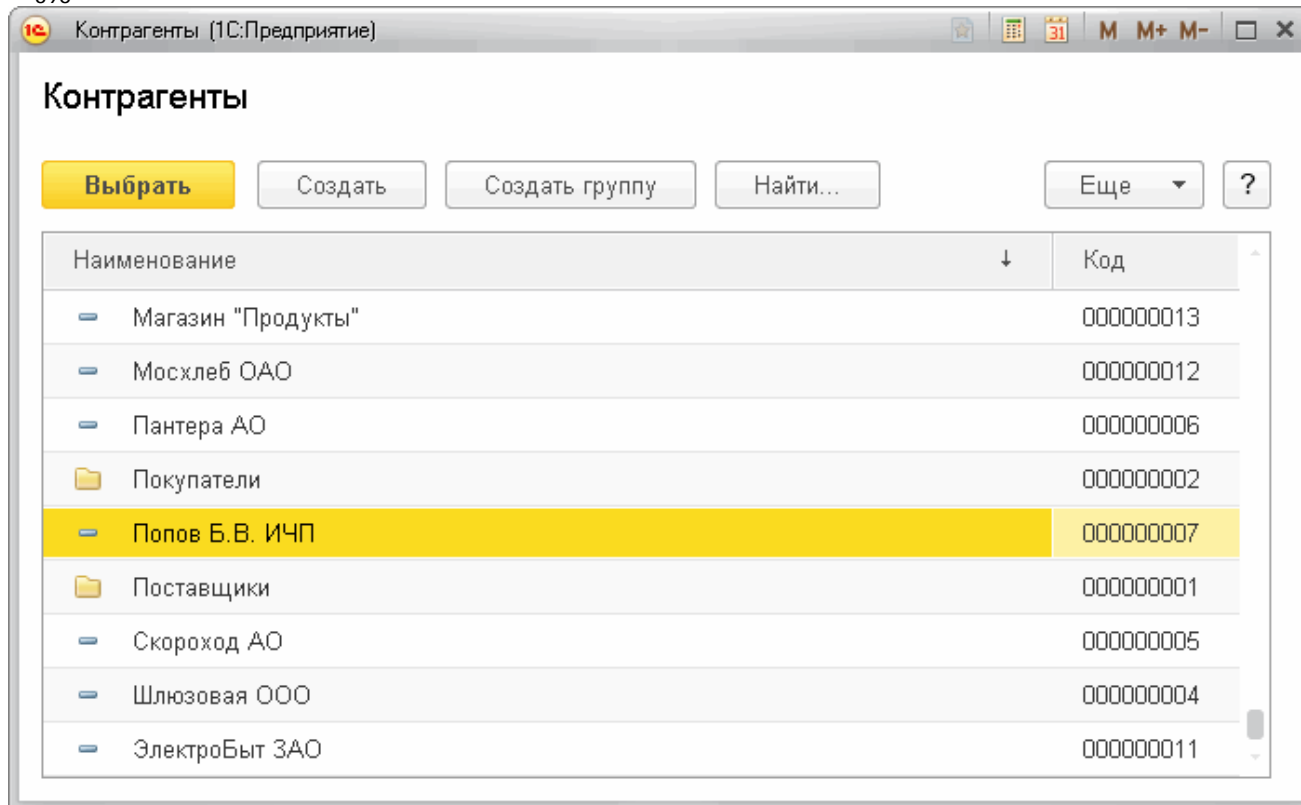


Рис. 73. Выбор значения реквизита из списка

В форме для выбора элемента также можно выполнять те же действия, что и в форме списка: добавлять в список новые элементы, редактировать дополнительные реквизиты существующих элементов списка, переносить элементы в другие группы (если в окне есть дерево групп) и тому подобное.

Например, если в списке организаций нужной организации нет, ее можно тут же ввести, а затем выбрать для использования в качестве реквизита документа. А для существующей организации можно исправить какие-либо реквизиты (если это доступно конкретному пользователю).

ПРИМЕЧАНИЕ. Следует учитывать принципиальное отличие поведения клавиши **Enter** в формах. В форме для выбора нажатие клавиши **Enter** приводит к выбору нужного элемента и закрытию (если иного не предусмотрено в конфигурации), а в формах списка – к открытию формы для просмотра и редактирования. Для редактирования элемента списка в форме для выбора используется команда меню **Еще** или клавиша **F2**.

Информация в форме выбора синхронизирована с информацией формы списка, и все изменения, которые будут производиться в одном окне, будут отражаться в другом.

Для выбора элемента списка необходимо установить курсор в строку с нужным элементом и нажать клавишу **Enter** или кнопку **Выбрать** командной панели (если она предусмотрена в конфигурации).

Для иерархических списков в формах элементов можно указывать группу, к которой принадлежит данный элемент (если это предусмотрено в конфигурации).

Для некоторых реквизитов допускается выбор не только элементов, но и групп. Для быстрого выбора группы достаточно использовать стандартный прием выбора при нажатой клавише **Shift**.

СОВЕТ. На самом деле для выбора данных из списка необязательно пользоваться формой выбора. Достаточно ввести в поле ввода код или начало наименования необходимого элемента. Особенно удобно использовать код при вводе данных в тех случаях, когда коды постоянно используются и легко запоминаются (например, номер подразделения) или присутствуют в первичном документе (например, артикул или штрихкод товара).

3.3.2.1. Предопределенные элементы списка

Список может иметь предопределенные элементы (если это определено в конфигурации). Такие элементы введены на этапе конфигурирования и не могут быть удалены в режиме **1С:Предприятие**. Допускается только редактирование реквизитов таких элементов.

3.3.3. Поиск в списках

Если на форме присутствует поле строки поиска, то чтобы начать поиск, начните набирать текст. При этом необязательно переключаться в поле поиска. В списке отобразятся результаты поиска, совпадения будут выделены цветом. Перейти на строку поиска можно и с помощью сочетания клавиш **Ctrl+F**.

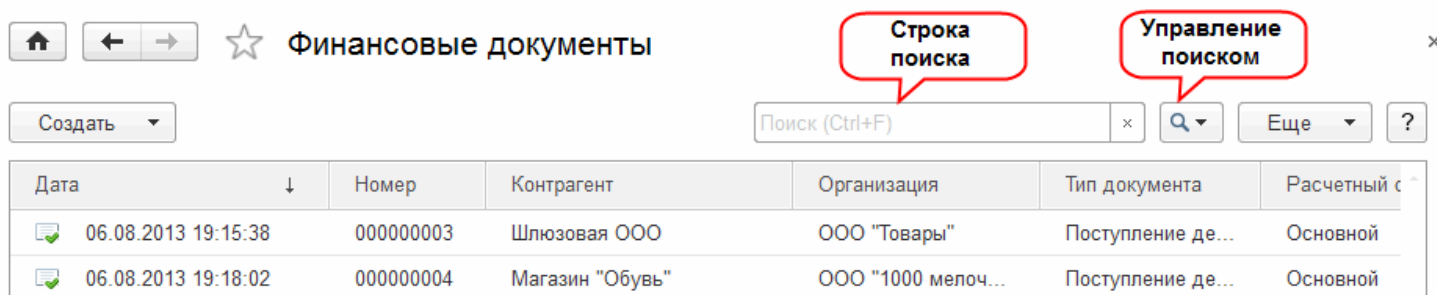


Рис. 74. Строка поиска на форме списка

Поиск будет выполнен по всем видимым колонкам списка одновременно, по нескольким значениям. Например, как показано на рисунке ниже.

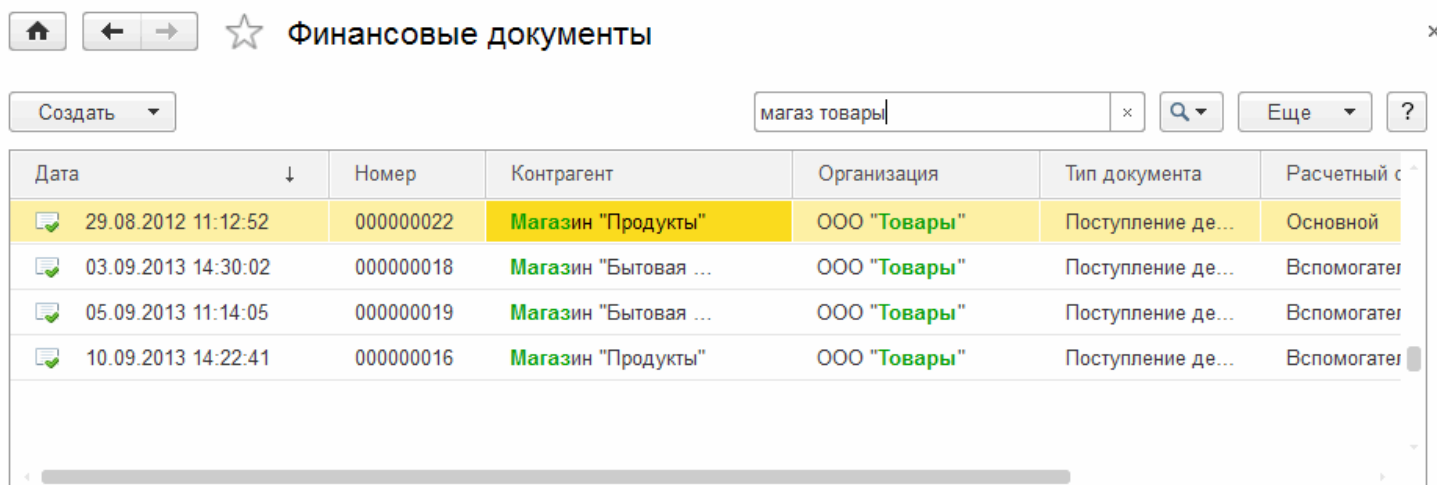
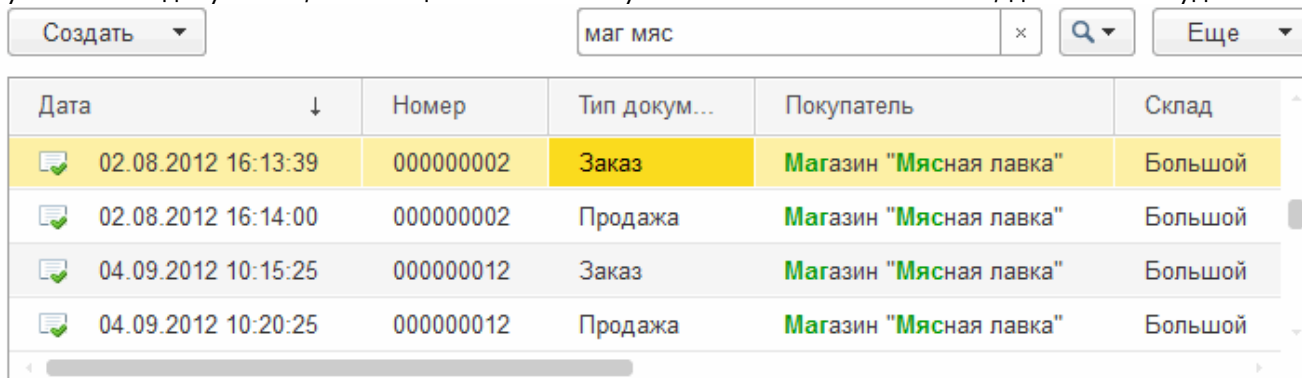


Рис. 75. Поиск в списке

При этом строка выражения поиска разбивается на части. В результат попадут строки списка, удовлетворяющие каждому условию:

- Строки ищутся по вхождению.

Если для списка включен полнотекстовый поиск, то поиск будет выполняться только по началу слов. Например, если нужно найти документы, относящиеся к магазину с названием **Мясная лавка**, достаточно будет ввести **маг**



МЯС.

Рис. 76. Поиск строки в списке

Если для списка не включен полнотекстовый поиск, то поиск будет выполняться в любом месте строки. И тогда, например, при поиске по строке **маг** будут найдены не только магазины, но и универмаги:

Создать ×

Дата	↑	Номер	Тип документа	Покупатель	Склад
04.02.2015 14:01:00		000000018	Продажа	Универмаг "Столичный"	Малый
13.09.2012 15:01:22		000000020	Заказ	Магазин "Продукты"	Средний
13.09.2012 14:39:08		000000019	Заказ	Магазин "Обувь"	Малый
13.09.2012 10:15:00		000000018	Заказ	Магазин "Продукты"	Малый
10.09.2012 14:21:55		000000016	Продажа	Магазин "Продукты"	Малый

Рис. 77. Поиск в списке без полнотекстового поиска

При поиске объекта по номеру можно вводить номер как с ведущими нулями, так и без указания ведущих нулей. Например, если нужно найти [Заказ](#) с номером [000000016](#), достаточно ввести [16](#):

🏠 ⬅️ ➡️ ☆ Заказы товаров ×

Создать ×

Дата	↓	Номер	Покупатель	Склад
10.09.2012 14:21:40		000000016	Магазин "Продукты"	Малый

Рис. 78. Поиск документа Заказ № 000000016

При этом, если для списка используется полнотекстовый поиск и ведущие нули введены не полностью (например, [016](#)), ничего найдено не будет.

При поиске даты, ее следует вводить целиком (в формате ДД.ММ.ГГГГ). Поиск осуществляется за весь день. Например, найдем документы продаж на [02.08.2012](#):

🏠 ⬅️ ➡️ ☆ Документы продаж ×

Создать ×

Дата	↑	Номер	Тип документа	Покупатель
02.08.2012 19:16:02		000000004	Заказ	Магазин "Обувь"
02.08.2012 18:15:18		000000003	Заказ	Шлюзовая ООО
02.08.2012 16:14:00		000000002	Продажа	Магазин "Мясная лав..."
02.08.2012 16:13:39		000000002	Заказ	Магазин "Мясная лав..."

Рис. 79. Поиск даты в списке

Поиск чисел выполняется по точному равенству. Например, найдем документы на сумму 10500 рублей:

Создать

Дата	Номер	Тип документа	Контрагент	Сумма документа
10.09.2012 ...	000000016	Поступление денег	Магазин "Продукты"	10 500,00

Рис. 80. Поиск числа в списке

Для значений полей флажка, при неиспользуемом полнотекстовом поиске, поиск выполняется по значению **Да** или **Нет**. Например, если нужно найти неиспользуемый склад, у которого флажок **Не использовать** установлен, в строке поиска введем **Да**. Если полнотекстовый поиск используется, то значения флажка в поиске не участвуют.

Создать

Наименование	Код	Не использовать
Строящийся склад	000000005	<input checked="" type="checkbox"/>

Рис. 81. Поиск по значению флажка

Для ссылок на объекты, поиск выполняется по тем полям, из которых состоит "значимая" часть ссылки (Например, в регистре перемещений товаров поиск будет выполняться по коду и/или наименованию товара. А если в списке движений документов нужно найти **Поступление товара 000000009 от 25.06.2012** - поиск будет выполняться по номеру и дате документа - см. рисунок ниже).

Период	Регистратор	Номер стр...	Товар
25.06.2012 8:52:32	Поступление товара 000000009 ...	1	Тапочки
25.06.2012 8:52:32	Поступление товара 000000009 ...	2	Сапоги
25.06.2012 8:52:32	Поступление товара 000000009 ...	3	Ботинки

Рис. 82. Поиск по ссылке

Найденные фрагменты строк в таблице выделяются.

Чтобы отменить поиск, следует нажать кнопку [Очистить](#) в поле строки поиска или нажать клавишу [Esc](#).

Чтобы уточнить критерии поиска, следует выбрать команду [Управление поиском - Расширенный поиск](#) или нажать [Alt + F](#).

Рис. 83. Форма поиска

Если диалог был открыт по сочетанию клавиш, то значение текущей ячейки будет установлено в строке [Что искать](#) и будет установлен поиск [По точному совпадению](#).

Если диалог был открыт при начале ввода поисковой строки в списке, способ поиска будет выбран как [По части строки](#), а набранный текст будет установлен в поле [Что искать](#).

В поле [Где искать](#) следует ввести искомое значение или выбрать колонку из списка выбора, а в поле [Что искать](#) установить значение поиска или выбрать его из списка выбора.

Для строк можно использовать один из способов поиска: [По началу строки](#), [По части строки](#), [По точному совпадению](#).

Числовые и логические значения ищутся по точному соответствию (равенству).

Если в динамическом списке отображаются иерархические данные, можно искать данные с учетом иерархии – во всем списке или в текущей группе.

Для поиска нажмите кнопку [Найти](#). Поиск работает как фильтр – список показывает только те строки данных, которые соответствуют указанным условиям поиска (колонка + значение).

При поиске в иерархическом списке показываются элементы, которые соответствуют указанному поисковому выражению в виде простого (неиерархического) списка. Отмена поиска возвращает список обратно в иерархический режим просмотра.

Если список иерархический, то в диалоге отображается наименование текущей группы и флажки:

- [Искать только в текущей группе](#), - если флажок установлен, поиск будет выполняться только в текущей группе и подчиненных(вложенных) группах.
- [Исключить подчиненные группы](#) - флажок доступен, только если флажок [Искать только в текущей группе](#) установлен. Если флажок установлен, поиск во вложенных группах выполняться не будет.

Если ни одной строки не найдено, то список будет пуст.

Создать

Поиск (Ctrl+F)

Еще

Контрагент: Магазин "Продукты"

Дата	Номер	Контрагент	Организация	Тип документа	Расчетный счет
29.08.2012 11:12:52	000000022	Магазин "Продукты"	ООО "Товары"	Поступление де...	Основной
03.09.2012 14:30:00	000000011	Магазин "Продукты"	ООО "Все для до...	Поступление де...	Основной
10.08.2013 12:40:00	000000008	Магазин "Продукты"	ООО "Все для до...	Поступление де...	Основной
10.09.2013 14:22:41	000000016	Магазин "Продукты"	ООО "Товары"	Поступление де...	Вспомогател

Рис. 84. Уточненный поиск

В списке найденных объектов можно уточнить параметры поиска.

Создать

Поиск (Ctrl+F)

Еще

Организация: ООО "Товары"

Контрагент: Магазин "Продукты"

Дата	Номер	Контрагент	Организация	Тип документа	Расчетный счет
29.08.2012 11:12:52	000000022	Магазин "Продукты"	ООО "Товары"	Поступление де...	Основной
10.09.2013 14:22:41	000000016	Магазин "Продукты"	ООО "Товары"	Поступление де...	Вспомогател

Рис. 85. Два поисковых выражения

Чтобы изменить значение поиска, нужно нажать ссылку <Где искать>: <Что искать> в элементе поиска.

В подменю [Управление поиском](#) сохраняются введенные ранее значения поиска.

Установить период...

Найти: Контрагент - Магазин "Продукты" Ctrl+Alt+F

Расширенный поиск Alt+F

Контрагент - Магазин "Продукты"

Организация - ООО "Товары"

Контрагент - Магазин "Мясная лавк..."

Отменить поиск Ctrl+Q

Рис. 86. История поиска

Чтобы быстро найти все строки списка, содержащие то же значение, что и в выделенной ячейке, в контекстном

меню списка выберите команду Найти: <Имя колонки> - <Текущее значение>. Чтобы отменить поиск по значению, следует нажать кнопку X.

3.4. Настройка списка

Настройка порядка следования колонок в списке, их видимости, поведения списка при открытии и управления обновлением данными производится в диалоге [Настройка формы](#). Диалог открывается командой [Еще – Изменить форму](#). Подробнее о настройке формы описано в разделе «[Настройка формы](#)» [здесь](#).

Настройка отображения данных, сортировки строк списка, а также группировки списка в таблице и оформления списка производится в диалоге [Настройка списка](#). Диалог открывается командой [Еще – Настроить список](#).

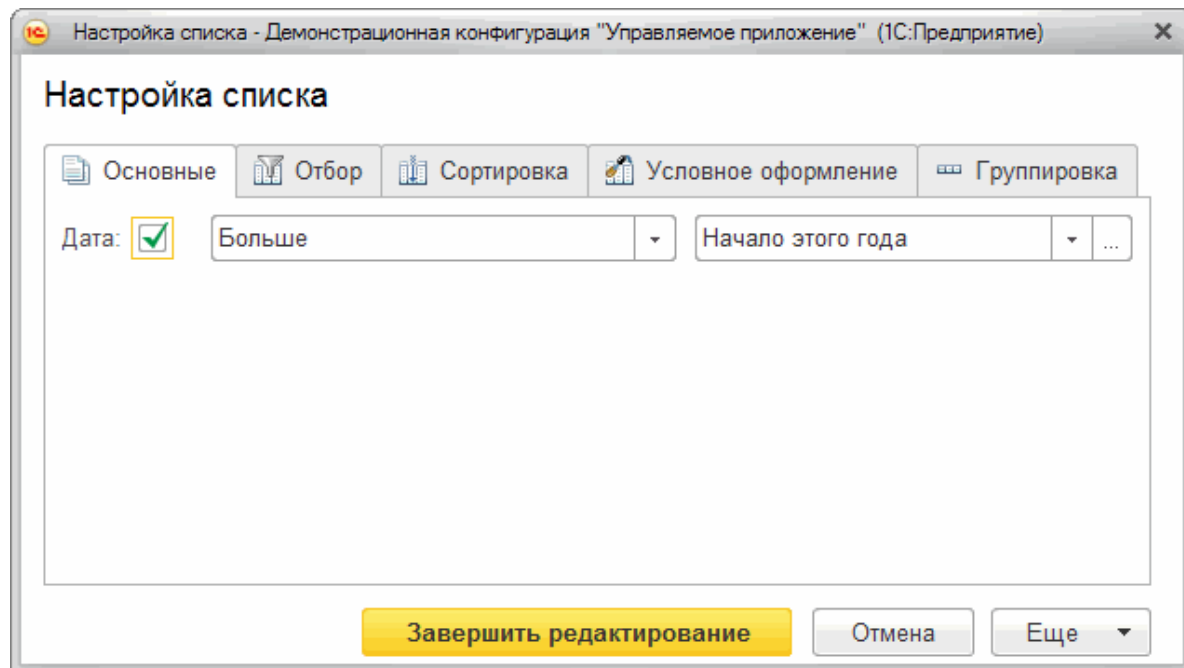


Рис. 87. Форма настройки списка

В диалоге предоставляется возможность настроить:

- отбор данных в списке;
- поля, по которым необходимо проводить упорядочивание;
- условное оформление списка;
- поля, по которым необходимо группировать данные;
- другие настройки, возможность изменения которых установлена разработчиком.

Все настройки распределены на закладках: [Основные](#), [Отбор](#), [Сортировка](#), [Условное оформление](#) и [Группировка](#). Чтобы установить значение настройки, следует установить значение настройки на соответствующей закладке.

Элемент настройки применяется к списку, если напротив него установлен флажок использования. Для элементов можно настраивать порядок применения в границах одной настройки.

Настройка отображения данных в списке производится аналогично настройке варианта отчета в системе компоновки данных.

Вернуть настройки списка по умолчанию можно командой [Еще – Установить стандартные настройки](#).

Для того чтобы определить состав настроек для изменения в основной форме списка, нужно выбрать [Еще – Изменить состав настроек](#) и в открывшемся диалоге добавить или удалить необходимые настройки.

Чтобы принять настройки, следует нажать кнопку [Завершить редактирование](#).

Установленные значения настроек сохраняются между сеансами работы.

Чтобы сохранить установленный набор настроек в форме списка, нужно выбрать [Еще – Сохранить настройки...](#) В открывшемся диалоге ввести название настройки и нажать кнопку [Сохранить](#).

Для использования созданных ранее настроек списков нужно выбрать [Еще – Выбрать настройки](#). В открывшемся диалоге выбрать требуемую настройку из списка ранее сохраненных настроек и нажать [Выбрать](#).

3.4.1. Отбор

Чтобы установить отбор в списке, следует в диалоге настройки списка перейти на закладку [Отбор](#) и добавить элементы отбора.

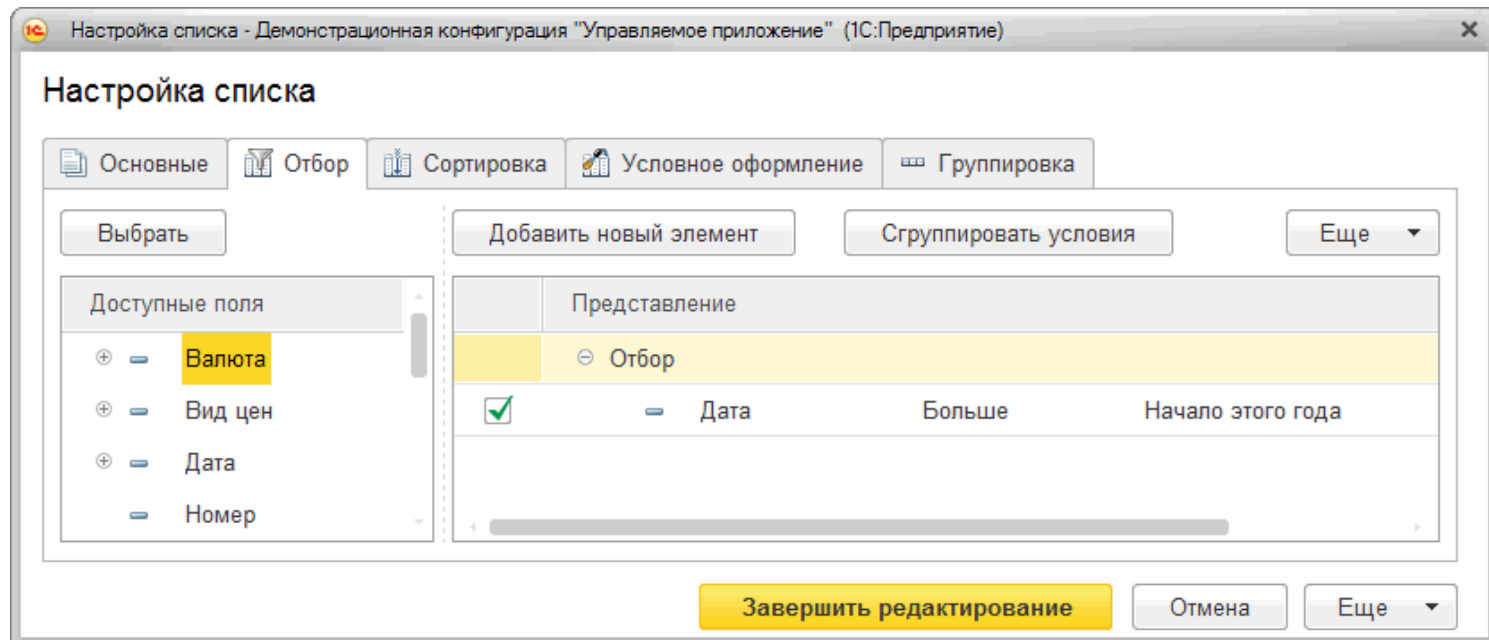


Рис. 88. Настройка отбора в списке

Для добавления элемента отбора следует нажать кнопку [Добавить новый элемент](#) или перетащить поле из списка [Доступные поля](#). Выбрать [Вид сравнения](#) в списке, всплывающем при нажатии кнопки [Выбрать](#) в колонке [Вид сравнения](#).

При необходимости можно изменить порядок, в котором условия будут применяться к списку, с помощью кнопок командной панели [Переместить вверх](#) и [Переместить вниз](#).

Чтобы применить настройку отбора в списке, следует нажать кнопку [Завершить редактирование](#).

Подробнее о работе с отбором и о видах сравнения см. раздел «Отбор» [здесь](#).

3.4.2. Сортировка

Чтобы установить сортировку в списке, следует перейти на закладку [Сортировка](#) и выбрать значение сортировки или установить флажок использования напротив существующего элемента сортировки.

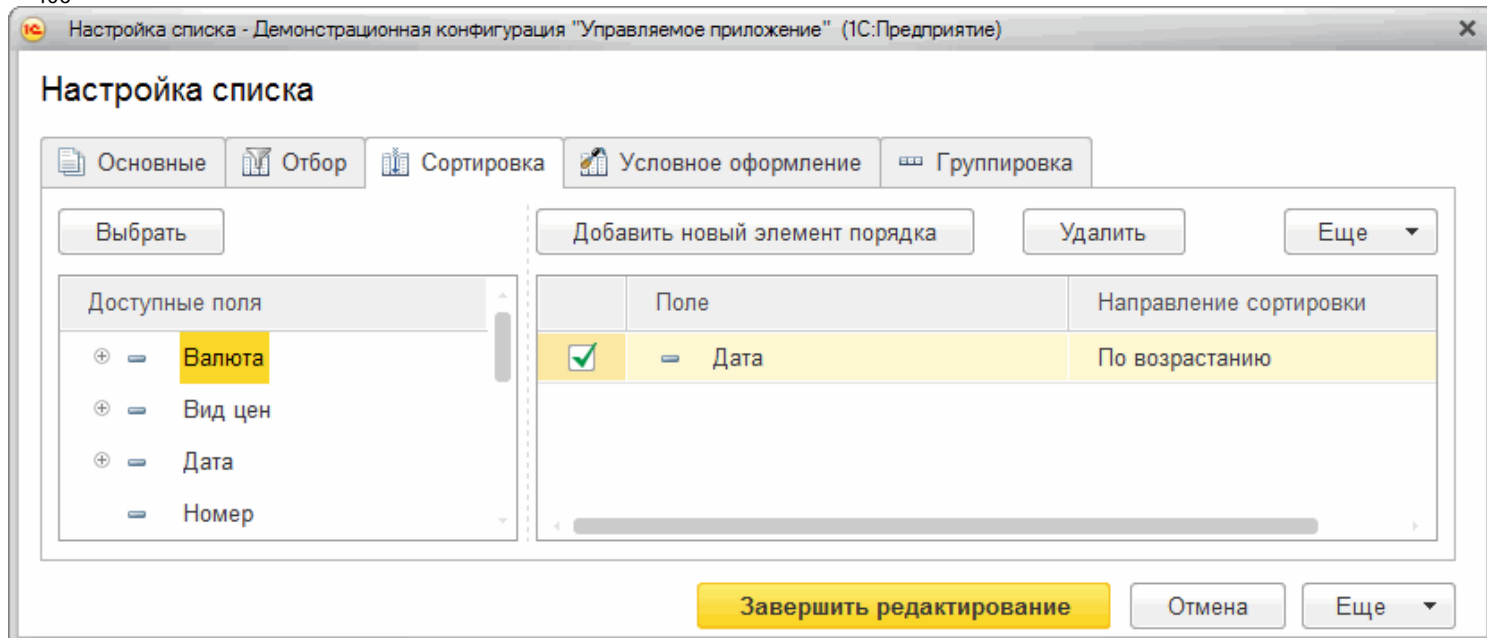


Рис. 89. Настройка сортировки списка

Для добавления элемента сортировки следует нажать [Добавить новый элемент порядка](#) или перетащить поле из списка [Доступные поля](#) и выбрать [Направление сортировки \(По возрастанию/По убыванию\)](#).

Если необходимо, установить порядок элементов сортировки кнопками [Переместить вверх](#) и [Переместить вниз](#).

Если полей сортировки несколько, то в списке данные будут отсортированы вначале по первому полю, потом по второму и т. д.

Чтобы применить настройку сортировки в списке, нажать кнопку [Завершить редактирование](#).

Подробнее о работе с сортировкой см. раздел «Сортировка» [здесь](#).

ПРИМЕЧАНИЕ. Сортировка списка по полям, содержащим строки неограниченной длины, не разрешается.

3.4.3. Группировка

Чтобы сгруппировать записи в списке, следует в диалоге настройки перейти на закладку [Группировка](#) и на форме добавить поле, по которому необходимо провести группировку. Если было выбрано несколько полей, то записи в списке будут последовательно сгруппированы в порядке, установленном для полей группировки.

Чтобы применить настройку группировки, следует нажать кнопку [Завершить редактирование](#).

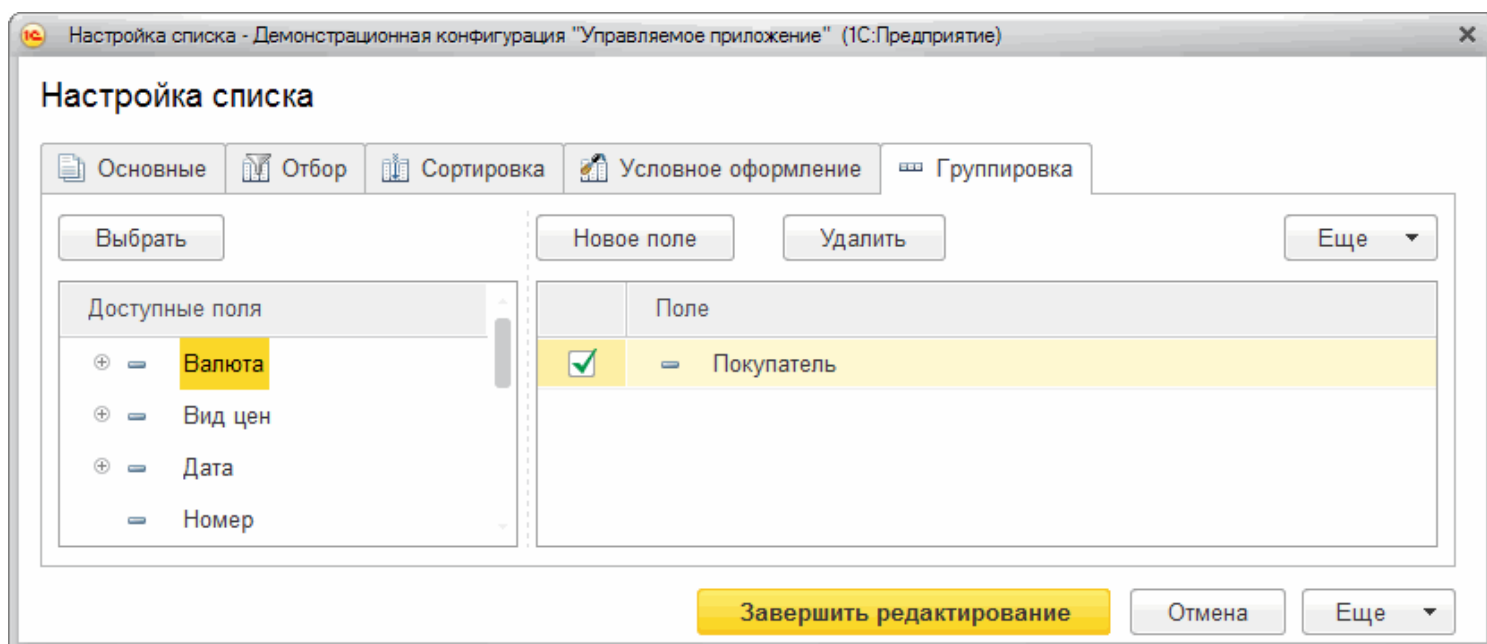


Рис. 90. Настройка группировки в списке

ПРИМЕЧАНИЕ. Группировка списка по полям, содержащим строки неограниченной длины, не разрешается.

3.4.4. Условное оформление

Для настройки условного оформления следует в диалоге настройки перейти на закладку [Условное оформление](#) и в табличном поле добавить необходимые элементы.

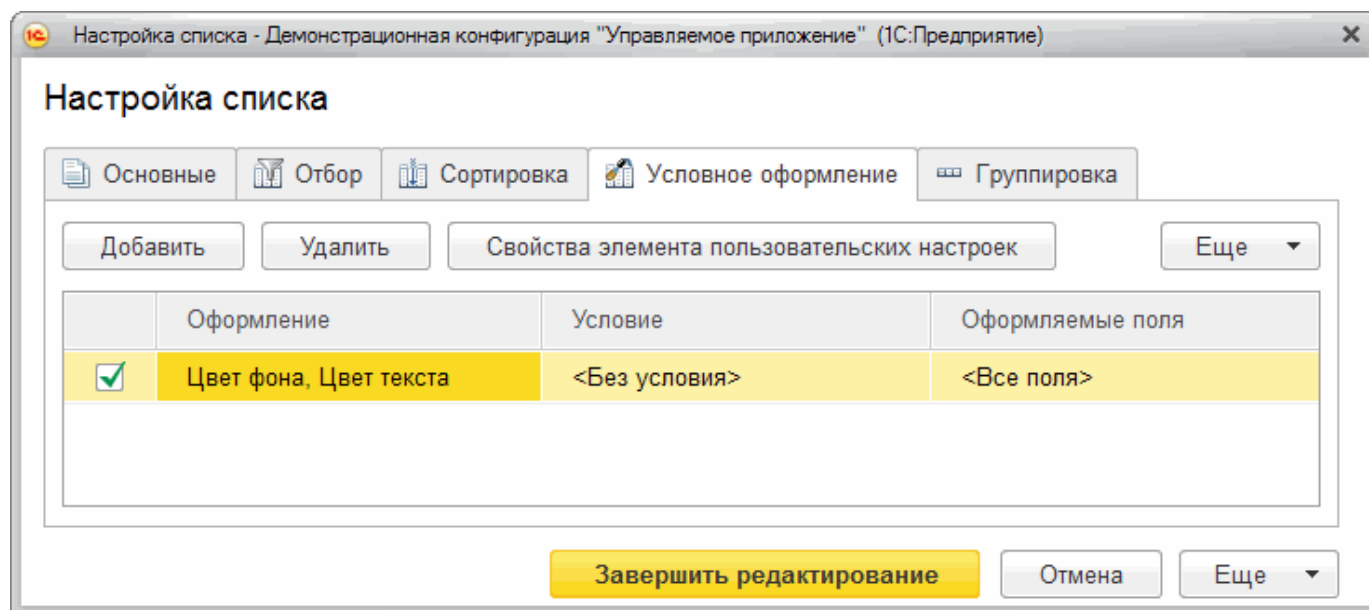


Рис. 91. Настройка условного оформления в списке

Чтобы создать элемент условного оформления, нужно нажать кнопку [Добавить](#). Редактирование элемента условного оформления выполняется в отдельном окне:

- на закладке [Оформление](#) выбрать требуемые параметры оформления: цвет фона, цвет текста и т. д.;
- на закладке [Условие](#) поставить условие, при выполнении которого оформление будет применяться к полям. Условие устанавливается аналогично отбору;
- на закладке [Оформляемые поля](#) выбрать поля из списка доступных полей, к которым будет применяться оформление, если условие выполнено.

Если необходимо, можно установить порядок используемых элементов оформления с помощью команд [Переместить вверх](#) и [Переместить вниз](#).

Порядок элементов оформления устанавливает последовательность, в которой оформление будет применяться к элементам отчета. Если по некоторым условиям к области отчета применяются два разных оформления, то в итоге применяется последнее из списка элементов условного оформления.

Подробнее о настройке условного оформления см. раздел «Условное оформление» [здесь](#).

Чтобы применить настройку условного оформления, следует нажать кнопку [ОК](#).

3.4.4.1. Сохранение настроек между сеансами

Настройки динамического списка сохраняются автоматически при закрытии формы и автоматически загружаются при открытии формы в новом сеансе.

Режим просмотра списка (Иерархический список / Дерево / Список) сохраняется автоматически.

3.4.5. Настройка периода

Для некоторых списков существует возможность ограничить отображаемые данные по установленному периоду.

Интервал следует указать в диалоге [Настройка периода](#), вызываемом командой [Еще – Установить период](#).

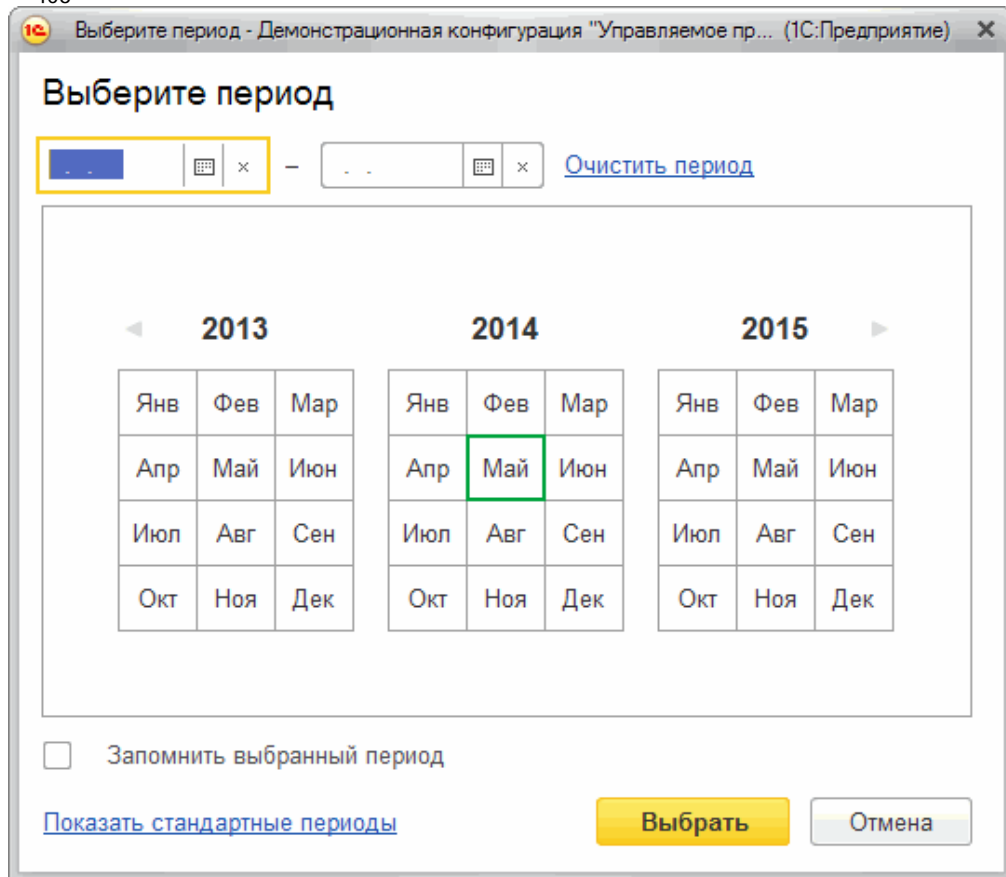


Рис. 92. Настройка периода в списке

В открывшейся форме доступны следующие действия:

- Установить период в поле выбора периода.
- Выбрать один из стандартных периодов.
- Также период можно указать произвольный период, выбрав даты начала и окончания периода в полях выбора дат.

Чтобы очистить выбор периода, следует нажать ссылку [Очистить период](#).

Для сохранения настроек периода при следующем открытии списка следует установить флажок [Запомнить выбранный период](#).

Чтобы завершить настройку периода, следует нажать кнопку [Выбрать](#). Также текущая позиция курсора устанавливается в качестве выбранного периода при нажатии клавиши [Пробел](#) или [Enter](#).

Для отмены настройки периода следует нажать кнопку [Отмена](#).

Выбранный период будет отображен в виде дополнительной кнопки на панели команд формы списка. Чтобы изменить период, нужно нажать на ссылку в новой кнопке.

Для того, чтобы сбросить фильтр по периоду, следует нажать на крестик этой кнопки.

Выбор произвольного периода

На форме период выбирается с помощью мыши одним из следующих способов:

- Нажать левую кнопку мыши на первом месяце периода. Удерживая кнопку мыши, переместить курсор до последнего месяца периода. Отпустить кнопку мыши. Период можно задавать и в обратном порядке (от последнего до первого месяца).
- Нажать левую кнопку мыши на первом месяце периода. После этого нажать левую кнопку мыши на последнем месяце периода. Период может быть задан и в обратном порядке.
- При двойном нажатии на месяце левой кнопки мыши этот месяц выбирается как период.

Добавление/удаление месяцев из периода

Чтобы добавить месяцы в диапазон, нажмите на требуемом месяце левую кнопку мыши, удерживая клавишу **Shift**. При этом:

- Если выбран месяц после последнего месяца текущего выбранного периода, то в период добавляются месяцы от последнего до выбранного (включительно).
- Если выбран месяц ранее первого месяца текущего выбранного периода, то в период добавляются месяцы от выбранного до первого месяца текущего выбранного периода.
- Если выбран месяц внутри выбранного периода, изменений не происходит.
- Если период не был выбран, то период устанавливается так же, как и без удерживания клавиши **Shift**.

Нажатие левой кнопки мыши вместе с клавишей **Ctrl** добавляет или удаляет месяцы в период:

- Если выбран месяц, следующий за последним месяцем выбранного периода или предшествующий первому месяцу выбранного периода, этот месяц добавляется в период.
- Если выбран первый или последний месяц в периоде, тот этот месяц из диапазона удаляется.
- Если период не был выбран, то период устанавливается так же, как и без удерживания клавиши **Ctrl**.

Использование клавиш управления курсором

При нажатии на клавиатуре клавиш **Стрелка вверх**, **Стрелка вниз**, **Стрелка вправо**, **Стрелка влево**:

- Если текущий период - первая строка в году и нажата **Стрелка вверх**, изменений не происходит.
- Если текущий период - последняя строка в году и нажата **Стрелка вниз**, изменений не происходит.
- Иначе период устанавливается на месяц, который находится соответственно сверху, снизу, справа или слева от выбранной позиции.

При нажатии клавиш управления курсором одновременно с клавишей **Shift** текущий выбранный период не сбрасывается, а соответствующие месяцы добавляются в период.

Выбор стандартного периода

Чтобы перейти к выбору стандартного периода в форме настройки следует нажать ссылку **Показать стандартные периоды**.

В левой части формы отобразится список вариантов для каждого выбранного периода.

В правой части формы необходимо нажать кнопку с названием требуемого периода и в левой части выбрать вариант периода.

Чтобы вернуться к выбору произвольного периода, следует нажать ссылку **Показать произвольный период** или выбрать произвольную дату в полях **От** или **до**.

3.5. Печать списка

Для вывода списка в текстовый или табличный документ нужно выбрать пункт **Еще – Вывести список**.

В отобразившемся диалоге выбрать тип документа и требуемые колонки.

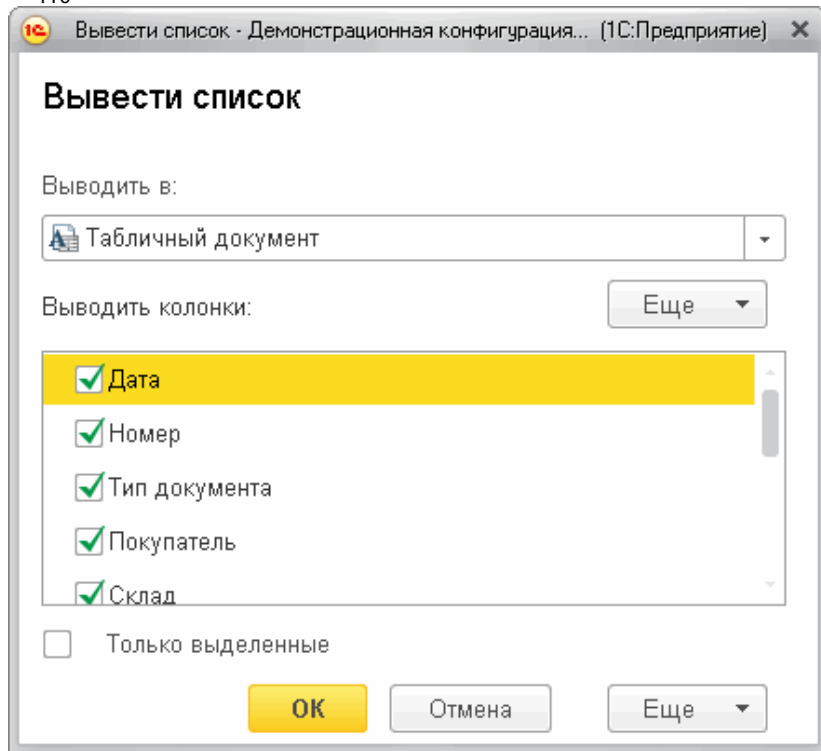


Рис. 93. Диалог настройки вывода списка

Если в списке предусмотрен режим множественного выделения строк, то в диалоге можно указать режим печати только выделенных строк (флажок [Только выделенные](#)).

Если в таблице выводится иерархический список, то в диалоге можно указать режим печати данных с включением подчиненных групп и элементов (флажок [С подчиненными](#)).

ПРИМЕЧАНИЕ: При выводе списка дополнительные настройки колонок (группировка и положение) не учитываются.

Глава 4. Работа с данными различных видов

Данная глава содержит описание функций для работы с такими данными, как документы, журнал документов, и информацию о специфических свойствах объектов информационной базы.

4.1. Нумерация элементов

Код. Одним из обязательных реквизитов объекта конфигурации является его код (для документов – номер).

В зависимости от конфигурации при создании элемента его код (номер) может формироваться автоматически или вводиться вручную.

Если для конкретного объекта установлена автоматическая нумерация, то после сохранения элемента поле для ввода кода (номера) будет содержать код (номер), сформированный системой. Этот код (номер) можно исправить, однако программа проследит, чтобы указанный код (номер) не совпадал с кодами (номерами) объектов этого же вида, уже существующих в базе.

Если при сохранении элемента система обнаружит, что такой код (номер) уже использован, она выдаст соответствующее предупреждение.

Префикс. Для объектов конфигурации может быть задан префикс кода (номера). Таким образом, при создании элемента очередной код (номер) будет предложен не только исходя из правил автоматического присвоения кодов (номеров), но и с определенным префиксом. Тем не менее пользователь может полностью отредактировать предложенный код (номер), включая его префиксную часть.

4.2. Ввод элемента на основании

В платформе «1С:Предприятие» существует механизм создания новых элементов на основании имеющихся данных.

Этот механизм позволяет, например, выставить счет на оплату, используя информацию из ранее созданной расходной накладной.

Порядок переноса конкретных реквизитов объекта-образца в объект-копию устанавливается при создании объектов в процессе разработки конфигурации.

Информацию об использовании возможности ввода объекта на основании другого объекта можно получить в описании конфигурации.

Для ввода объекта на основании другого объекта в форме списка следует выбрать объект-основание, на командной панели выбрать пункт [Создать на основании](#) и в выпадающем подменю выбрать вид создаваемого объекта.

После этого на экран будет выдана форма создания объекта выбранного вида, в котором реквизиты будут заполнены информацией из объекта-основания.

При создании документа ему присваивается очередной порядковый номер. В качестве даты документа будет установлена текущая дата.

4.3. Удаление (пометка удаления) элемента (группы)

Различают два режима удаления: непосредственное удаление и пометка удаления. Доступность режимов определяется правами конкретного пользователя. Если разрешено непосредственное удаление, то ответственность за нарушение ссылочной целостности информационной базы возлагается на администратора, разрешившего использование данного режима, и пользователя программы, выполнившего удаление элементов, на которые в информационной базе имеются ссылки.

Как правило, если это предполагается в конкретной конфигурации, система поддерживает ссылочную целостность объектов информационной базы: в этом режиме не проводится непосредственное удаление таких объектов системы, как, например, документы, – можно лишь установить пометку удаления объекта, но не удалить его. Собственно удаление производится при выполнении соответствующей сервисной процедуры – пункт [Поиск и удаление помеченных объектов](#) из списка [Все функции](#), если эта операция доступна конкретному конечному пользователю.

Чтобы установить пометку удаления элемента, нужно поместить курсор на строку с ним и выбрать пункт [Еще – Пометить на удаление](#).

Для отмены пометки на удаление элемента списка поместить курсор на строку с ранее помеченным элементом и выбрать пункт [Еще – Снять пометку удаления](#). Это будет отражено в изменении условного значка в крайней левой колонке списка. Также команды установки и снятия пометки удаления могут быть доступны в форме объекта, в меню [Еще](#).

ВНИМАНИЕ! Установка и снятие пометки удаления группы распространяются на все элементы, входящие в группу. Это же касается элементов подчиненного списка.

Подробнее о режимах удаления (непосредственное удаление, установка и снятие пометки удаления) см. в книге «1С:Предприятие 8.3. Руководство администратора».

4.4. Работа с данными из внешнего источника

Программа «1С:Предприятие» предоставляет возможность работать с данными из внешних баз. В зависимости от настроек программы данные из внешнего источника можно просматривать, использовать в отчетах, работать с ними через веб-клиента.

Общие параметры соединения с внешним источником настраивает администратор системы.

При любом обращении к данным внешнего источника, если к этому источнику подключение не было выполнено, будет выполнено подключение. Если подключение выполнено успешно, то выбранное действие продолжает выполняться. Если не были установлены параметры соединения, то при попытке подключения программа отобразит диалог для ввода этих параметров.

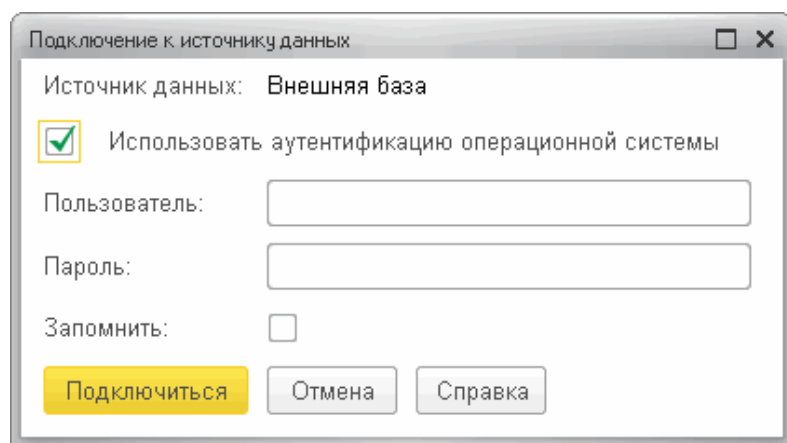


Рис. 94. Диалог подключения к внешнему источнику данных

При наличии соответствующих прав имя пользователя и пароль могут быть сохранены и использованы при последующих сеансах работы. Чтобы сохранить параметры, нужно установить флажок [Запомнить](#). После ввода данных программа выполнит подключение к внешнему источнику и отобразит сообщение [Соединение с внешним источником данных выполнено. Повторите действие](#). После этого можно продолжить работу с данными внешнего источника.

4.5. Печатная форма объекта

Если в процессе настройки конфигурации для объекта помимо экранной формы диалога была создана также печатная форма, то в диалоге редактирования объекта, как правило, присутствует кнопка, нажатие которой вызывает построение этой формы. Обычно такая кнопка имеет надпись [Печать](#), хотя может иметь и какую-либо другую подобную надпись.

После нажатия кнопки [Печать](#) (или выполнения каких-либо иных действий, если это указано в описании конфигурации) будет построена печатная форма.

Созданная печатная форма может открываться в отдельном окне стандартного табличного редактора системы «1С:Предприятие». Если в окне видна только часть формы, можно использовать клавиши управления курсором и линейки прокрутки для вывода на экран остальной части формы.

При необходимости автоматически построенная печатная форма может быть отредактирована. Чтобы иметь такую возможность, следует включить режим редактирования с помощью пункта главного меню [Таблица – Вид – Редактирование](#).

Отредактированная печатная форма может быть сохранена в файле на диске. Для этого следует выбрать пункт главного меню [Файл – Сохранить](#) или [Файл – Сохранить как](#).

Работа в табличном редакторе системы «1С:Предприятие» подробно описана в Приложении 3 книги «1С:Предприятие 8.3. Руководство пользователя», доступной в электронной версии документации.

Для непосредственного вывода печатной формы на принтер следует выбрать пункт главного меню **Файл – Печать** главного меню. На экран будет выдан стандартный диалог для настройки параметров печати. Подробнее см. раздел «Настройка печати» Приложения 3 книги «1С:Предприятие 8.3. Руководство пользователя».

4.6. Работа нескольких пользователей

При работе нескольких пользователей в системе один и тот же объект могут пытаться отредактировать несколько человек. В этом случае только первому начавшему правку удастся завершить редактирование объекта. Всем остальным при попытке начать редактировать объект система выдаст предупреждение с номером сеанса и именем компьютера, с которого был заблокирован этот объект. Например, так:

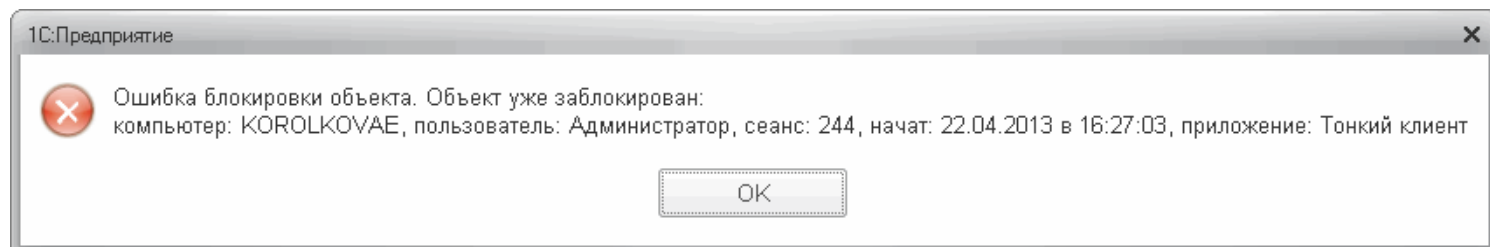


Рис. 95. Ошибка блокировки объекта

Чтобы внести свои изменения в освобожденный объект, его следует перечитать из базы данных.

4.7. Документы и журналы документов

В системе «1С:Предприятие» информация о хозяйственных операциях, совершаемых на предприятии, регистрируется при помощи документов. Документы могут отображаться в списках документов одного вида или журналах. Журнал представляет собой список документов нескольких типов.

В общем случае таблица журнала документов всегда включает в себя колонку **Тип документа**, которая содержит наименование документа.

Дата	Номер	Тип документа	Контрагент	Расчетный счет
12.09.2012 15:32:00	000000017	Поступление денег	Попов Б.В. ИЧП	Основной
10.09.2012 14:22:41	000000016	Поступление денег	Магазин "Продукты"	Вспомогательный
07.09.2012 12:00:00	000000023	Поступление денег	Шлюзовая ООО	Основной
06.09.2012 9:34:25	000000015	Поступление денег	Шлюзовая ООО	Вспомогательный
05.09.2012 13:46:50	000000014	Поступление денег	Попов Б.В. ИЧП	Дополнительный
05.09.2012 12:10:05	000000013	Поступление денег	Магазин "Мясная л..."	Основной
05.09.2012 11:14:05	000000019	Поступление денег	Магазин "Бытовая ..."	Вспомогательный
04.09.2012 12:00:00	000000020	Оплата	Животноводство О...	Основной

Рис. 96. Журнал документов «Финансовые документы»

Как правило, левая графа журнала и списка документов – служебная. В ней различными пиктограммами обозначается состояние документа:

	Дата	↑	Номер	Тип документа
Записан	15.06.2012 14:16:36		000000013	Оплата
Проведен	07.06.2012 12:18:23		000000003	Оплата
К удалению	27.05.2012 13:20:03		000000010	Оплата

Рис. 97. Состояния документа в журнале

- **Записан** – документ сохранен (записан), но не проведен;
- **Проведен** – документ проведен, или для документа не предусмотрено проведение;
- **К удалению** – документ помечен на удаление.

4.7.1. Просмотр журнала документов

Для просмотра журналов документов используются общие приемы работы со списками. Подробно эти приемы описаны в разделе «[Просмотр списка](#)» [здесь](#).

Если в списке журнала документов выбран режим сортировки по дате, то при показе журнала записи выводятся в хронологическом порядке. При этом выполняются следующие правила:

- записи в журнале упорядочены по дате,
- внутри даты записи упорядочены по времени документа.

Если сортировка по дате отключена, то документы выводятся в порядке ввода.

4.7.2. Ввод нового документа из журнала документов

Если в журнале отображаются документы нескольких видов, то команда **Создать** представляет собой подменю, в котором можно выбрать вид создаваемого документа.

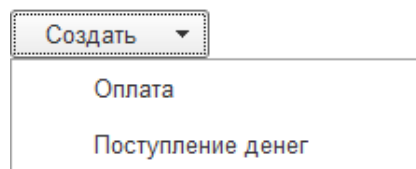


Рис. 98. Запрос выбора вида документа

В этом подменю будут выданы наименования только тех видов документов, которые отображаются в текущем журнале.

Если соответствующая команда предусмотрена в группе **Создать** панели действий, ввод документов можно проводить, не открывая журнала или списка документов.

В списке видов документов необходимо указать наименование нужного вида документов и нажать кнопку **OK**. После этого на экран будет выдана форма нового документа для заполнения его реквизитов.

Если в журнале хранятся документы только одного вида, список видов документов выдаваться не будет, а сразу будет выдан диалог для заполнения реквизитов документа.

4.7.3. Дата и время документа

Практически любой документ имеет поля для ввода даты и номера документа.

Возможность указания времени документа является важной особенностью системы «1С:Предприятие». Документы выстраиваются в хронологическом порядке и зачастую обрабатываются именно в той хронологической последовательности, которую задают дата и время документа. Таким образом, время документа служит не столько для отражения астрономического времени его ввода в систему, сколько для четкого упорядочивания документов внутри даты.

Предложенную системой дату можно изменить. При записи документа ему, как правило, устанавливается

текущее время.

4.7.4. Проведение документа

Проведением документа называется действие, которое отражает данные документа в тех или иных учетных механизмах на основании информации документа. Возможность проведения – свойство документа, определенное заранее. При проведении документа информация, содержащаяся в документе, учитывается в регистрах путем создания записи регистров. Записи регистров – это информация о том, как изменяется состояние регистра в результате проведения документа.

При нажатии кнопки формы документа, для которой определено проведение документа (обычно это кнопка [Провести и закрыть](#) или [Провести](#)), производится проведение документа и документ закрывается. В журнале документов проведенный документ будет помечен пиктограммой (см. [рис. 99](#)).

В некоторых случаях при проведении документа возникает ситуация, из-за которой проведение не может быть выполнено. Например, при проведении расходной накладной обнаружено, что на складе не числится нужного количества товара. При этом будет выдано сообщение о невозможности проведения документа. В этом случае документ не закроется автоматически. Можно внести в него исправления и заново попробовать провести.

Чтобы отменить проведение документа, следует выбрать пункт [Еще – Отмена проведения](#). После подтверждения пиктограмма для документа изменится (см. [рис. 99](#)).





	Дата	Номер
Проведен	 05.09.2012 12:05:00	000000013
	 12.09.2012 15:30:45	000000017
Проведение отменено	 03.08.2012 18:15:38	000000003
	 06.08.2012 14:29:00	000000006

Рис. 99. Состояние проведения документа

При отмене проведения документа, если иного не задано в данной конфигурации, отменяются все действия, которые он выполнил в процессе проведения.

Непроведенный документ можно откорректировать, записать, провести (или не проводить), а пиктограмма в крайней левой колонке журнала будет правильно отражать текущее состояние документа.

4.7.5. Непроводимые документы

Документы, для которых при разработке конфигурации не предусмотрено проведение, не проводятся, но в списках документов показываются такой же пиктограммой, как у проведенного документа. Нельзя отменить проведение у непроводимых документов (см. [100](#)).



	Дата	↓	Номер
Непроводимый документ	 29.04.2013 15:42:24		000000001
Непроводимый документ помечен на удаление	 29.04.2013 15:44:45		000000002

Рис. 100. Список непроводимых документов

4.7.6. Просмотр движений документов

Для анализа тех изменений, которые произвел в регистрах тот или иной проведенный документ, в конфигурации может быть предусмотрена возможность просмотра движения регистров. В описании конфигурации указано, каким образом выполняется просмотр движений регистров.

Форма просмотра движения регистров представляет собой список. В нем отображаются движения регистров. Состав колонок списка зависит от структуры выбранного регистра.

Для регистра товарных запасов форма просмотра движений может выглядеть следующим образом:

Период	↑	Регистратор	Ном...	Товар	Склад	Количество
- 12.09.2012 15:30:45		Продажа 000...	1	Йогурт	Малый	300,00
- 10.09.2012 14:21:55		Продажа 000...	1	Йогурт	Малый	300,00
+ 06.09.2012 12:00:00		Поступление ...	1	Sony K3456P	Средний	3,00
- 06.09.2012 9:30:12		Продажа 000...	1	Sony K3456P	Средний	2,00
- 05.09.2012 13:46:50		Продажа 000...	1	Bosch15	Большой	3,00
- 05.09.2012 12:05:00		Продажа 000...	1	Молоко	Большой	12,00
+ 04.09.2012 12:00:00		Поступление ...	1	Колбаса	Большой	10,00
- 04.09.2012 10:20:25		Продажа 000...	2	Сметана	Большой	11,00
- 04.09.2012 10:20:25		Продажа 000...	1	Колбаса	Большой	10,00
- 03.09.2012 14:25:15		Продажа 000...	1	Йогурт	Малый	50,00
+ 03.09.2012 12:00:00		Поступление ...	1	Молоко	Большой	20,00
+ 31.08.2012 12:17:05		Поступление ...	2	Сметана	Большой	11,00

Рис. 101. Регистр товарных запасов

Форма списка регистра всегда имеет две обязательные колонки.

Колонка **Номер строки** может содержать номер записи в наборе записей регистра, создаваемого при обработке документа.

В колонке **Период** отображается особая пиктограмма, которая служит для обозначения характера произведенных изменений в данном регистре. Знак + (плюс) обозначает прирост абсолютного значения измерений регистра, знак – (минус) – уменьшение. Именно знак + (плюс) позволил определить в приведенном выше примере, что количество товара увеличилось.

4.8. Бизнес-процессы

Бизнес-процессы в «1С:Предприятии» предназначены для объединения отдельных операций в цепочки взаимосвязанных действий, приводящих к достижению конкретной цели. Например, цепочку по выписке счета, приему наличной оплаты и отпуску товара со склада можно представить как бизнес-процесс **Продажа товара**.

Цепочки взаимосвязанных действий бизнес-процесса представляются с помощью **карты маршрута бизнес-процесса**. Карта маршрута описывает логику бизнес-процесса и весь его жизненный цикл от точки старта до точки завершения в виде схематического изображения последовательности прохождения взаимосвязанных точек маршрута.

Точка маршрута отражает этап жизненного цикла бизнес-процесса, связанный с выполнением, как правило, одной автоматической или ручной операции.

Задачи в «1С:Предприятии» позволяют вести учет заданий по исполнителям и служат отражением продвижения бизнес-процессов по точкам маршрута. При этом задачи могут создаваться не только бизнес-процессами, но и другими объектами информационной базы и непосредственно пользователями.

При переходе бизнес-процесса на какую-либо точку маршрута, предусматривающую действия конкретного исполнителя, формируется задача (или несколько задач в случае группового действия). После того как исполнитель отметит задачу как выполненную, бизнес-процесс автоматически переходит к следующей точке маршрута в соответствии с картой. Таким образом, задачи являются движущей силой бизнес-процессов.

Описание конкретных бизнес-процессов и их взаимосвязи с задачами приводится в описании конфигурации.

Общие принципы работы с бизнес-процессами и задачами аналогичны работе с другими объектами (например,

документами и списками), поэтому в данном руководстве будут описаны только особенности.

4.8.1. Список бизнес-процессов

По умолчанию в списке бизнес-процессов отображается следующая информация:

- **Дата** – дата и время создания бизнес-процесса;
- **Номер** – номер (является уникальным для данного бизнес-процесса);
- **Стартован** – пометка старта бизнес-процесса;
- **Завершен** – пометка завершения бизнес-процесса, т. е. все задачи, порожденные бизнес-процессом, были выполнены;
- **Ведущая задача** – задача, на основе которой был создан данный бизнес-процесс.

Список бизнес-процессов **Продажа товара** может выглядеть следующим образом:

Дата	Номер	Стартован	Завершен	Ведущая задача
29.04.2013 15:55:46	000000001	✓		Кофеварка в ООО "Арго"...
29.04.2013 15:58:11	000000002			Чайники в ООО "Слон" О...
29.04.2013 16:05:56	000000003	✓	✓	Пылесосы в ТД "Осинка"...
29.04.2013 16:07:41	000000004			Телевизоры Samsung ОО...

Рис. 102. Список бизнес-процессов «Продажа товара»

С помощью пиктограмм обозначается состояние бизнес-процесса:

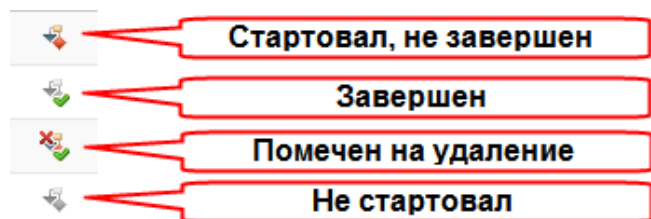


Рис. 103. Состояния бизнес-процесса

Если бизнес-процесс еще не стартовал, т. е. еще нет ни одной соответствующей ему задачи, не установлена пометка **Стартован**, то эта пиктограмма отображается черно-белой, а не цветной.

Бизнес-процесс считается завершенным, когда достигает точки завершения в соответствии с картой маршрута и все задачи по нему выполнены.

4.8.2. Список задач

По умолчанию в списке задач отображается следующая информация:

- **Номер** – номер (является уникальным для данной задачи);
- **Наименование** – наименование задачи;

- **Дата** – дата и время создания задачи;
- **Выполнена** – отметка о выполнении задачи;
- **Бизнес-процесс** – бизнес-процесс, породивший данную задачу;
- **Точка** – точка маршрута бизнес-процесса, в которой была создана данная задача.

Кроме этих данных могут быть определены дополнительные колонки для вывода значений любых других реквизитов бизнес-процесса или задачи.

Состояние задачи обозначается с помощью пиктограмм:

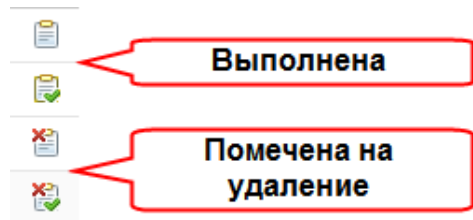


Рис. 104. Состояния задачи

4.8.3. Выполнение задачи

Выполнение задачи – важный этап в жизненном цикле бизнес-процесса. При выполнении задачи бизнес-процесс осуществляет переход на следующую точку маршрута в соответствии с картой маршрута, что приводит к формированию новых заданий в рамках данного бизнес-процесса.

Для того чтобы выполнить задачу, следует выбрать пункт **Еще – Выполнено** для нужной задачи в форме задачи.

В системе может быть предусмотрено открытие дополнительных окон при выполнении задачи, например, для выдачи предупреждающих сообщений или ввода дополнительной информации. Также может быть предусмотрена проверка, которая запрещает выполнение задачи, если не соблюдены определенные условия (например, не согласован документ или не установлена скидка по счету).

Выполненные задачи помечаются специальной пиктограммой, и у них устанавливается флажок **Выполнена**.

Глава 5. Отчеты

Для получения данных из информационной базы в «1С:Предприятии» используются отчеты.

В режиме [1С:Предприятие](#) пользователь может сформировать отчет с предложенными системой настройками или установить свои настройки. Опытный пользователь может самостоятельно разработать свой собственный вариант отчета.

Внешний вид формы отчета по умолчанию представлен на рисунке ниже.

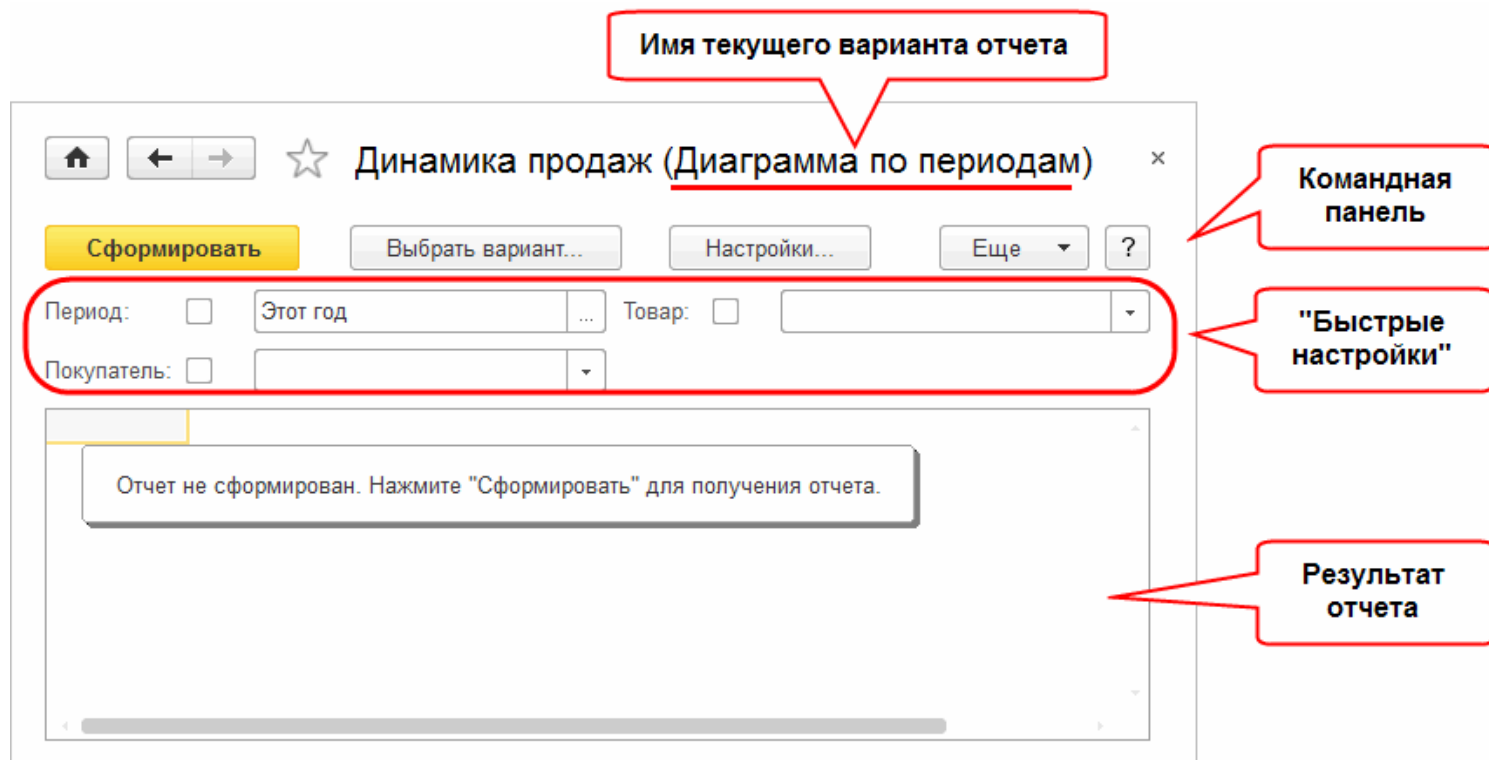


Рис. 105. Форма отчета

Чтобы сформировать отчет с предустановленными настройками, достаточно нажать кнопку [Сформировать](#) на командной панели отчета.

Если для отчета предусмотрено несколько вариантов, то перед выполнением следует выбрать требуемый вариант. Для этого на форме отчета используется кнопка [Выбрать вариант](#). Подробнее о работе с вариантами см. раздел [«Варианты отчета»](#) [здесь](#).

Для того чтобы установить настройки отчета, можно воспользоваться полем быстрых настроек или же специальным диалогом, открываемым нажатием кнопки [Настройки...](#) Подробнее о настройках см. разделы [«Использование быстрых настроек»](#) и [«Настройки»](#) далее.

После того как результат отчета сформирован системой и выведен в табличный документ, можно использовать механизм расшифровки. Подробнее см. раздел [«Работа с расшифровкой отчета»](#) [здесь](#).

В зависимости от настройки отчета на форме отчета в подменю [Еще](#) могут быть доступны следующие команды:

- [Настройки](#) – открывает форму настроек;
- [Выбрать настройки](#) – открывает форму для выбора ранее сохраненной настройки;
- [Сохранить настройки](#) – сохраняет измененные настройки отчета;
- [Установить стандартные настройки](#) – загружает настройки из варианта отчета;
- [Изменить вариант](#) – открывает окно настройки варианта отчета;
- [Выбрать вариант](#) – открывает форму для выбора варианта отчета;
- [Сохранить вариант](#) – открывает форму для сохранения варианта отчета;

- 420
- [Открыть новое окно](#) – открывает новую форму отчета с текущими настройками;
 - [Изменить форму](#) – открывает форму настройки формы отчета.

Подробнее о работе с настройками см. раздел «Использование быстрых настроек» и главу «Изменение варианта отчета» далее.

Работа с расшифровкой отчета описана в разделе «[Работа с расшифровкой отчета](#)» [здесь](#).

ПРИМЕЧАНИЕ. На [рис. 105](#) и далее в примерах приведены формы, в которых системные команды скрыты.

5.1. Настройка отчета

Некоторые настройки отчета можно редактировать в отдельной форме. При выполнении отчета такие настройки используются вместе с основными настройками отчета.

Чтобы открыть форму редактирования настроек, следует нажать кнопку [Настройки...](#) на форме отчета.

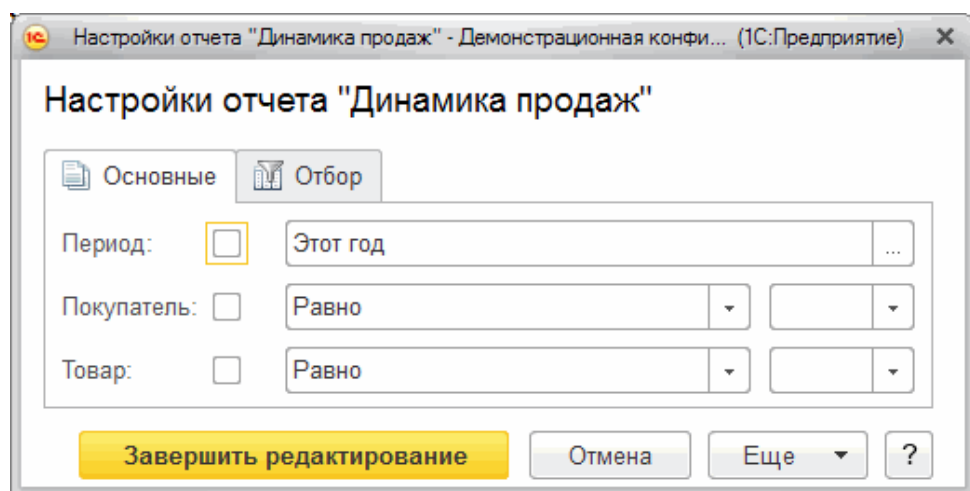


Рис. 106. Форма настроек отчета «Динамика продаж»

Для сохранения выбранных настроек следует нажать кнопку [Завершить редактирование](#).

Чтобы отредактировать состав настроек, отображаемых на форме отчета, нужно выбрать команду [Еще – Изменить состав настроек](#). Форма редактирования списка настроек откроется в новом окне.

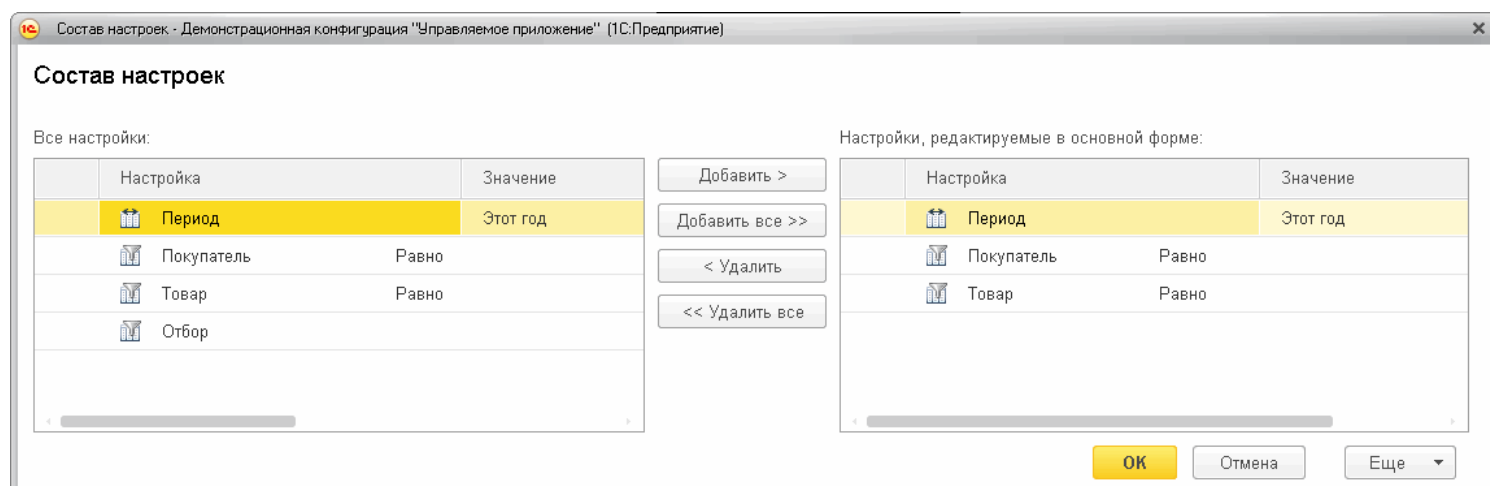


Рис. 107. Изменение состава настроек отчета

Слева отображаются все доступные настройки отчета, справа – настройки, редактируемые в форме отчета.

Для управления составом пользовательских настроек используются кнопки:

- [Добавить >](#) – добавляет настройку из левого списка в правый;
- [Добавить все >>](#) – добавляет все настройки из левого списка в правый;

- < [Удалить](#) – удаляет настройку из правого списка;
- << [Удалить все](#) – удаляет все настройки из правого списка;

Стандартные пользовательские настройки восстанавливаются командой [Еще – Установить стандартные настройки](#).

Если при помощи настроек не удастся получить требуемый отчет, то можно создать собственный вариант отчета. Создание варианта отчета требует наличия некоторого опыта работы с отчетами. Подробнее о работе с вариантами отчета см. главу «[Изменение варианта отчета](#)» [здесь](#).

При закрытии формы отчета или смене текущего варианта отчета пользовательские настройки автоматически сохраняются. Пользовательские настройки сохраняются отдельно для каждого варианта отчета.

Среди настроек могут быть выделены такие настройки, которые предполагается редактировать наиболее часто (например, фильтр по товару в отчете [Динамика продаж](#) или фильтр по организации в бухгалтерском отчете). Такие настройки будут редактироваться непосредственно в форме отчета.

Если в форме настроек отчета установить отбор, то элемент отбора будет добавлен в список настроек и в настройки, отображаемые в форме отчета.

Чтобы использовать в отчете выбранную настройку, нужно установить значение настройки в соответствующем поле.

В зависимости от типа настройки для изменения значения следует выполнить одно из действий:

- выбрать значение из всплывающего списка по нажатию кнопки [Выбрать](#) в поле настройки,
- установить значение в новом окне, отобразившемся по нажатию кнопки [Выбрать](#) в поле настройки.

Пример выбора настройки отчета в новом окне представлен на рисунке:

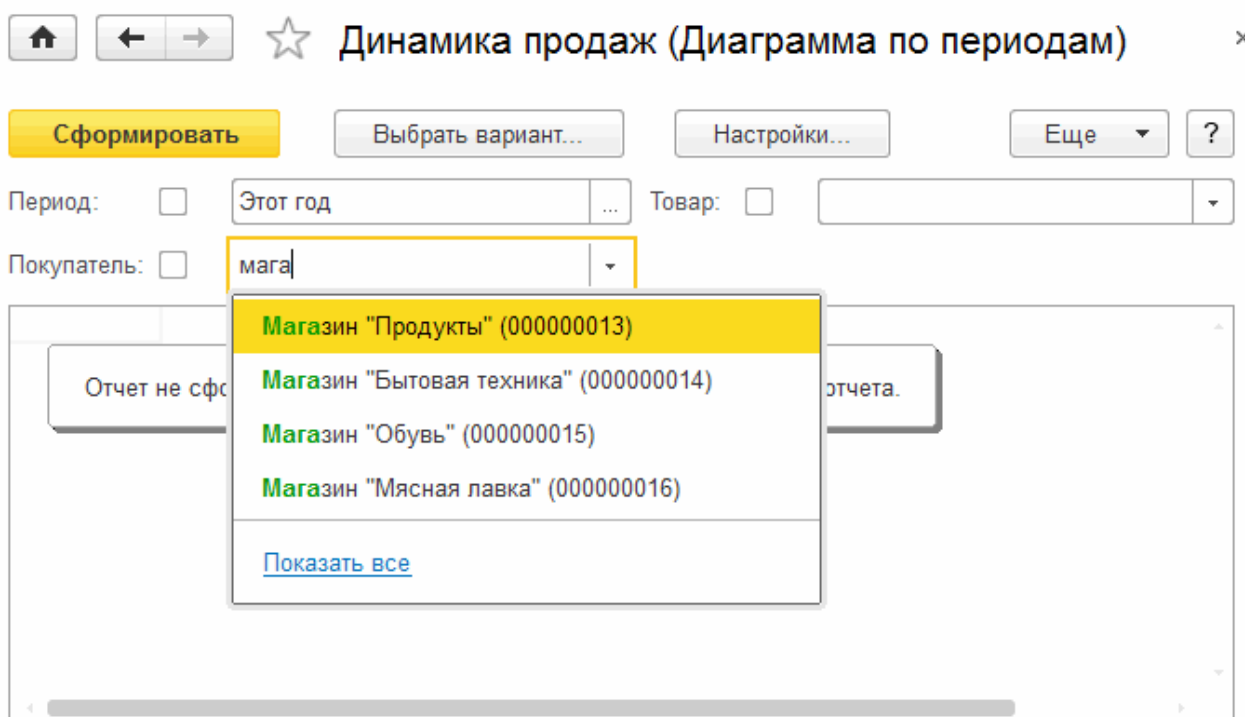


Рис. 108. Выбор значения быстрой настройки отчета

При изменении настроек не происходит формирования отчета.

Чтобы сформировать отчет с установленными настройками, следует нажать кнопку [Сформировать](#).

Чтобы отменить использование выбранной настройки в отчете, нужно снять флажок в соответствующей строке.

5.2. Варианты отчета

Для отчета может быть создано несколько вариантов. Вариант определяет, в каком виде будут выводиться данные в результат отчета. Например, для отчета [Динамика продаж](#) может быть создан вариант, в котором данные о продаже товаров по периодам отображаются в виде диаграммы, и вариант, в котором данные по продажам для покупателей отображаются в виде таблицы. При этом каждый вариант отчета обладает своим набором настроек.

Система предоставляет возможность создания нового варианта отчета непосредственно в режиме «1С:Предприятие». Перед выполнением этого действия внимательно ознакомьтесь с разделом «Изменение варианта отчета». В случае вызова этого действия новый вариант отчета сохраняется в информационной базе.

Имеется возможность определять несколько вариантов настроек для каждого отчета.

Подробнее работа с вариантами настроек описана в главе «Изменение варианта отчета» [здесь](#).

5.3. Работа с отчетами

Рассмотрим подробнее работу пользователя с отчетом на примере отчета [Динамика продаж](#) демонстрационной конфигурации. Запустим отчет с помощью соответствующей команды на панели действий. Форма отчета откроется в новом окне.

Рис. 109. Форма отчета «Динамика продаж»

При запуске для отчета установлен вариант [Диаграмма по периодам](#). Предположим, нам нужен другой вариант.

Тогда нужно нажать кнопку [Выбрать вариант...](#) и в новом окне выбрать вариант отчета [Продажи товаров по покупателям](#).

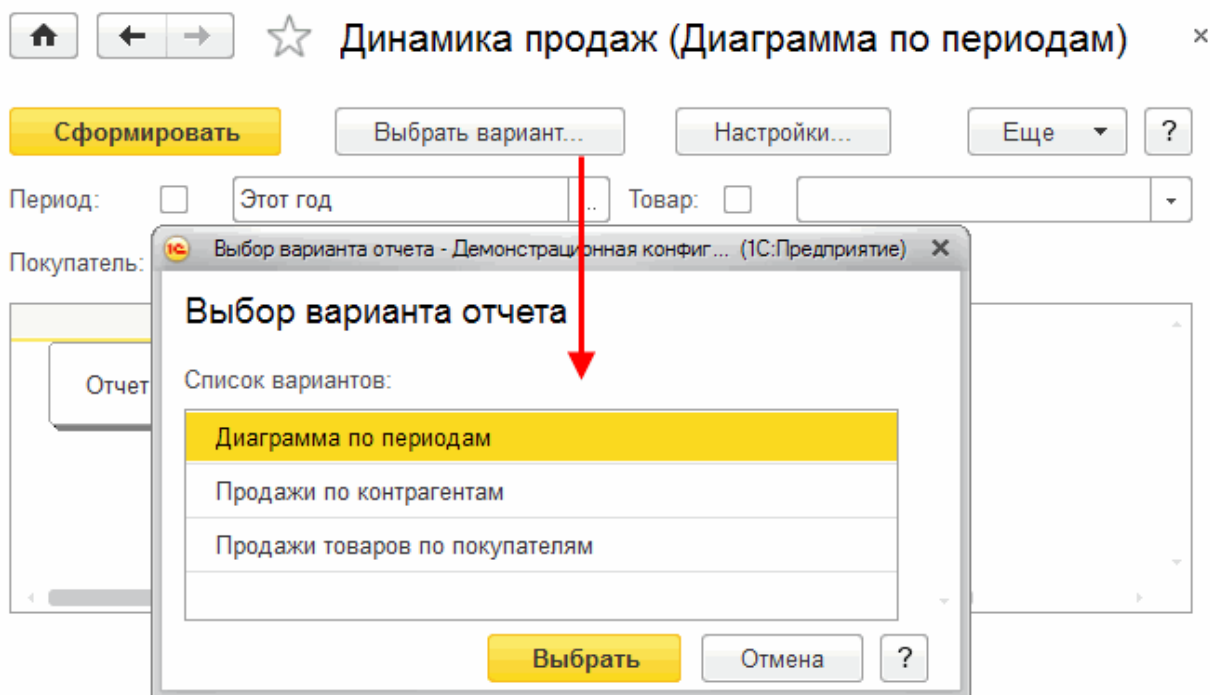


Рис. 110. Выбор варианта отчета «Динамика продаж»

Заметим, что этому варианту отчета были установлены настройки [Период – Этот год](#) и группировка по полям [Товар](#) и [Покупатель](#).

Предположим, что это настройки, необходимые в данный момент, и сформируем отчет нажатием кнопки [Сформировать](#).

Товар	Сумма
Покупатель	Оборот
Bosch1234	5 000,00
Попов Б.В. ИЧП	5 000,00
Bosch15	8 100,00
Попов Б.В. ИЧП	8 100,00
Sony K3456P	13 500,00
Шлюзовая ООО	13 500,00
Veko345MO	10 000,00
Магазин "Бытовая техника"	10 000,00
Veko67NE	15 000,00
Шлюзовая ООО	15 000,00
Босоножки	12 500,00
Попов Б.В. ИЧП	12 500,00
Ботинки	6 000,00
Магазин "Обувь"	6 000,00
Валенки	4 500,00
Магазин "Обувь"	4 500,00
Доставка	300,00

Рис. 111. Отчет «Динамика продаж» в варианте «Продажи товаров по покупателям»

ПРИМЕЧАНИЕ. Для удобства работы в отчетах, содержащих одну таблицу или группировку, заголовок таблицы автоматически фиксируется. При этом если область фиксации табличного документа превышает

5.3.1. Использование быстрых настроек

Допустим, необходимо получить отчет о продажах за предыдущий год. Для этого нам следует изменить параметр **Период** и нажать кнопку **Сформировать**:

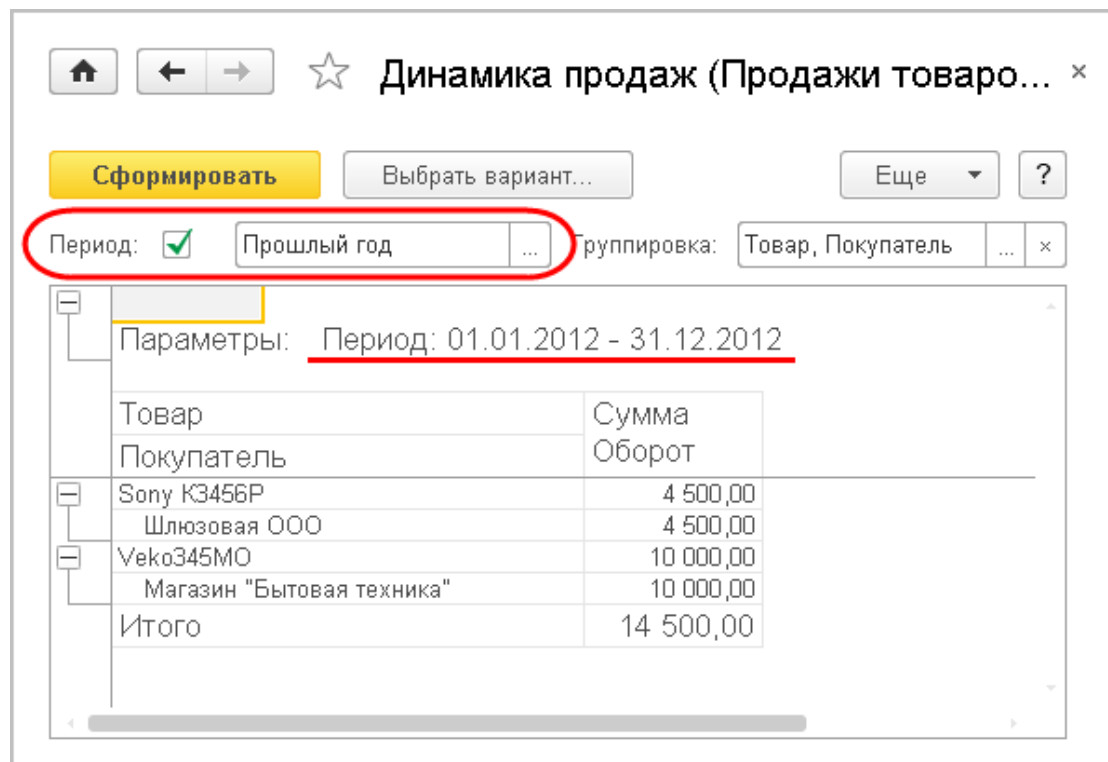


Рис. 112. Отчет «Динамика продаж» за прошлый год

5.3.2. Настройки

Предположим, нам недостаточно изменить видимые параметры (**Период** и **Группировка**). Чтобы увидеть параметры, которые можно настроить для варианта отчета, следует нажать кнопку **Настройки....** Список настроек, распределенных по вкладкам, откроется в отдельном окне:

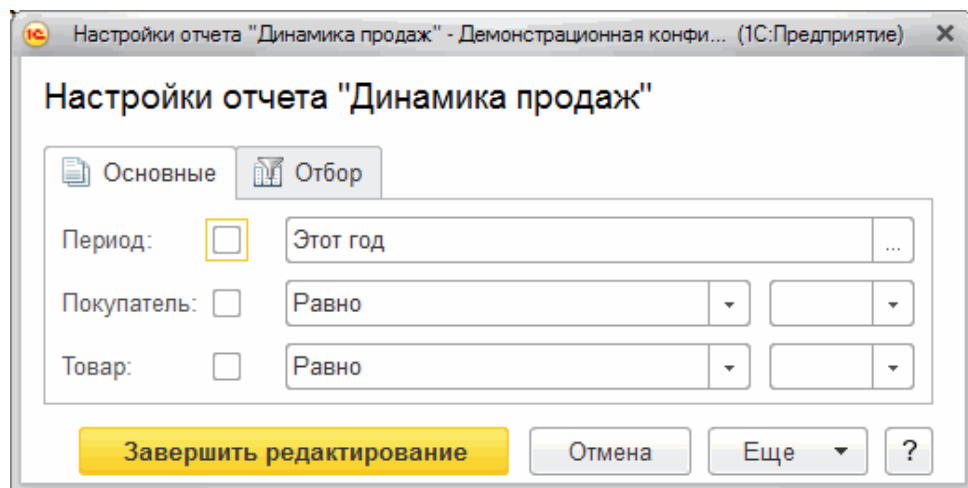


Рис. 113. Форма настроек отчета «Динамика продаж»

5.3.2.1. Отбор

Чтобы установить отбор, следует перейти на закладку **Отбор**.

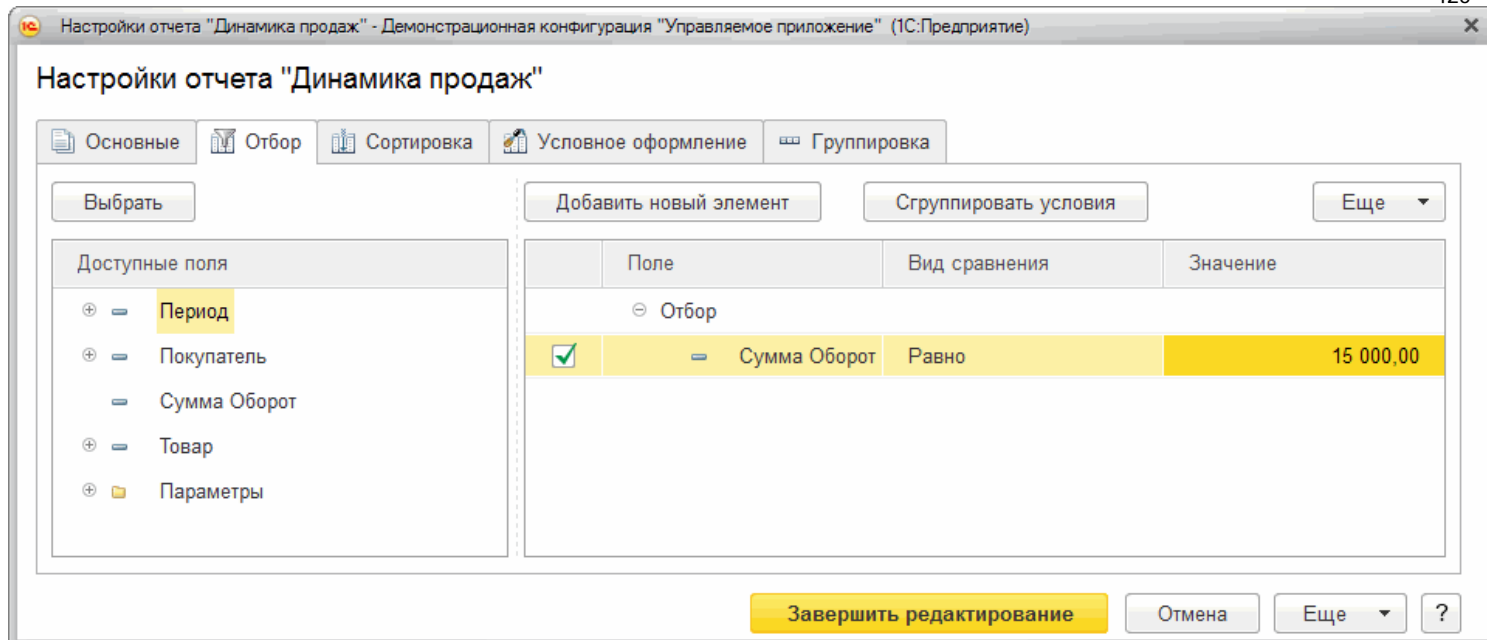


Рис. 114. Форма настройки отбора для отчета «Динамика продаж»

1. На командной панели закладки нажать кнопку [Добавить новый элемент](#) или перетащить поле из списка [Доступные поля](#).
2. Выбрать [Вид сравнения](#) в списке, всплывающем при нажатии кнопки [Выбрать](#).
3. Установить значение отбора, выбрав его в списке, открывающемся при нажатии кнопки [Выбрать](#), или набрать его в колонке.
4. Если необходимо, установить порядок элементов отбора с помощью команд [Переместить вверх](#) и [Переместить вниз](#) из меню [Еще](#). Для удобства можно выбрать несколько условий и добавить их в одну группу с помощью команды [Сгруппировать условия](#).
5. Нажать кнопку [Завершить редактирование](#).

Подробнее о работе с отбором и о видах сравнения см. раздел «Отбор» [здесь](#).

5.3.2.2. Сортировка

Чтобы установить порядок записей результата отчета, в форме настроек нужно перейти на закладку [Сортировка](#).

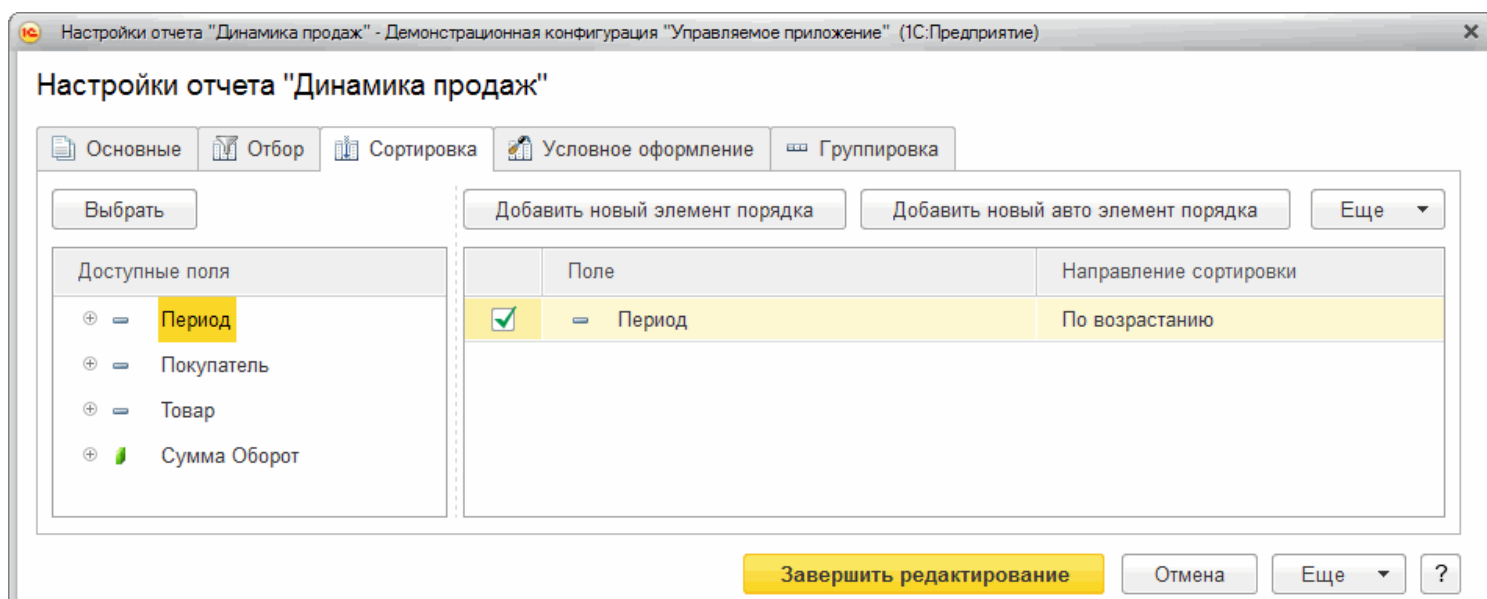


Рис. 115. Настройка сортировки результата отчета «Динамика продаж»

1. На закладке нажать кнопку [Добавить новый элемент](#) и выбрать поле или перетащить поле из списка [Доступные поля](#). Также можно добавить авто элемент порядка нажатием кнопки [Добавить новый авто элемент](#)

порядка (подробнее об авто элементах порядка см раздел "" стр..

2. Выбрать [Направление сортировки \(По возрастанию / По убыванию\)](#).
3. Если необходимо, установить порядок элементов сортировки с помощью команд [Переместить вверх](#) и [Переместить вниз](#) из меню [Еще](#).
4. Нажать кнопку [ОК](#).

Если полей сортировки несколько, то в отчете данные будут отсортированы вначале по первому полю, потом по второму и т. д.

Подробнее о работе с сортировкой см. раздел «Сортировка» [здесь](#).

Пример настройки сортировки

В качестве примера отсортируем отчет [Отчет по оплатам](#) по поставщику и в разрезе поставщика – по дате документа. Настройки отбора будут выглядеть следующим образом:

	Поле	Направление сортировки
<input checked="" type="checkbox"/>	Поставщик	По возрастанию
<input checked="" type="checkbox"/>	Дата	По возрастанию

Рис. 116. Настройка сортировки в отчете

В форме редактирования нажмем [ОК](#). Результирующий отчет примет вид, как на рисунке:

Дата	Номер	Поставщик	Расчетный счет поставщика	Сумма
18.05.2012 12:57:57	000000009	Животноводство ООО	Основной	62 500,00
18.05.2012 14:29:35	000000014	Животноводство ООО	Основной	50 000,00
07.06.2012 12:18:23	000000003	Животноводство ООО	Основной	9 000,00
29.07.2012 14:40:29	000000015	Животноводство ООО	Основной	50 000,00
14.08.2012 14:06:43	000000018	Животноводство ООО	Основной	22 500,00
03.09.2012 14:30:01	000000016	Животноводство ООО	Основной	3 970,00
04.09.2012 12:00:00	000000020	Животноводство ООО	Основной	2 500,00
07.05.2012 12:19:03	000000004	Корнет ЗАО	Основной	60 000,00
29.06.2012 14:06:12	000000011	Мосхлеб ОАО	Основной	4 500,00
06.07.2012 12:19:28	000000005	Мосхлеб ОАО	Основной	4 500,00
05.04.2012 12:19:50	000000006	Пантера АО	Основной	50 000,00
27.05.2012 13:20:03	000000010	Пантера АО	Основной	150 000,00
11.03.2012 9:31:27	000000002	Скороход АО	Основной	2 000,00
09.03.2012 12:20:14	000000007	ЭлектроБыт ЗАО	Основной	50 000,00
09.05.2012 12:20:32	000000008	ЭлектроБыт ЗАО	Основной	70 000,00
15.06.2012 14:16:36	000000013	ЭлектроБыт ЗАО	Основной	500 000,00
23.06.2012 13:14:40	000000012	ЭлектроБыт ЗАО	Основной	100 000,00
04.07.2012 11:18:50	000000019	ЭлектроБыт ЗАО	Основной	31 500,00
29.08.2012 11:12:51	000000017	ЭлектроБыт ЗАО	Основной	105 000,00

Рис. 117. Результат настройки сортировки отчета

5.3.2.3. Условное оформление

Чтобы установить условное оформление записей результата отчета, на форме настроек следует выбрать [Значение](#) в строке [Условное оформление](#) и нажать кнопку выбора или клавишу [F4](#).

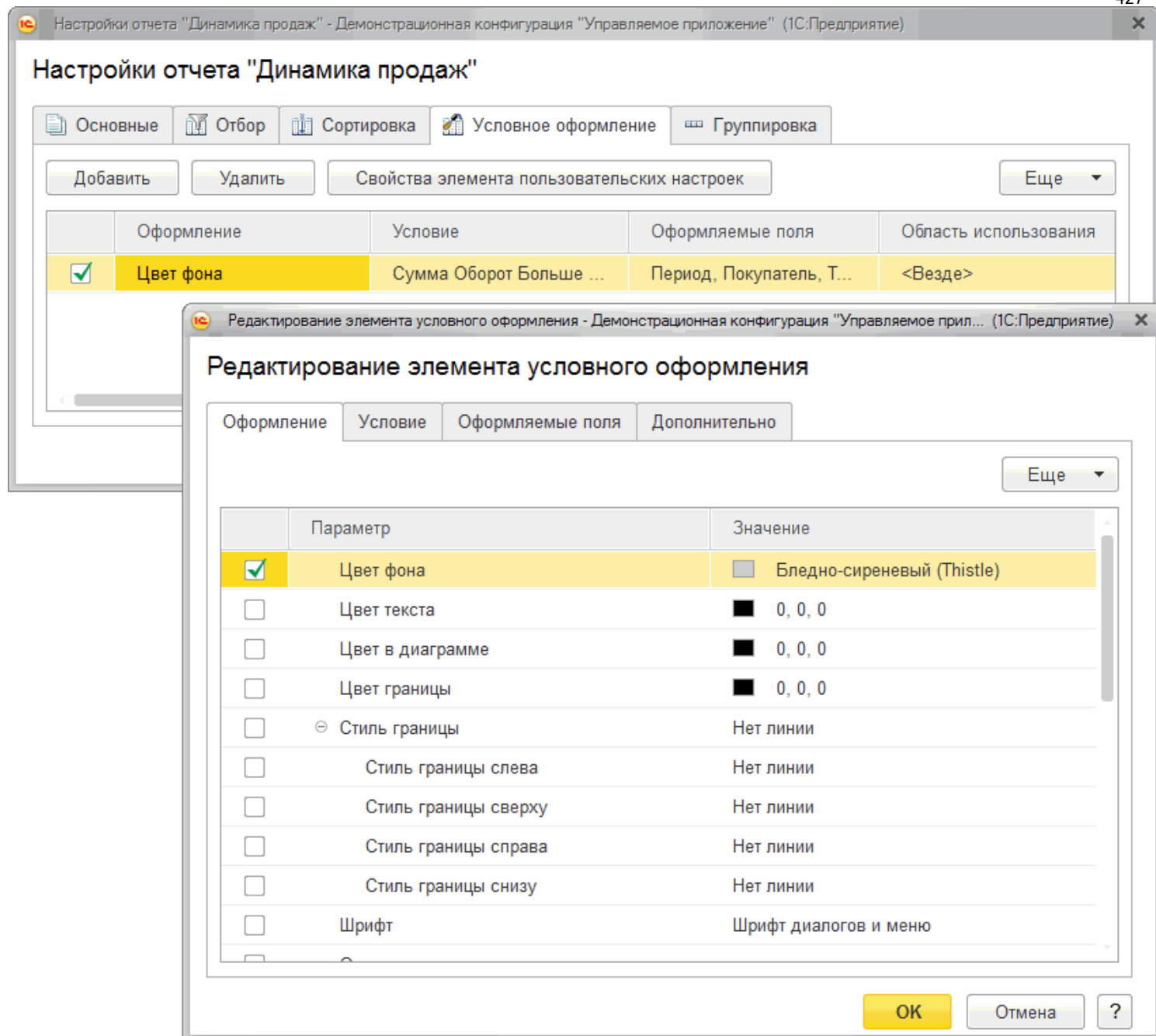


Рис. 118. Настройка условного оформления отчета

1. В открывшемся окне нажать кнопку [Добавить](#).
2. На закладке [Оформление](#) выбрать требуемые параметры оформления: цвет фона, цвет текста и т. д.
3. На закладке [Условие](#) поставить условие, при выполнении которого оформление будет применяться к полям. Условие ставится аналогично отбору.
4. На закладке [Оформляемые поля](#) выбрать поля из списка доступных полей, к которым будет применяться оформление, если условие выполнено.
5. Если необходимо, установить порядок используемых элементов оформления с помощью команд [Переместить вверх](#) и [Переместить вниз](#) из меню [Еще](#).
Порядок элементов оформления устанавливает последовательность, в которой оформление будет применяться к элементам отчета. Если по некоторым условиям к области отчета применяются два разных оформления, то в итоге применяется последнее из списка элементов условного оформления.
6. На закладке [Дополнительно](#) выбрать области отчета, к которым будет применен данный элемент условного оформления.
7. Нажать кнопку [OK](#).

Подробнее о настройке условного оформления см. раздел «Условное оформление» [здесь](#).

Пример настройки условного оформления

В качестве примера в отчете [Отчет по взаиморасчетам](#) выделим документы, которые относятся к **ООО Животноводство**.

Для этого в настройках отчета выполним следующие действия:

1. На закладке [Условное оформление](#) добавим новый элемент.
2. В диалоге редактирования элемента условного оформления на закладке [Оформление](#) нажмем кнопку [Выбрать](#) и в диалоге [Выбор шрифта](#) установим флажок напротив пункта **Жирный** в группе [Начертание](#).
3. На закладке [Условие](#) добавим элемент: [Документ.Организация Равно ООО Животноводство](#).
4. Нажмем [ОК](#) и убедимся, что в колонке [Оформляемые поля](#) отображается **<Все поля>**, а в колонке [Область использования](#) отображается **<Везде>**.

Настройка условного оформления отчета примет вид, как показано на рисунке:

	Оформление	Условие	Оформляемые поля	Область использования
<input checked="" type="checkbox"/>	Шрифт	Поставщик Равно "Животно...	<Все поля>	<Везде>

Рис. 119. Пример настройки условного оформления

В диалоге настройки варианта нажмем кнопку [Завершить редактирование](#). Отчет примет вид, как показано на рисунке.

Дата	Номер	Поставщик	Расчетный счет поставщика	Сумма
07.06.2012 12:18:23	000000003	Животноводство ООО	Основной	9 000,00
07.05.2012 12:19:03	000000004	Корнет ЗАО	Основной	60 000,00
06.07.2012 12:19:28	000000005	Мосхлеб ОАО	Основной	4 500,00
05.04.2012 12:19:50	000000006	Пантера АО	Основной	50 000,00
09.03.2012 12:20:14	000000007	ЭлектроБыт ЗАО	Основной	50 000,00
09.05.2012 12:20:32	000000008	ЭлектроБыт ЗАО	Основной	70 000,00
18.05.2012 12:57:57	000000009	Животноводство ООО	Основной	62 500,00
27.05.2012 13:20:03	000000010	Пантера АО	Основной	150 000,00
29.06.2012 14:06:12	000000011	Мосхлеб ОАО	Основной	4 500,00
23.06.2012 13:14:40	000000012	ЭлектроБыт ЗАО	Основной	100 000,00
15.06.2012 14:16:36	000000013	ЭлектроБыт ЗАО	Основной	500 000,00
18.05.2012 14:29:35	000000014	Животноводство ООО	Основной	50 000,00
29.07.2012 14:40:29	000000015	Животноводство ООО	Основной	50 000,00
11.03.2012 9:31:27	000000002	Скороход АО	Основной	2 000,00
03.09.2012 14:30:01	000000016	Животноводство ООО	Основной	3 970,00
29.08.2012 11:12:51	000000017	ЭлектроБыт ЗАО	Основной	105 000,00
14.08.2012 14:06:43	000000018	Животноводство ООО	Основной	22 500,00
04.07.2012 11:18:50	000000019	ЭлектроБыт ЗАО	Основной	31 500,00
04.09.2012 12:00:00	000000020	Животноводство ООО	Основной	2 500,00

Рис. 120. Настройка условного оформления отчета

5.3.2.4. Группировка

Чтобы создать группировку результата отчета, на форме настроек нужно выбрать [Значение](#) в строке [Группировка](#) и нажать кнопку выбора или клавишу **F4**.

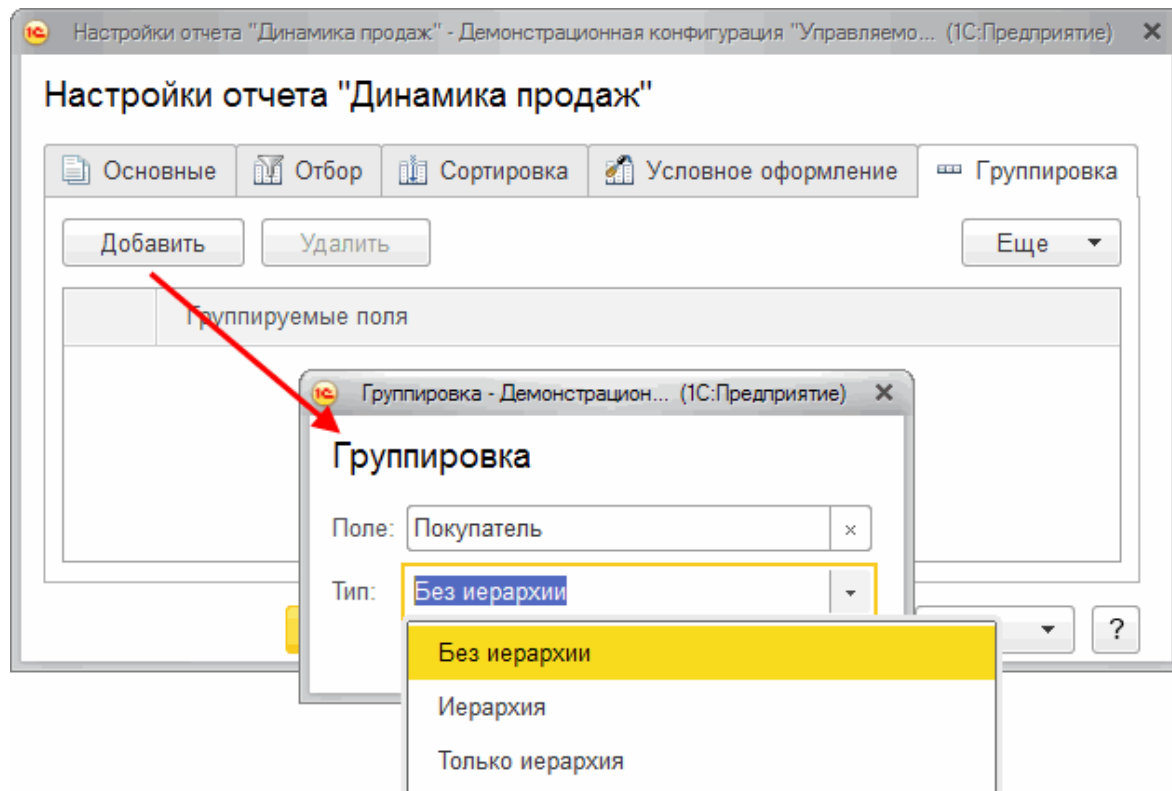


Рис. 121. Настройка группировки отчета

1. Нажать кнопку [Добавить](#).
2. В открывшемся окне выбрать [Поле группировки](#) и указать тип группировки:
 - [Без иерархии](#) – в группировку попадут только неиерархические записи (элементы);
 - [Иерархия](#) – в группировку попадут иерархические записи и неиерархические записи (элементы и группы);
 - [Только иерархия](#) – в группировке будут выводиться только иерархические записи (группы).
3. Нажать кнопку [ОК](#).
4. Если необходимо, установить порядок группировок с помощью кнопок [Переместить вверх](#) и [Переместить вниз](#).
5. Нажать кнопку [ОК](#) в окне редактирования группировок.

Подробнее о работе с группировками см. раздел «Группируемые поля» [здесь](#).

Пример настройки группировки

В качестве примера сгруппируем результат отчета [Отчет по оплатам](#) по имени поставщика. Для этого в качестве поля группировки выберем поле [Поставщик](#) с типом [Иерархия](#).

Результат выполнения отчета показан на рисунке:

Поставщик				
Дата	Номер	Расчетный счет поставщика	Сумма	
Поставщики				
Животноводство ООО				
07.06.2012 12:18:23	000000003	Основной	9 000,00	
18.05.2012 12:57:57	000000009	Основной	62 500,00	
18.05.2012 14:29:35	000000014	Основной	50 000,00	
29.07.2012 14:40:29	000000015	Основной	50 000,00	
03.09.2012 14:30:01	000000016	Основной	3 970,00	
14.08.2012 14:06:43	000000018	Основной	22 500,00	
04.09.2012 12:00:00	000000020	Основной	2 500,00	
Корнет ЗАО				
07.05.2012 12:19:03	000000004	Основной	60 000,00	
Мосхлеб ОАО				
06.07.2012 12:19:28	000000005	Основной	4 500,00	
29.06.2012 14:06:12	000000011	Основной	4 500,00	
Пантера АО				
05.04.2012 12:19:50	000000006	Основной	50 000,00	
27.05.2012 13:20:03	000000010	Основной	150 000,00	

Рис. 122. Отчет по оплатам, сгруппированный по полю «Поставщик»

5.4. Работа с расшифровкой отчета

При наведении указателя мыши на ячейку результата отчета, для которой возможна расшифровка, указатель изменит свою форму (см. рисунок). Теперь, если дважды щелкнуть левой кнопкой мыши на этой ячейке, в новом окне откроется значение, находящееся в ячейке, а для ячейки с ресурсами будет предложено выбрать поле для расшифровки, после чего выполнится расшифровка.

При щелчке правой кнопкой мыши отобразится меню расшифровки.

Товар	Сумма
Покупатель	Оборот
Bosch1234	5 000,00
Попов Б.В. ИЧП	5 000,00
Bosch15	8 100,00
	8 100,00
	10 000,00
	10 000,00
	15 000,00
	6 000,00
	6 000,00
	4 500,00
	4 500,00
	22 500,00
	10 500,00

Расшифровать...

Открыть "Товар = Bosch15"

Отфильтровать ▶

Упорядочить ▶

Сгруппировать ▶

Оформить ▶

Копировать

Свойства

Рис. 123. Меню расшифровки для ячейки отчета

Расшифровать – получить более детальную информацию о содержимом поля. Любая расшифровка выполняется в новом окне.

Например, узнаем, когда продавались чайники Bosch15. Для этого в сформированном отчете **Динамика продаж** выберем ячейку Bosch15 и вызовем команду **Расшифровать**. В появившемся окне выбора поля выберем поле **Период**.

🏠 ← → ☆ **Динамика продаж (Продажи товаров по покупателям)** ×

Сформировать Выбрать вариант... Еще ▾ ?

Период: Этот год ... Группировка: ... ×

Параметры: Период: 01.01.2013 - 31.12.2013

Товар	Сумма
Покупатель	Оборот
Bosch1234	5 000,00
Попов Б.В. ИЧП	5 000,00
Bosch15	8 100,00
Расшифровать...	100,00
Открыть "Товар = Bosch15"	000,00
Отфильтровать	000,00
Упорядочить	500,00
Сгруппировать	000,00
Оформить	500,00
Копировать	300,00
Свойства	300,00

Выбор поля - Демонстрационная ко... (1С:Предприятие) ×

Выбор поля

Доступные поля

- Период
- Покупатель
- Сумма Оборот

Выбрать Отмена Еще ▾

Рис. 124. Выбор поля расшифровки отчета

Нажмем кнопку **Выбрать**. Детали расшифровки откроются в новом окне.

Параметры: Период: 01.01.2013 - 31.12.2013
 Отбор: Товар Равно "Bosch15"

Период	Сумма
	Оборот
05.03.2013 0:00:00	8 100,00
Итого	8 100,00

Рис. 125. Расшифровка отчета

Команда **Открыть "..."** открывает форму просмотра данных, показанных в ячейке. Например, окно с формой элемента списка.

Главная | Файлы | Характеристики | Цены товаров

Записать и закрыть | Записать | Создать на основании | Еще | ?

Группа: Чайники
 Код: 000000018
 Наименование: Bosch15
 Вид: Товар Услуга
 Артикул: Ч-0002
 Поставщик: ЭлектроБыт ЗАО
 Штрихкод:

Рис. 126. Форма элемента списка

Отфильтровать – позволяет отфильтровать текущий элемент структуры отчета по значению выбранного поля либо установить отбор по значению другого поля, используя команду **Дополнительно** и задав условие отбора. Например, отфильтруем строки отчета, в которых **Сумма (Оборот)** больше 50 000:

Редактирование условия отбора - Демонстр... (1С:Предприятие)

Редактирование условия отбора

Левое значение: Сумма Оборот
 Вид сравнения: Больше
 Правое значение: 10 000,00

ОК | Отмена | Еще

Рис. 127. Диалог редактирования условия отбора

Результат отображается в новом окне.

Отбор: Сумма Оборот Больше "10 000"

Товар	Сумма
Покупатель	Оборот
Веко67NE	15 000,00
Шлюзовая ООО	15 000,00
Босоножки	12 500,00
Попов Б.В. ИЧП	12 500,00
Йогурт	12 000,00
Попов Б.В. ИЧП	12 000,00
Колбаса	12 500,00
Магазин "Мясная лавка"	12 500,00
Итого	52 000,00

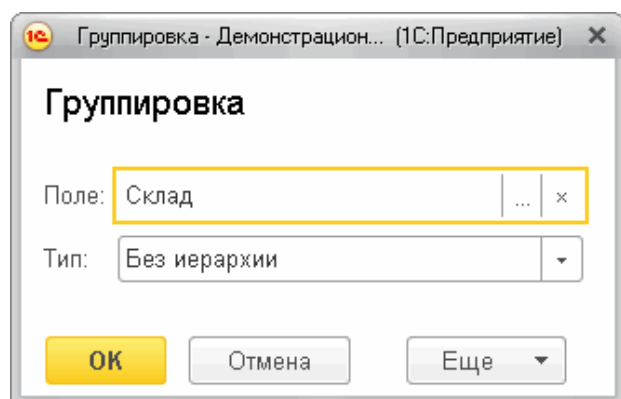
Рис. 128. Результат отбора

Упорядочить – выполняет упорядочивание содержимого текущего элемента отчета по значению текущего поля. По команде **Дополнительно** можно задать произвольное поле. Например, упорядочить таблицу в отчете по полю **Товар**.

	Товар	Сумма
	Покупатель	Оборот
[-]	Bosch1234	5 000,00
	Попов Б.В. ИЧП	5 000,00
[-]	Bosch15	17 700,00
	Попов Б.В. ИЧП	17 700,00
[-]	Sony K3456P	23 500,00
	Магазин "Продукты"	10 000,00
	Шлюзовая ООО	13 500,00
[-]	Veko345MO	10 000,00
	Магазин "Бытовая техника"	10 000,00
[-]	Veko67NE	15 000,00
	Шлюзовая ООО	15 000,00
[-]	VekoNT02	7 000,00
	Магазин "Продукты"	7 000,00
[-]	Босоножки	12 500,00

Рис. 129. Результат упорядочивания отчета

Сгруппировать – выполняет выбор поля и устанавливает тип группировки в текущем элементе структуры отчета. Например, в отчете **Остатки товаров** нужно узнать, какая обувь на каком складе находится. Для этого выберем поле группировки **Склад**.

**Рис. 130. Выбор поля группировки**

Отфильтруем по значению **Товар В Группе "Обувь"**. Результат отчета приведен ниже.

Отбор: Товар В группе "Обувь"				
Склад	Большой	Малый	Средний	Итого
Товар	Количество Остаток	Количество Остаток	Количество Остаток	Количество Остаток
Большой	180,00			180,00
Босоножки	95,00			95,00
Ботинки	12,00			12,00
Валенки	23,00			23,00
Кроссовки	20,00			20,00
Сапоги	15,00			15,00
Сланцы	2,00			2,00
Тапочки	7,00			7,00
Туфли	6,00			6,00
Малый		104,00		104,00
Ботинки		47,00		47,00
Валенки		37,00		37,00
Сапоги		6,00		6,00
Сланцы		2,00		2,00
Тапочки		10,00		10,00
Туфли		2,00		2,00
Средний			53,00	53,00
Пинетки			50,00	50,00
Сапоги			1,00	1,00
Туфли			2,00	2,00
Итого	180,00	104,00	53,00	337,00

Рис. 131. Результат группировки отчета

Оформить – позволяет указать условие применения оформления и настроить оформление для текущего элемента отчета. Например, оформить поля, где **Количество Остаток** < 10 единиц.

Отбор: Товар В группе "Обувь"				
Склад	Большой	Малый	Средний	Итого
Товар	Количество Остаток	Количество Остаток	Количество Остаток	Количество Остаток
Большой	180,00			180,00
Босоножки	95,00			95,00
Ботинки	12,00			12,00
Валенки	23,00			23,00
Кроссовки	20,00			20,00
Сапоги	15,00			15,00
Сланцы	2,00			2,00
Тапочки	7,00			7,00
Туфли	6,00			6,00
Малый		104,00		104,00
Ботинки		47,00		47,00
Валенки		37,00		37,00
Сапоги		6,00		6,00
Сланцы		2,00		2,00
Тапочки		10,00		10,00
Туфли		2,00		2,00
Средний			53,00	53,00
Пинетки			50,00	50,00
Сапоги			1,00	1,00
Туфли			2,00	2,00
Итого	180,00	104,00	53,00	337,00

Рис. 132. Результат оформления отчета

Глава 6. Изменение варианта отчета

Изменять вариант настройки отчета рекомендуется только опытным пользователям, знакомым с настройкой отчетов в системе компоновки данных. Для того чтобы отредактировать вариант отчета, следует использовать команду [Еще – Изменить вариант...](#) При этом откроется форма варианта отчета.

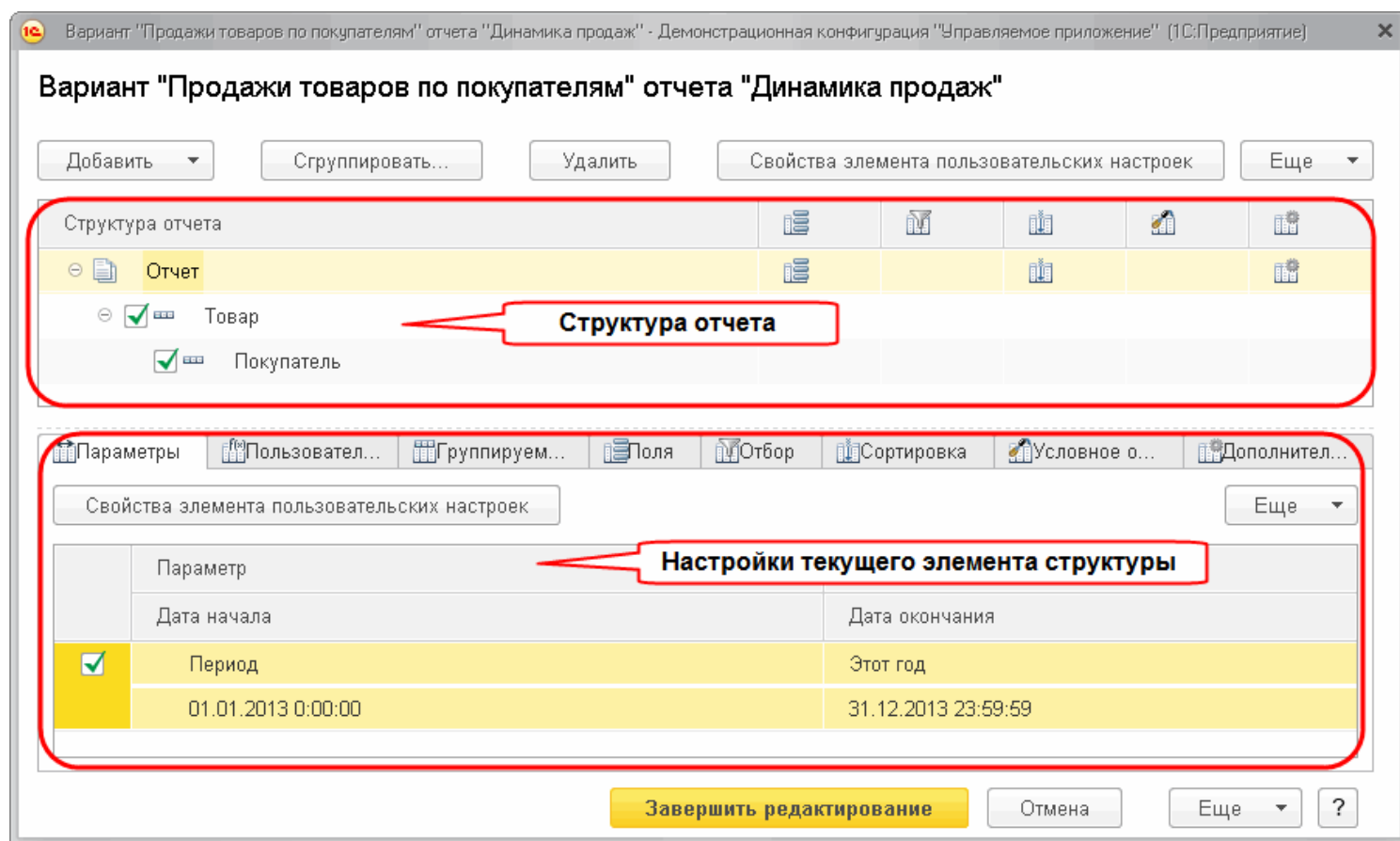


Рис. 133. Форма настройки варианта отчета

Настройка варианта отчета заключается в изменении структуры отчета и установке необходимых параметров, создании пользовательских полей, установке полей группировки, отбора, сортировки для каждого элемента отчета.

В любой момент измененный вариант настройки отчета можно вернуть в первоначальный вид, используя команду [Еще – Стандартные настройки](#) командной панели формы варианта отчета.

Команда [Еще – Сохранить настройки](#) позволяет сохранить настройки отчета в файл.

Настройки из файла могут быть загружены командой [Еще – Загрузить настройки](#). Настройки загружаются с учетом имеющихся прав доступа у текущего пользователя.

Команда [Изменить форму](#) меню [Еще](#) позволяет изменять саму форму настройки варианта отчета. Подробнее о настройке формы написано в разделе «Настройка формы» [здесь](#).

6.1. Настройка отчета в целом

Для того чтобы установить настройки для всего отчета, следует выбрать узел [Отчет](#) в структуре отчета, как это сделано на [рис. 134](#). Состав настроек (и соответствующих им закладок) зависит от наличия доступных полей и выбранного элемента настройки.

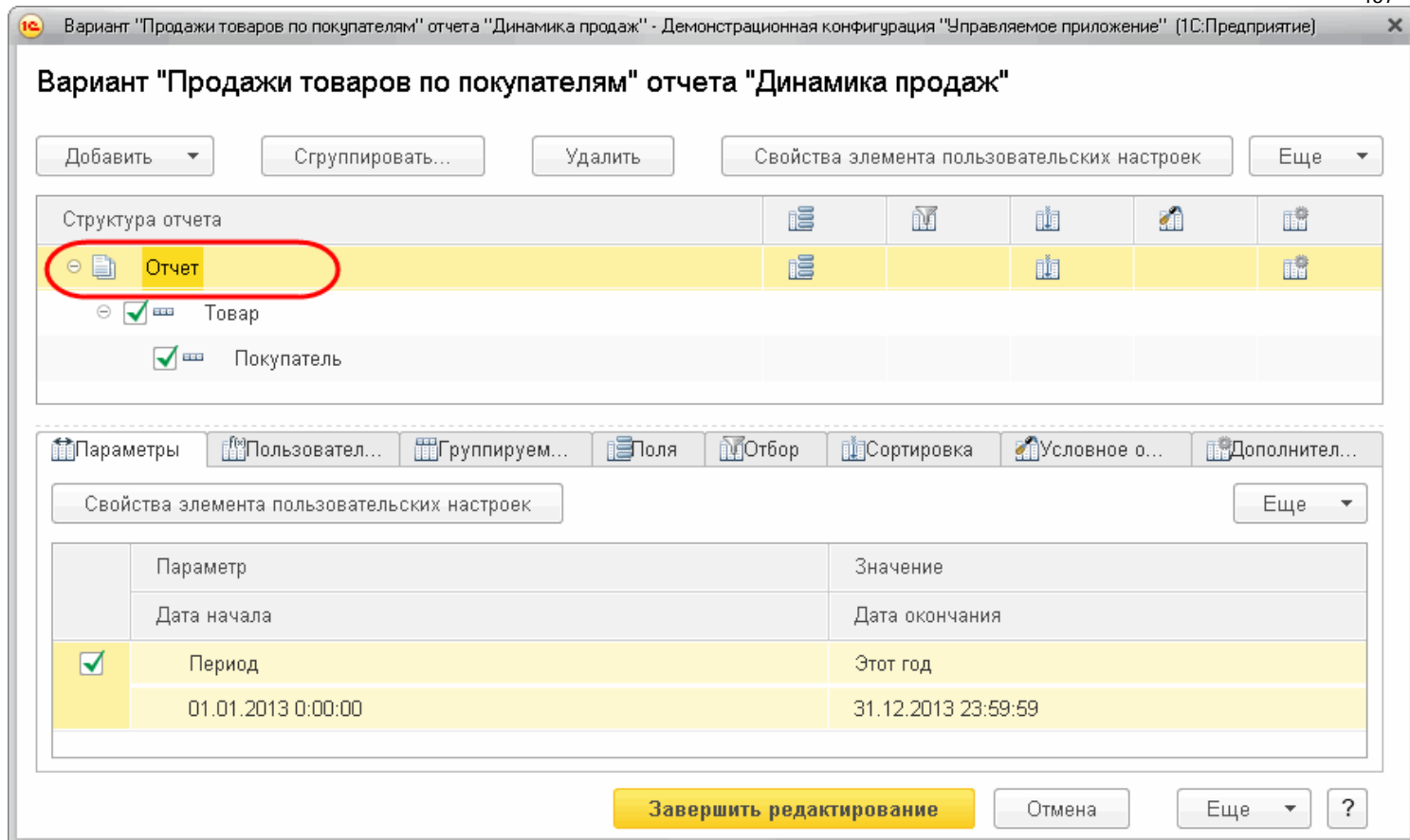


Рис. 134. Настройка варианта отчета

6.2. Изменение структуры отчета

Элементами отчета в системе компоновки данных могут быть группировка, таблица, диаграмма, а также вложенный отчет.

Группировка – выводит данные в виде списка. Чтобы добавить группировку в отчет, нужно выбрать пункт **Добавить – Новая группировка** или нажать клавишу **Insert**. В появившемся окне выбрать поле и тип группировки, как показано на рисунке.

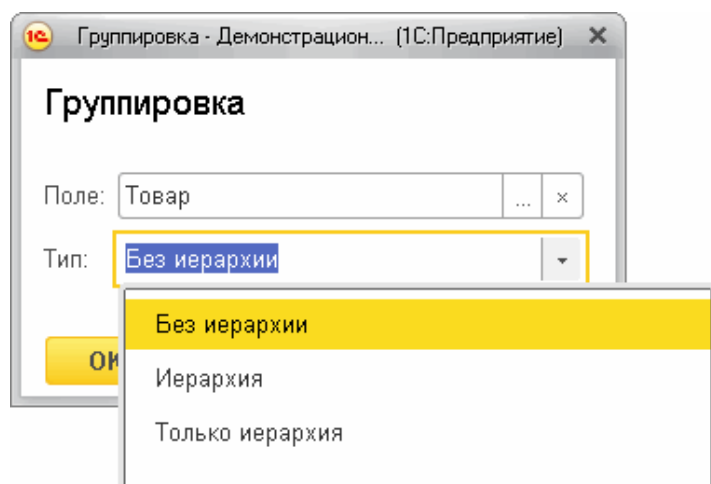


Рис. 135. Форма редактирования группировки

Если поле группировки не выбрано, то в отчет будут добавлены детальные записи – записи, полученные из базы данных в том порядке, в каком они расположены в базе.

В простейшем случае после добавления группировки по полю **Товар** отчет примет вид, как на рисунке ниже.

Структура отчета

Отчет

Товар

Товар	Количество Остаток
Босоножки	95,00
Ботинки	59,00
Валенки	60,00
Кроссовки	20,00
Пинетки	50,00
Сапоги	22,00
Сланцы	4,00
Тапочки	17,00
Туфли	10,00
Итого	337,00

Рис. 136. Группировка в отчете

СОВЕТ. Также можно выбрать элементы структуры, которые требуется сгруппировать, и выбрать команду [Еще – Сгруппировать](#).

Команда [Разгруппировать](#) удаляет группировку, оставляя ее содержимое.

[Таблица](#) – выводит данные в виде таблицы. Чтобы добавить таблицу в структуру, нужно выбрать пункт [Добавить – Новая таблица](#). В качестве строк и колонок таблицы используются группировки.

Например, в структуре отчета может быть создана таблица, строками которой является список товаров, а в колонках отображаются данные о количестве товаров на складе (см. рисунок ниже).

Структура отчета

Отчет

Таблица

Строки

Товар

Колонки

Склад

Товар	Большой Количество Остаток	Малый Количество Остаток	Средний Количество Остаток	Итого Количество Остаток
Босоножки	95,00			95,00
Ботинки	12,00	47,00		59,00
Валенки	23,00	37,00		60,00
Кроссовки	20,00			20,00
Пинетки			50,00	50,00
Сапоги	15,00	6,00	1,00	22,00
Сланцы	2,00	2,00		4,00
Тапочки	7,00	10,00		17,00
Туфли	6,00	2,00	2,00	10,00
Итого	180,00	104,00	53,00	337,00

Рис. 137. Таблица в структуре отчета

[Диаграмма](#) – отображает данные в виде диаграммы. Чтобы добавить диаграмму в структуру, нужно выбрать

пункт [Добавить – Новая диаграмма](#). В зависимости от типа диаграммы необходимо добавить точки и серии. В качестве значений диаграммы будут использованы выбранные ресурсы.

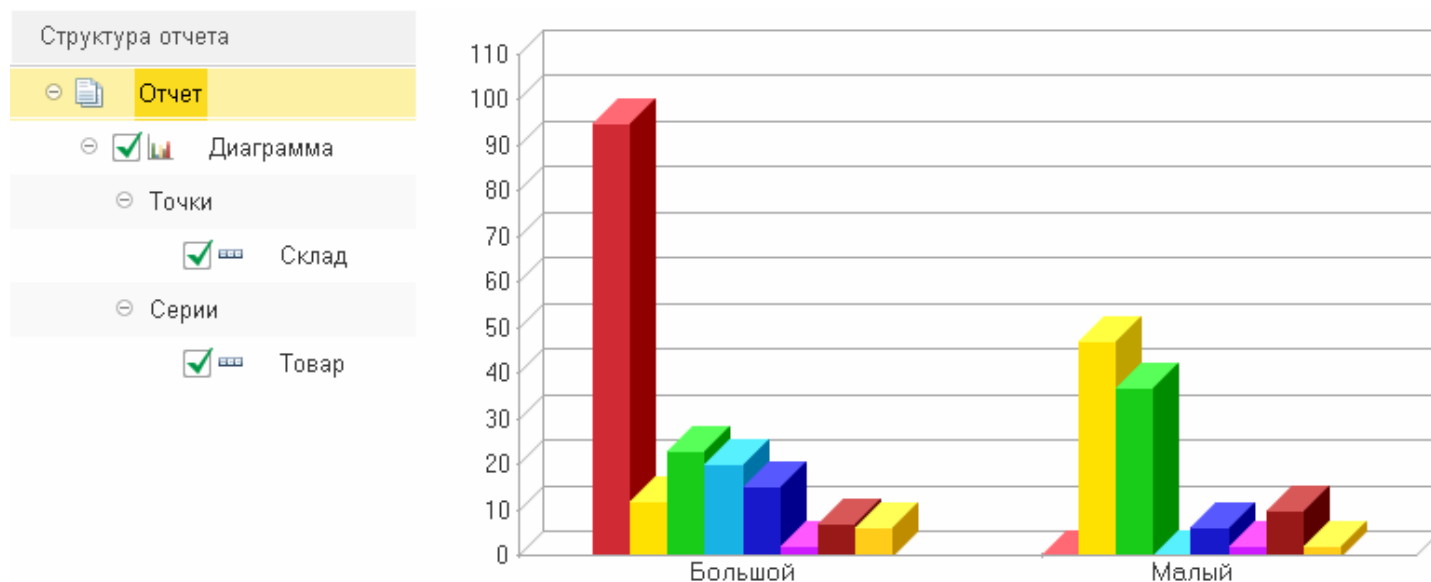


Рис. 138. Диаграмма в структуре отчета

Вложенный отчет – если при создании отчета на этапе конфигурирования для него был создан вложенный отчет, то он также может быть использован в структуре отчета в качестве элемента.

СОВЕТ. Наиболее часто используемые команды для работы со структурой отчета также можно вызывать из контекстного меню поля [Структуры](#).

Чтобы удалить элемент из структуры, нужно выбрать его в поле структуры и нажать кнопку [Удалить элемент](#) или клавишу [Delete](#).

6.2.1. Настройка элемента структуры отчета

Для установки настроек для конкретного элемента структуры отчета нужно выбрать его в структуре и на закладке настройки установить флажок [Текущий элемент имеет собственные настройки <имя настройки>](#) (для [детальной настройки элемента](#)).

Следует отметить, что возможности настройки зависят от типа элемента:

- для элемента типа [Группировка](#) можно настроить: [Поля группировки](#), [Выбранные поля](#), [Отбор](#), [Сортировка](#), [Условное оформление](#), [Другие настройки](#);
- для элементов типа [Таблица](#) и [Диаграмма](#) настраиваются: [Выбранные поля](#), [Условное оформление](#), [Другие настройки](#).

ПРИМЕЧАНИЕ. Если в настройках отчета доступен вложенный отчет, то следует учитывать, что настройки отчетов разделяются.

6.2.2. Элементы настройки

6.2.2.1. Доступные поля

Список доступных полей (см. [рис. 139](#)) позволяет легко выбирать доступные поля и переносить их в списки полей группировки, выбора, отбора, сортировки с помощью стандартного механизма перетаскивания. Используя меню [Еще](#) доступных полей, можно выбрать текущее или все доступные поля для того, чтобы поместить их в соответствующую коллекцию (команда [Выбрать все](#) выберет все поля, которые в дереве доступных полей располагаются на одном уровне с текущей строкой, при этом папки не выбираются). В списке доступных полей различаются поля, поля-ресурсы и папки полей.

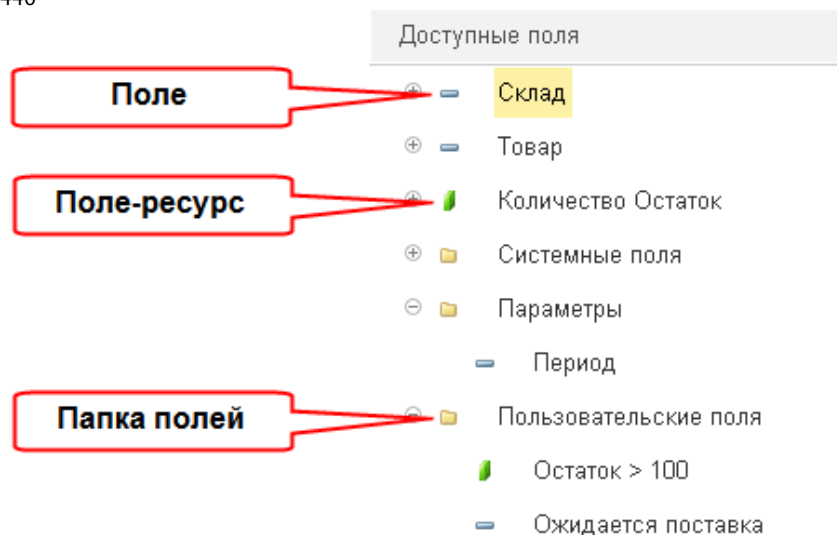


Рис. 139. Список доступных полей отчета

Для каждого числового ресурса в отчете могут быть использованы следующие вложенные поля:

- **% в строке или серии** – при выводе в таблице содержит отношение значения ресурса в текущей ячейке к значению ресурса в итоге по строке. При выводе вне таблицы выдает 100 %.
- **% в колонке или точке** – при выводе в таблице содержит отношение значения ресурса в текущей ячейке к значению ресурса в итоге по колонке. При выводе вне таблицы значение поля равно полю **% общий**.
- **% в группировке по строке или серии** – при выводе в таблице содержит отношение ресурса в текущей ячейке к значению ресурса итога по текущей группировке по строке. При выводе вне таблицы выдает 100 %.
- **% в группировке по колонке или точке** – при выводе в таблице содержит отношение ресурса в текущей ячейке к значению ресурса итога по текущей группировке по колонке, что равно полю **% в группировке**. При выводе вне таблицы равно полю **% в группировке**.
- **% в группе иерархии по строке или серии** – при выводе в таблице содержит отношение ресурса в текущей ячейке к значению ресурса итога на текущем уровне иерархии по текущей группировке по строке. При выводе вне таблицы содержит 100 %.
- **% в группе иерархии по колонке или точке** – при выводе в таблице содержит отношение ресурса в текущей ячейке к значению ресурса итога на текущем уровне иерархии по текущей группировке по колонке, что равно полю **% в группе иерархии**. При выводе вне таблицы равно полю **% в группе иерархии**.

При выводе этих полей в диаграмме в результате из заголовка поля убирается упоминание о строках и колонках. Например, поле **% в строке или серии** в результирующей диаграмме будет иметь заголовок **% в серии**.

При выводе этих полей вне диаграммы в результате из заголовка поля убирается упоминание о сериях и точках. Например, поле **% в строке или серии** в результирующей таблице будет иметь заголовок **% в строке**.

Для полей типа **Дата** доступен список дочерних полей, разбитый по группам: **Даты начала**, **Даты конца**, **Части дат**. С их помощью можно уточнить дату и время. Например, **Начало дня**, **Начало недели**, **Конец недели** и т.д.

Доступные поля

- ⊖ = Период
 - ⊖ 📁 Даты начала
 - = Начало дня
 - = Начало недели
 - = Начало декады
 - = Начало месяца
 - = Начало квартала
 - = Начало полугодия
 - = Начало года
 - = Начало минуты
 - = Начало часа
 - ⊕ 📁 Даты конца
 - ⊕ 📁 Части дат

Рис. 140. Уточнение даты

Для полей, в составе которых есть дата:

- [Начало дня](#) – Возвращает дату и время начала дня.
- [Начало недели](#) – Возвращает дату и время начала недели.
- [Начало декады](#) – Возвращает дату и время начала декады.
- [Начало месяца](#) – Возвращает дату и время начала месяца.
- [Начало квартала](#) – Возвращает дату и время начала квартала.
- [Начало полугодия](#) – Возвращает дату и время начала полугодия.
- [Начало года](#) – Возвращает дату и время начала года.
- [Конец дня](#) – Возвращает дату и время конца дня.
- [Конец недели](#) – Возвращает дату и время конца недели.
- [Конец декады](#) – Возвращает дату и время декады.
- [Конец месяца](#) – Возвращает дату и время месяца.
- [Конец квартала](#) – Возвращает дату и время квартала.
- [Конец полугодия](#) – Возвращает дату и время полугодия.
- [Конец года](#) – Возвращает дату и время конца года.
- [День](#) – Возвращает номер дня месяца.
- [День недели](#) – Возвращает номер дня недели (неделя начинается с понедельника).
- [Название дня недели](#) – Возвращает представление дня недели.
- [День года](#) – Возвращает номер дня года.
- [Неделя года](#) – Возвращает номер недели года.

- **Месяц** – Возвращает номер месяца года.
- **Название месяца** – Возвращает представление месяца.
- **Квартал** – Возвращает номер квартала.
- **Год** – Возвращает четырехзначное значение года.

Для дат, в составе которых указано время:

- **Начало минуты** – Возвращает дату и время начала минуты.
- **Начало часа** – Возвращает дату и время начала часа.
- **Конец минуты** – Возвращает дату и время конца минуты.
- **Конец часа** – Возвращает дату и время конца часа.
- **Минута** – Возвращает номер минуты.
- **Час** – Возвращает номер часа.

6.2.2.2. Параметры

Если у отчета имеются параметры, доступные для настройки, они отобразятся в списке доступных полей, в папке **Параметры**, и также станут доступны для включения/исключения из отчета и выбора значения (если параметру было задано несколько значений) на закладке **Параметры**. Например, если используется период, за который выполняется отчет, значение периода должно быть установлено пользователем.

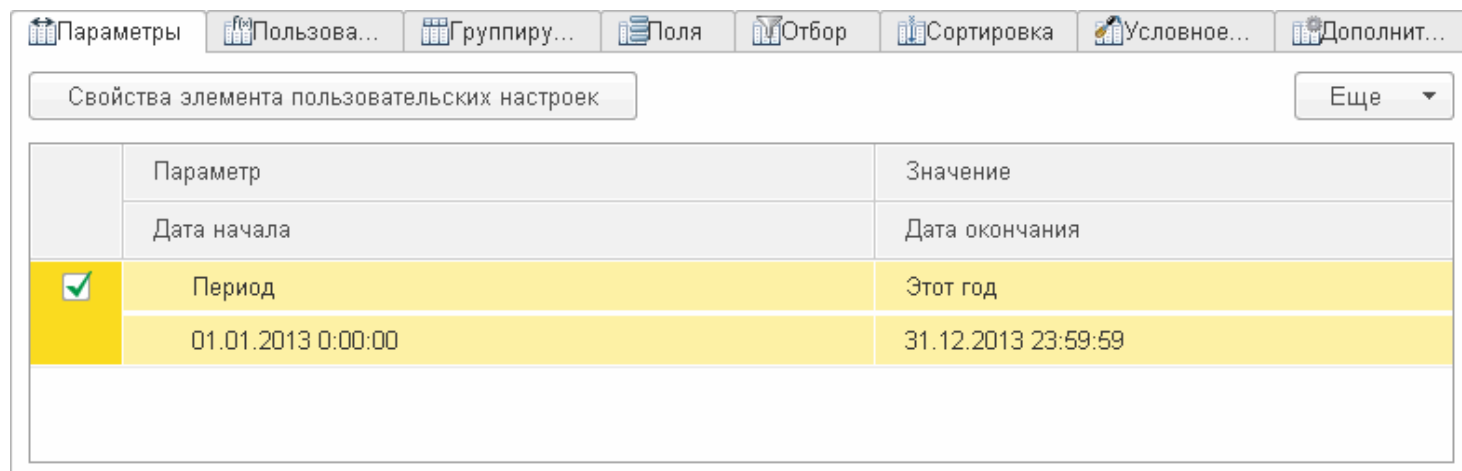


Рис. 141. Параметры отчета в форме настройки варианта

В различных настройках отчета параметры можно использовать как поля. В настройках, где будут использоваться эти поля, при выполнении отчета будут подставляться значения параметров.

Например, если в отчете есть параметр **Организация**, то в отборе будет доступно поле **Параметры.Организация**. При выборе этого поля в качестве значения отбора будет установлено значение параметра **Организация**.

Для параметра типа **Дата** в качестве значения можно использовать не только конкретную дату, но и одну из стандартных дат: начало этого дня, начало следующего дня и т. д.

6.2.2.3. Пользовательские поля

На закладке **Пользовательские поля** можно создавать поля для использования в отчете. Для того чтобы вывести пользовательское поле в отчет, нужно добавить его в список выбранных полей. При этом возможность расположения пользовательского поля в элементе отчета определяется системой автоматически.

Пользовательские поля могут быть двух видов: поле-выбор или поле-выражение.

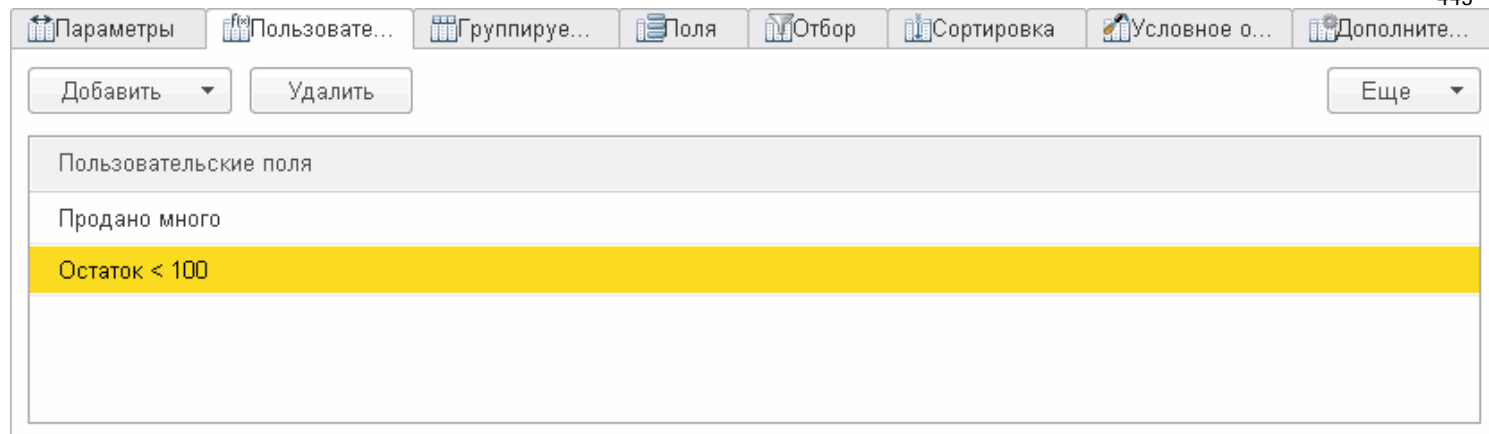


Рис. 142. Пользовательские поля в настройках варианта отчета

Полю-выбору можно задать несколько значений. Для вывода в отчет будет выбрано первое значение, для которого выполнится условие отбора. Нужно задать заголовок и выражение отбора, значение и, при необходимости, краткое представление отбора.

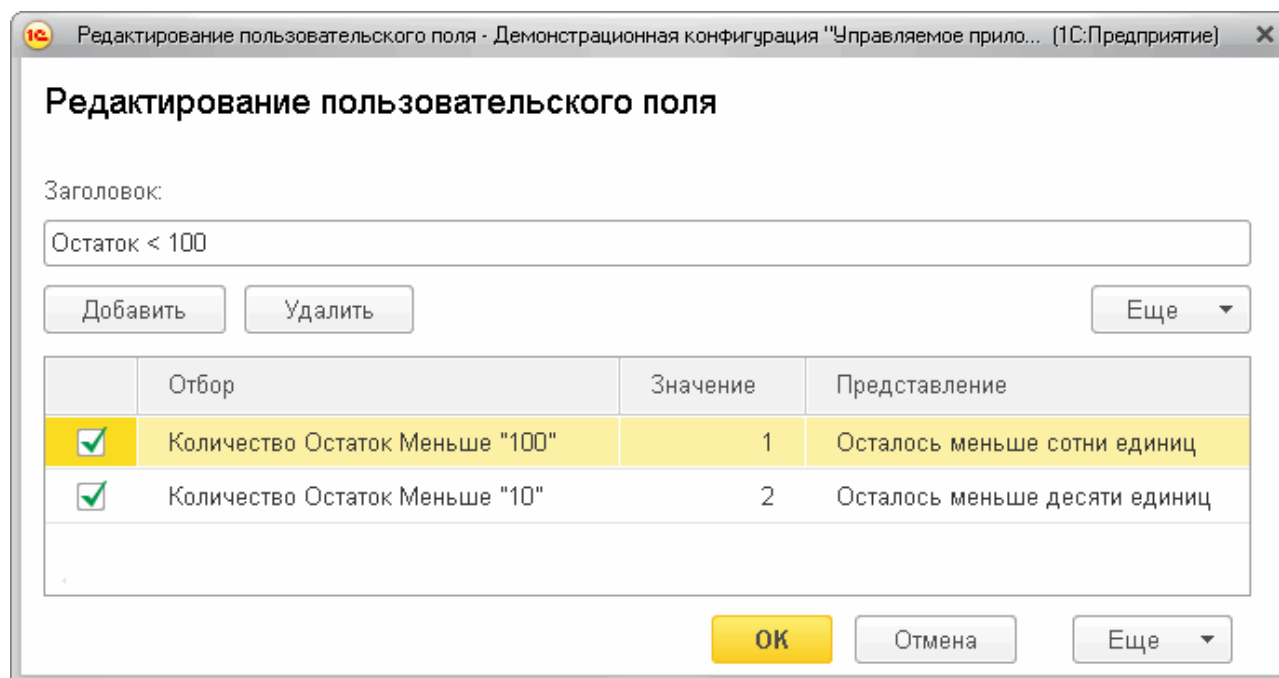


Рис. 143. Настройка отчета. Пользовательское поле-выбор

Условия отбора задаются в отдельном окне и добавляются нажатием кнопки [Добавить](#). После этого следует выбрать поле из списка доступных, вид сравнения и установить значение для сравнения.

Для того чтобы настроить последовательность элементов отбора, нужно использовать стрелки командной панели или же переместить элемент на требуемую строку с помощью мыши.

Пользовательское **поле-выражение** будет выводиться в отчете как результирующее значение выражений. Выражения составляются при помощи доступных полей отчета и выражений языка системы компоновки данных. Пользовательские поля могут использоваться в любом месте отчета. Пользовательские поля-ресурсы не могут использоваться в качестве поля группировки.

В диалоге [Редактирование пользовательского поля](#) следует задать заголовок и выражения для детальных и итоговых записей.

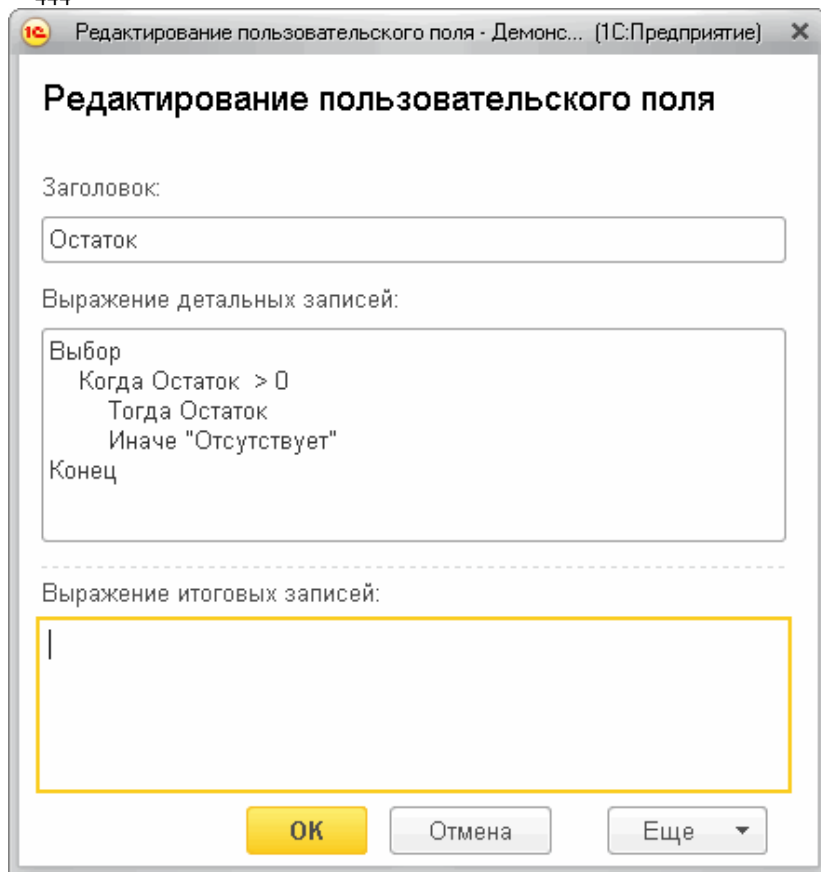


Рис. 144. Пользовательское поле-выражение

Названия полей, содержащие пробелы и/или спецсимволы, должны заключаться в квадратные скобки [].

Выражение для итоговых записей должно быть записано с помощью агрегатных функций, например: [Сумма \(Поле\) * 2](#).

Подробнее о языке выражений компоновки данных см. Приложение 1. «Язык выражений системы компоновки данных», доступное в электронной версии документации.

6.2.2.4. Группируемые поля

На закладке [Группируемые поля](#) выбираются поля, по которым будет проводиться группировка, и тип группировки.

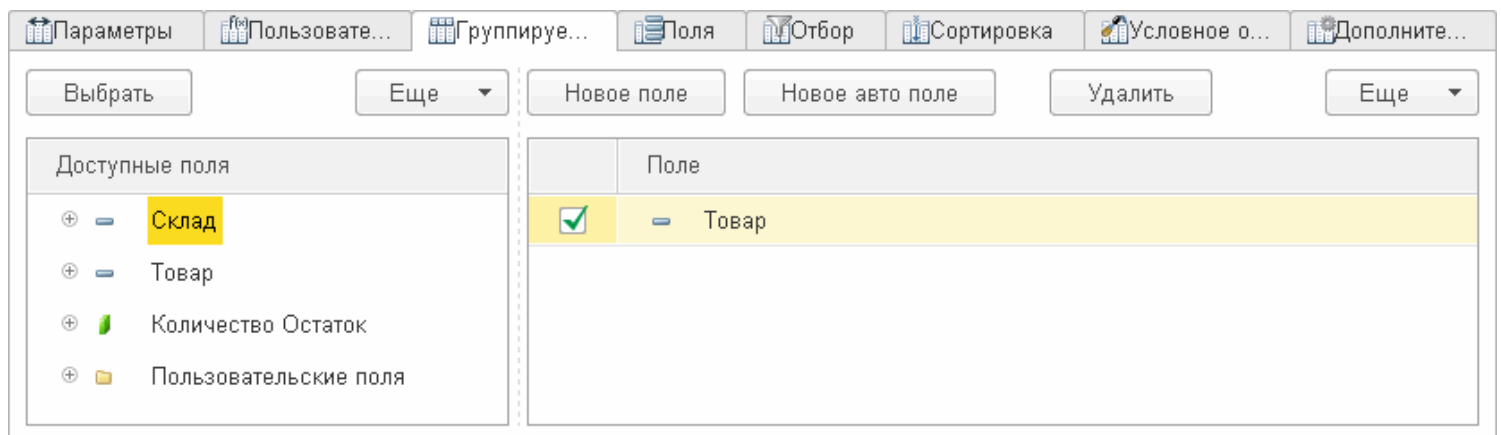


Рис. 145. Группируемые поля в элементе отчета

Группировка может иметь один из следующих типов:

- [Без иерархии](#) – в группировку попадут только неиерархические записи (элементы);
- [Иерархия](#) – в группировку попадут иерархические записи и неиерархические записи (элементы и группы);
- [Только иерархия](#) – в группировке будут выводиться только иерархические записи (группы).

С помощью меню можно добавить новое поле или новое автополе.

Автополе группировки. Группируемое **автополе** представляет собой набор полей, формируемый системой по следующему принципу. Выбираются используемые в отчете выбранные поля, которые:

- доступны для использования в полях группировки,
- не являются ресурсами,
- не являются реквизитами других выбранных полей,
- не являются реквизитами существующих полей группировки.

Чтобы просмотреть набор полей, в который будет преобразовано автополе группировки при выполнении отчета, в настройках отчета нужно вызвать команду [Еще – Развернуть](#).

Если поле уже включено в данные поля группировки, повторно оно не добавляется.

Если допустимые типы поля группировки включают в себя дату, доступна настройка дополнения периода (для вывода в отчет дат, которые не попали в результат). Например, чтобы в отчете выводились данные для всех недель, выбирается тип дополнения [Неделя](#) и устанавливается начальная и конечная даты периода, как показано на рисунке ниже.

	Поле	Тип дополнения
	Начальная дата периода	Конечная дата периода
<input checked="" type="checkbox"/>	- Период	Неделя
	14.01.2013 0:00:00	24.04.2013 0:00:00

Рис. 146. Настройка дополнения группировки по периоду

Проиллюстрируем на примере. Предположим, в отчете [Взаиморасчеты](#) отображаются данные о взаиморасчетах с магазинами «Мясная лавка» и «Продукты».

Отбор: Контрагент В списке "Магазин "Продукты"; Магазин "Мясная лавка""

Контрагент	Сумма Начальный остаток	Сумма Конечный остаток
Период, месяц		
Магазин "Мясная лавка"		315 455,43
01.08.2012 0:00:00		312 142,86
01.09.2012 0:00:00	312 142,86	315 455,43
Магазин "Продукты"		515 250,00
01.05.2012 0:00:00		45 000,00
01.08.2012 0:00:00	45 000,00	503 742,86
01.09.2012 0:00:00	503 742,86	515 250,00
Итого		830 705,43

Рис. 147. Отчет без дополнения

При этом необходимо вывести данные поквартально. Для этого группировке [Период, месяц](#) на закладке [Группировка](#) укажем [Дополнение](#).

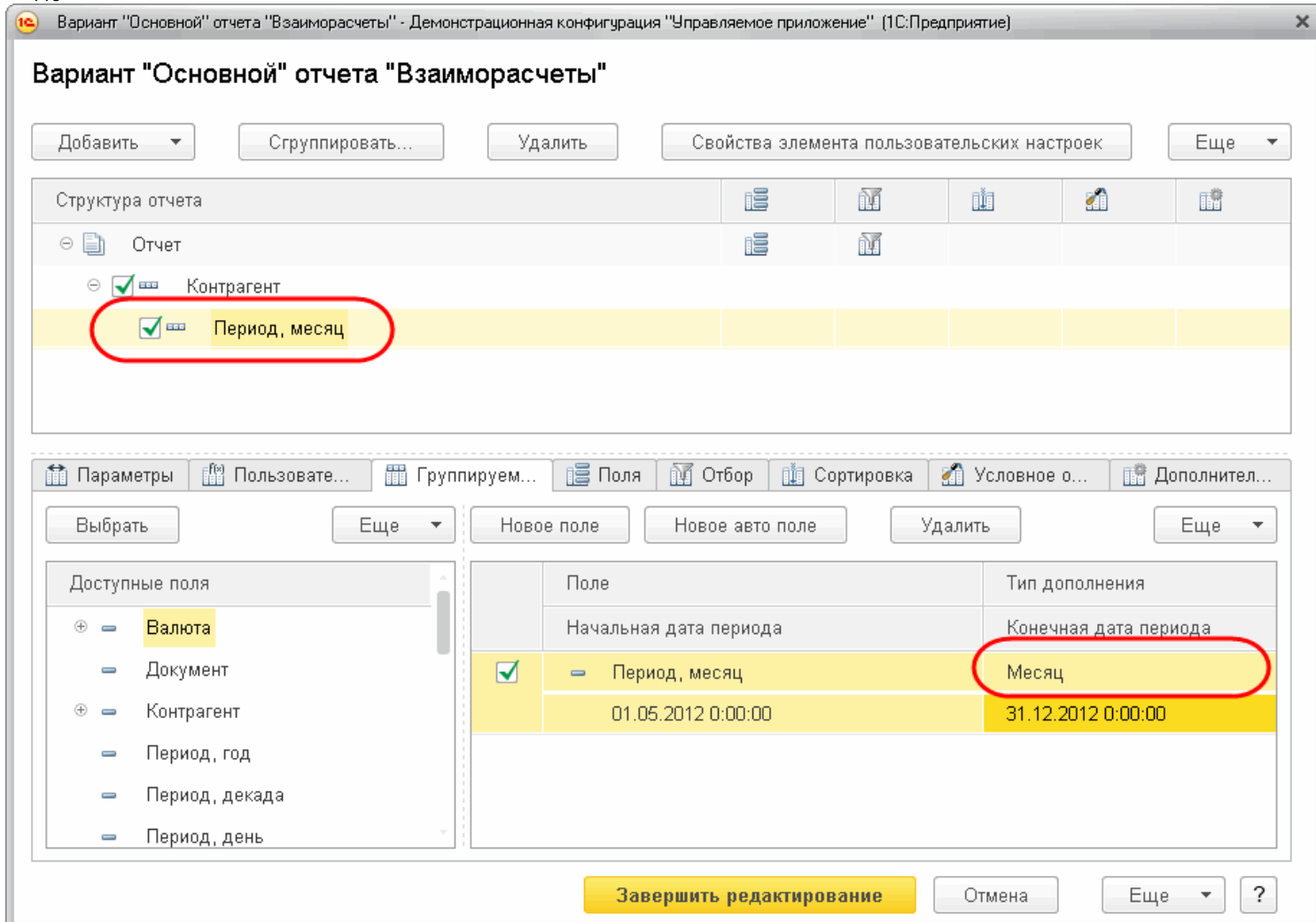


Рис. 148. Настройка дополнения в отчете

В результате отчет [Взаиморасчеты](#) примет вид:

Отбор: Контрагент В списке "Магазин "Продукты"; Магазин "Мясная лавка""

Контрагент	Сумма Начальный остаток	Сумма Конечный остаток
Магазин "Мясная лавка"		315 455,43
01.05.2012 0:00:00		
01.06.2012 0:00:00		
01.07.2012 0:00:00		
01.08.2012 0:00:00		312 142,86
01.09.2012 0:00:00	312 142,86	315 455,43
01.10.2012 0:00:00	315 455,43	315 455,43
01.11.2012 0:00:00	315 455,43	315 455,43
01.12.2012 0:00:00	315 455,43	315 455,43
Магазин "Продукты"		515 250,00
01.05.2012 0:00:00		45 000,00
01.06.2012 0:00:00	45 000,00	45 000,00
01.07.2012 0:00:00	45 000,00	45 000,00
01.08.2012 0:00:00	45 000,00	503 742,86
01.09.2012 0:00:00	503 742,86	515 250,00
01.10.2012 0:00:00	515 250,00	515 250,00
01.11.2012 0:00:00	515 250,00	515 250,00
01.12.2012 0:00:00	515 250,00	515 250,00
Итого		830 705,43

Рис. 149. Отчет с дополнением

6.2.2.5. Поля

На закладке **Поля** выбираются поля, которые будут отображены в отчете. Если выбранных полей нет, в отчет выводится пустой элемент. Имеется возможность с помощью меню или контекстного меню добавить новое поле, новую группу полей, новое автополе. Группе полей можно дать свое название и также можно управлять их размещением в текущем элементе отчета.

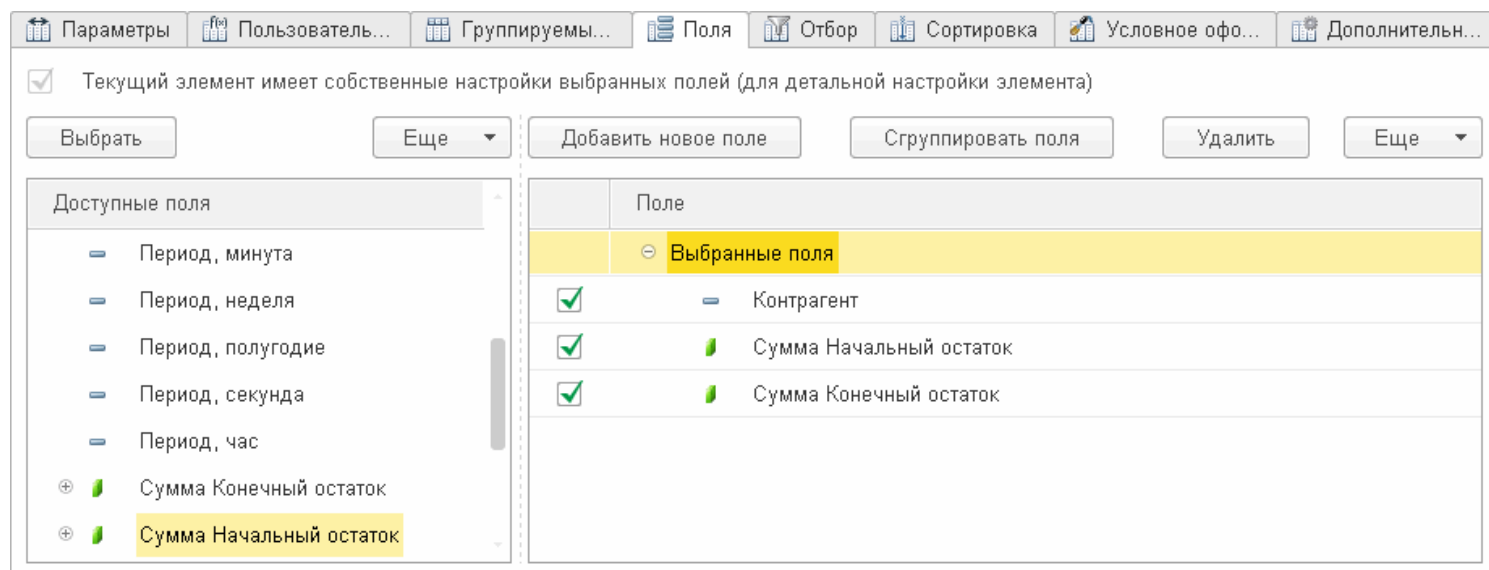


Рис. 150. Настройка отчета. Поля

6.2.2.6. Автополе выбора

В процессе настройки система позволяет использовать автоматические наборы полей в качестве полей. При создании элементов структуры отчета система автоматически добавляет поле **<Авто>** в качестве поля выбора. **Автополе выбора** представляет собой набор полей, в который оно будет преобразовано при выполнении отчета. Чтобы просмотреть набор полей, используется команда **Еще – Развернуть**. Также если нужно добавить автополе, следует использовать кнопку **Новое авто поле**.

Состав набора полей зависит от того, какому элементу структуры принадлежит разворачиваемое автополе и в какой части структуры этот элемент располагается. Для каждого элемента система обходит все родительские элементы структуры отчета и из выбранных полей этих элементов отбирает ресурсы и поля по следующим правилам:

- Для **группировки** и **группировки таблицы** на место автополя подставляются все используемые поля этой группировки, которые доступны для использования в выбранных полях; поля, которые являются реквизитами ее полей группировки, и ресурсы родительских элементов.

ВНИМАНИЕ! Система учитывает при обходе только группировки с типом **Без иерархии** или **Иерархия**.

- Для **группировки диаграммы** ресурсы не выбираются, а обходятся все родительские элементы структуры настроек, и из выбранных полей этих элементов выбираются поля группировок, если по данному полю была задана группировка с типом **Только иерархия**.

- Для **группировок типа Детальные записи (группировка, группировка таблицы, группировка диаграммы)** из основных выбранных полей настроек, которым принадлежит группировка, выбираются все используемые поля, кроме полей, участвовавших в вышестоящих группировках, и реквизитов этих полей. Если же такая группировка имеет тип **Только иерархия**, то ее поля и реквизиты будут использоваться системой при формировании набора полей выбора. Для группировки диаграммы ресурсы также не выбираются.

- Для **диаграммы** автополе выбора заменяется всеми ресурсами, указанными для диаграммы. В зависимости от настроек прикладного решения автополе выбора может заменяться ресурсом, первым из встреченных при описанном выше обходе.

- Для **таблицы** автополе выбора преобразовывается в набор ресурсов, используемых родительскими элементами.

ПРИМЕЧАНИЕ. Если поле уже включено в данные выбранные поля, повторно оно не добавляется.

При этом поля добавляются в набор в следующем порядке: вначале поля собственных полей группировки (для группировок), потом поля из глобальных настроек (для группировок типа [Детальные записи](#)), и самыми последними – ресурсы и поля из родительских элементов.

Выбранные поля можно объединять в группы, которые добавляются командой [Сгруппировать](#) (чтобы команда была доступна, все выделенные строки таблицы должны иметь одного родителя) и удаляются, сохраняя вложенные поля, командой [Разгруппировать](#). Для группы можно задавать расположение внутри текущего элемента структуры отчета. Для этого следует выбрать необходимый вид расположения из списка в колонке [Расположение](#).

6.2.2.7. Отбор

На закладке [Отбор](#) выбираются поля для фильтрации записей результата отчета. С помощью меню, контекстного меню или двойным щелчком мыши по выбранному полю из доступных полей можно добавить элемент или группу элементов. На этой странице можно выбрать отбор из заданных разработчиком по представлению, если таковое устанавливалось для отчета, или добавить другой.

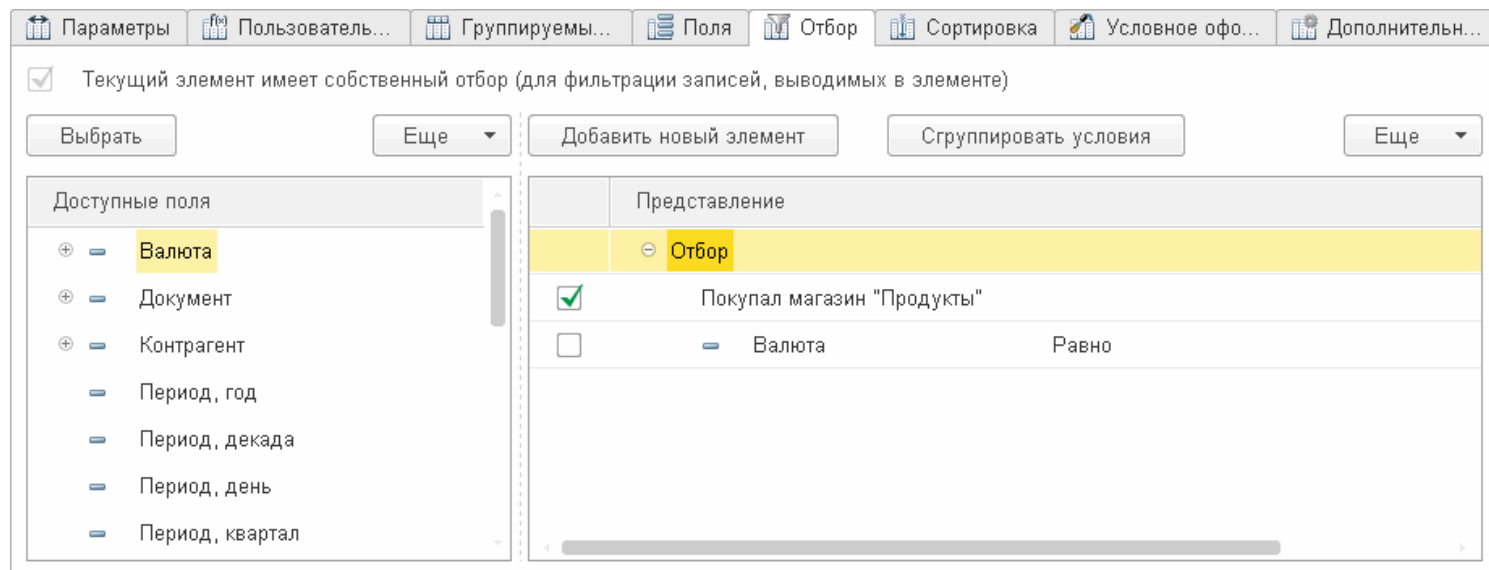


Рис. 151. Настройка элемента отчета. Представление отбора

При вызове команды [Еще – Подробно](#) на закладке отобразятся условия отбора вместе с представлением.

	Поле	Вид срав...	Значение	Режим отображ...	Представление
⊖ Отбор					
<input checked="" type="checkbox"/>	Контрагент	Равно	Магазин "Продукты"	Быстрый доступ	Покупал магазин "Продукты"
<input type="checkbox"/>	Валюта	Равно		Быстрый доступ	

Рис. 152. Настройка элемента отчета. Отбор

В колонках этой закладки элементам можно установить значения, по которым будет проводиться отбор записей результата отчета.

Представление. С помощью команды контекстного меню [Установить представление](#) созданному элементу или группе отбора можно задать представление, которое будет показывать информацию об элементе отбора в списке, если кнопка [Подробно](#) командной панели отжата. Если для отбора не было задано представления, то его подробное и непоборное представления на закладке [Отбор](#) не отличаются.

Применение. Для группировки, группировки в таблице, группировки в диаграмме для элементов отбора доступно еще одно свойство – [Применение](#).

	Поле	Вид срав...	Значение	Применение
⊖ Отбор				
<input checked="" type="checkbox"/>	Контрагент	Равно	Магазин "Продукты"	Обычное

Рис. 153. Настройка применения для элемента отбора

Если свойство **Применение** установлено в значение **После группировки или для иерархии**, то условие отбора проверяется после построения иерархии. Значение **Только иерархия** указывает, что условие отбора проверяется только для иерархических записей. Отфильтрованные записи продолжают влиять на итоговые значения ресурсов вышестоящих иерархических записей. Вложенные группировки записей, отфильтрованных при помощи этих свойств, также будут выводиться в отчет.

Виды сравнения. Программа поддерживает следующие виды сравнения:

- **Равно** – в результат попадут записи, для которых значение поля, указанного в качестве левого значения, равно правому значению.
- **Не равно** – в результат попадут записи, для которых значение поля, указанного в качестве левого значения, не равно правому значению.
- **Меньше** – в результат попадут записи, для которых значение поля, указанного в качестве левого значения, меньше правого значения.
- **Меньше или равно** – в результат попадут записи, для которых значение поля, указанного в качестве левого значения, меньше или равно правому значению.
- **Больше** – в результат попадут записи, для которых значение поля, указанного в качестве левого значения, больше правого значения.
- **Больше или равно** – в результат попадут записи, для которых значение поля, указанного в качестве левого значения, больше или равно правому значению.
- **В списке** – в результат попадут записи, в которых есть поля со значением, присутствующим в списке. Список значений устанавливается в отдельном окне, открываемом при выборе колонки **Левое значение**.
- **В группе из списка** – в результат попадут записи, значения которых относятся к группам, присутствующим в списке, либо записи, которые указаны в списке. Список групп и элементов устанавливается в отдельном окне, открываемом при выборе.
- **В группе** – в результирующий документ попадут записи, поле из левого значения которых находится в указанной группе элементов списка.
- **Не в списке** – в результат попадут записи, в которых нет полей со значением, присутствующим в списке. Список значений устанавливается в отдельном окне, открываемом при выборе колонки **Левое значение**.
- **Не в группе из списка** – в результирующий документ попадут записи, поле из левого значения которых находится в указанной группе элементов списка.
- **Содержит** – в результат попадут записи, содержащие подстроку, используемую в качестве значения.
- **Не содержит** – в результат попадут записи, которые не содержат подстроки, используемой в качестве значения.
- **Заполнено** – в результат попадут записи, у которых заполнено заданное поле.
- **Не заполнено** – в результат попадут записи, у которых заданное поле не заполнено.
- **Начинается с** – в результат попадут записи, у которых значение начинается на строку, заданную в условии сравнения.
- **Не начинается с** – в результат попадут записи, у которых значение не начинается на строку, заданную в условии сравнения.
- **Соответствует шаблону** – в результат попадут записи, значение которых удовлетворяет маске, заданной в условии сравнения.
- **Не соответствует шаблону** – в результат попадут записи, значение которых не удовлетворяет маске, заданной в условии сравнения.

В строке шаблонов используется следующий синтаксис:

- / - следующий символ нужно интерпретировать как обычный символ;
- % - процент: последовательность, содержащая ноль и более произвольных символов;
- _ - подчеркивание: один произвольный символ.

Объединение условий отбора. Условия отборов можно объединять в логические группы по **И**, **НЕ** и группы по **ИЛИ**. Чтобы объединить условия, нужно выбрать их и нажать кнопку **Сгруппировать условия**:

- если отборы находятся в группе по **И**, то в отчет будут выводиться данные, для которых выполняются все условия, находящиеся в группе;
- если отборы объединены в группу по **НЕ**, то в результирующий документ не попадут записи, удовлетворяющие всем условиям группы;
- если отборы находятся в группе по **ИЛИ**, то данные будут выводиться при выполнении хотя бы одного условия. С помощью контекстного меню группам можно задавать представления; если оно есть у группы, то в кратком режиме вложенные элементы группы показываться не будут.

6.2.2.8. Сортировка

На закладке **Сортировка** выбираются поля, по которым результат выполнения отчета будет отсортирован. Возможно добавление как элемента порядка, так и автоэлемента порядка. При выполнении отчета система преобразует (разворачивает) автоэлемент порядка в поля, по которым и проведет сортировку.

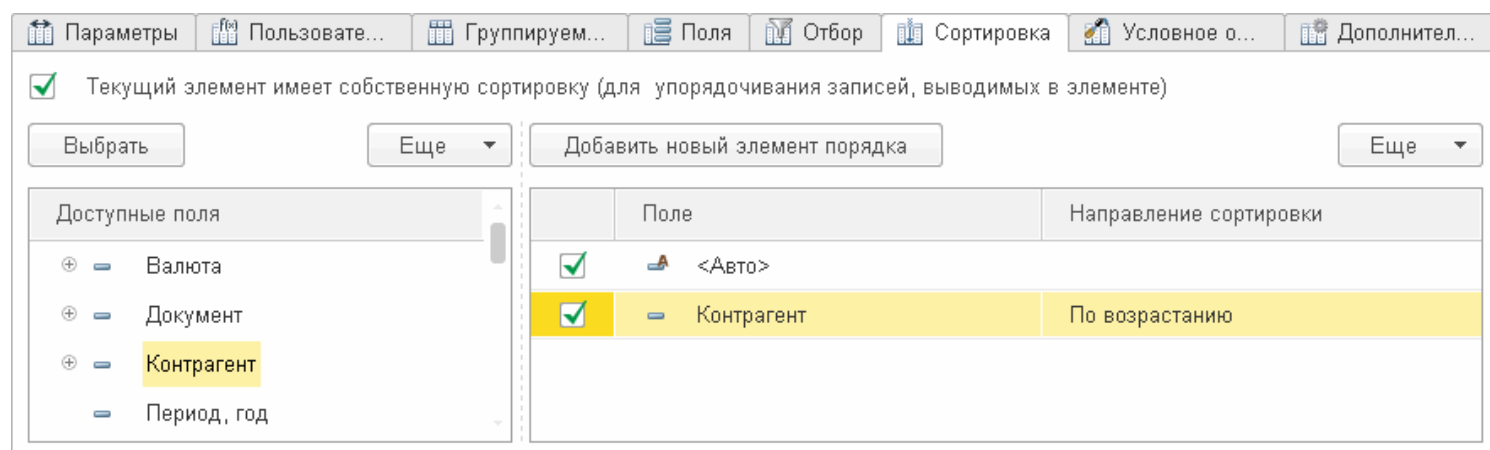


Рис. 154. Поля сортировки для элемента отчета

Для **автоэлемента** порядка в меню **Еще** есть команда **Развернуть**, которая позволяет просмотреть, в какой набор полей это поле будет преобразовано при выполнении отчета. При развороте автоэлемента (порядка) из глобального упорядочивания безусловно будут добавлены поля-ресурсы, а из полей нересурсов в порядок будут добавлены поля, являющиеся реквизитами поля группировки, и само поле группировки (для детальных записей будут занесены все поля). Поля группировки, которые не были указаны в глобальном упорядочивании, попадут в конец порядка.

6.2.2.9. Условное оформление

На закладке **Условное оформление** можно установить оформление для различных элементов отчета в зависимости от значений данных, которые выводятся в отчет. Например, выделить цветом отрицательные значения и т. д. Таким образом может быть оформлено несколько элементов структуры. При этом для каждого элемента задаются области, которые будут оформлены.

Условное оформление может состоять из нескольких элементов. Каждый элемент описывает область, которую нужно оформить.

Область представляет собой список полей, выбираемых из списка доступных. Если область не задана, то условное оформление применится ко всему элементу отчета.

Для каждой области могут задаваться условия, при выполнении которых к области будет применяться выбранное оформление.

Условия отбора полей указываются в отдельном окне, похожем на окно отбора пользовательского поля выбора. Подробнее про отборы можно прочитать [здесь](#).

Если по некоторым условиям к области применяются два разных оформления, то в итоге выбирается последнее из них в списке элементов условного оформления.

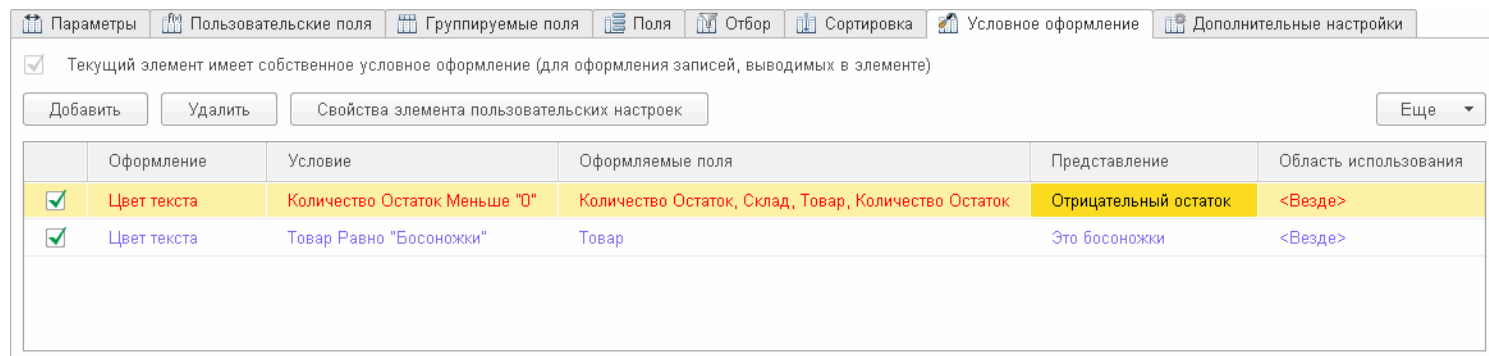


Рис. 155. Список элементов условного оформления в режиме «Подробно»

В колонке **Представление** задается обозначение условного оформления области, с которым оно будет показано в списке условных оформлений, если кнопка напротив команды **Подробно** меню **Еще** отжата.

Параметры условного оформления задаются в отдельном окне. Можно выбрать цвета фона, текста, границы ячейки, стиль границ ячейки, шрифт текста в ячейке и др.

- **Цвет фона** – устанавливает цвет фона;
- **Цвет текста** – устанавливает цвет текста для текстовых данных;
- **Цвет в диаграмме** – устанавливает цвет серии данных в диаграмме;
- **Цвет границы** – цвет, которым будет отображаться граница элемента данных;
- **Стиль границы** – стиль, которым будет отображаться граница элементов. Система позволяет настроить вид для каждой границы в отдельности;
- **Шрифт** – устанавливает шрифт, которым будут отображаться текстовые данные;
- **Отступ** – устанавливает смещение на заданное количество символов;
- **Автоотступ** – если указано значение больше 0, то при отображении группировок отступ будет определяться как произведение уровня группировки и значения, указанного в свойстве;
- **Горизонтальное положение** – в данном свойстве выбирается выравнивание текста по горизонтали;
- **Вертикальное положение** – в данном свойстве выбирается выравнивание текста по вертикали;
- **Размещение** – устанавливает способ отображения текста, если он не помещается в ячейке целиком;
- **Ориентация текста** – число, устанавливает угол наклона текста в ячейках группировок и таблиц;
- **Формат** – устанавливает формат отображения данных;
- **Выделять отрицательные** – если свойство установлено в значение **Да**, то в результате отчета отрицательные значения будут отображаться особым цветом;
- **Отметка незаполненного** – если свойство установлено в значение **Да**, то в результате отчета незаполненные значения будут отображаться особым образом;
- **Минимальная ширина** – число, устанавливает минимальное количество символов для отображения данных по ширине;
- **Максимальная ширина** – число, устанавливает максимальное количество символов для отображения данных по ширине;
- **Минимальная высота** – число, устанавливает минимальное количество символов для отображения данных по высоте;
- **Максимальная высота** – число, устанавливает максимальное количество символов для отображения данных по высоте;

- **Текст** – произвольный текст, описывающий данные.

Чтобы установить область применения элемента оформления, при двойном щелчке в колонке **Область использования** откроется закладка **Дополнительно** в диалоге редактирования элемента условного оформления. Если флажок напротив области установлен, то элемент оформления будет использован при отображении этой области.

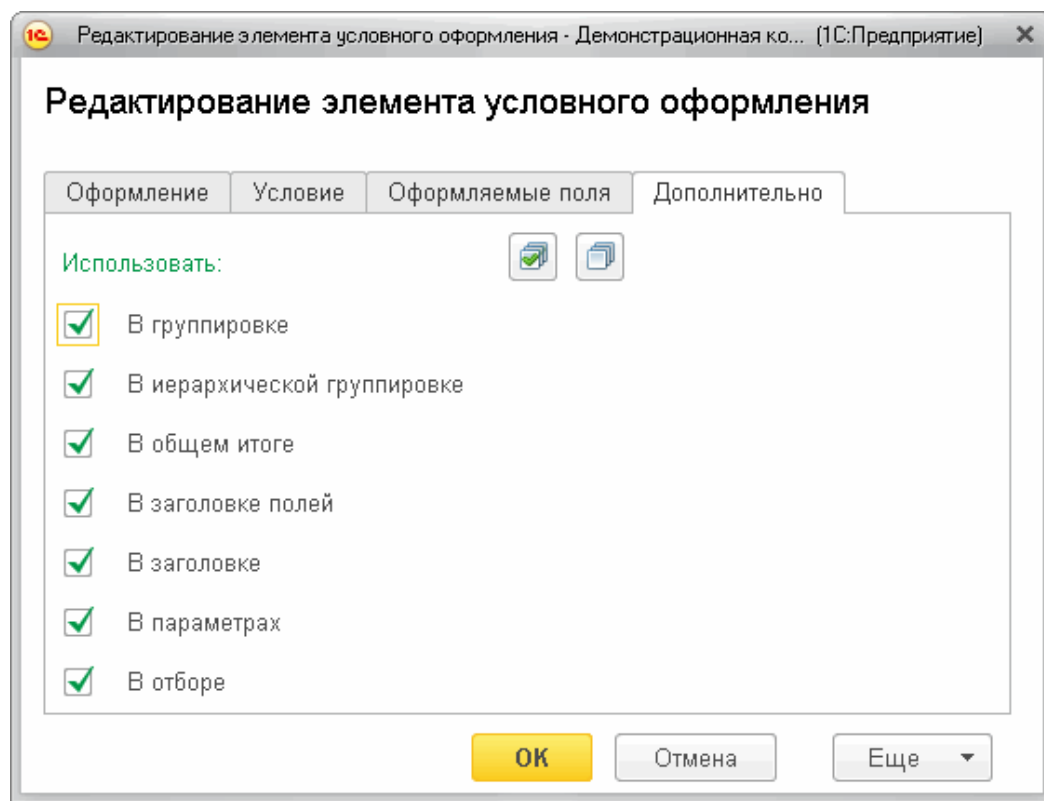


Рис. 156. Установка области использования элемента оформления

По умолчанию все флажки установлены. Для быстрого редактирования можно использовать кнопки **Установить все** и **Сбросить все**.

ПРИМЕЧАНИЕ. Возможность установки области использования элемента условного редактирования недоступна при настройке динамического списка.

6.2.2.10. Дополнительные настройки

На закладке **Дополнительные настройки** для выбранного элемента структуры задаются параметры вывода. Например, для группировки содержимое закладки будет иметь вид:

	Параметр	Значение
<input type="checkbox"/>	Макет оформления	Основной
<input type="checkbox"/>	Количество записей	
<input type="checkbox"/>	Процент записей	
<input type="checkbox"/>	Расположение итогов	Авто
<input type="checkbox"/>	Расположение полей группировок	Вместе
<input type="checkbox"/>	Расположение группировок	Начало
<input type="checkbox"/>	Расположение реквизитов	Вместе с владельцем
<input type="checkbox"/>	Расположение ресурсов	Горизонтально
<input type="checkbox"/>	Расположение общих итогов	Авто
<input type="checkbox"/>	Тип заголовка полей	Авто
<input type="checkbox"/>	Выводить заголовок	Авто
<input type="checkbox"/>	Заголовок	
<input type="checkbox"/>	Выводить отбор	Авто
<input type="checkbox"/>	Авто позиция ресурсов	После всех полей
<input type="checkbox"/>	Вариант использования группировки	Авто

Рис. 157. Дополнительные настройки элемента отчета

ПРИМЕЧАНИЕ 1. При компоновке отчета система выдаст ошибку, если во вложенном отчете используется поле верхнего отчета, значение которого невозможно определить.

ПРИМЕЧАНИЕ 2. При размещении диаграммы в структуре отчета следует иметь в виду, что при выборе вида диаграммы **Круговая**, если не заданы серии, но заданы точки, в качестве серий системой будут автоматически использоваться точки. А при выборе вида диаграммы **График**, если не заданы точки, но заданы серии, в качестве точек будут использоваться серии.

6.3. Выбор элементов настроек

Чтобы указать, что элемент настройки доступен на форме настроек отчета, в настройке элемента используется команда **Свойства элемента пользовательских настроек**.

В форме настройки пользовательского элемента можно установить признак того, что элемент является пользовательским, а также указать представление, которое будет использоваться для элемента, и режим его редактирования (обычный, быстрый доступ, недоступный), см. [рис. 158](#).

Скриншот интерфейса настройки элемента отчета. В центре экрана открыто диалоговое окно «Пользовательские настройки элемента». В диалоговом окне:

- Чекбокс «Включать в пользовательские настройки» отмечен галочкой и выделен красным овалом.
- Поле «Представление» пустое.
- Выбор «Режим редактирования» установлен на «Быстрый доступ».
- Кнопки «ОК», «Отмена» и «Еще» (с выпадающим меню) находятся в нижней части окна.

На заднем плане видна таблица с заголовками «Поле», «Вид сравнения», «Значение»:

Поле	Вид сравнения	Значение
Отбор		
<input type="checkbox"/> Товар	Равно	
<input type="checkbox"/> Склад	Равно	

Рис. 158. Добавление элемента пользовательских настроек

В списке структуры настроек команда [Свойства элемента пользовательских настроек](#) позволяет настраивать пользовательские настройки для текущего элемента структуры.

Для каждого элемента структуры определен свой состав настраиваемых элементов.

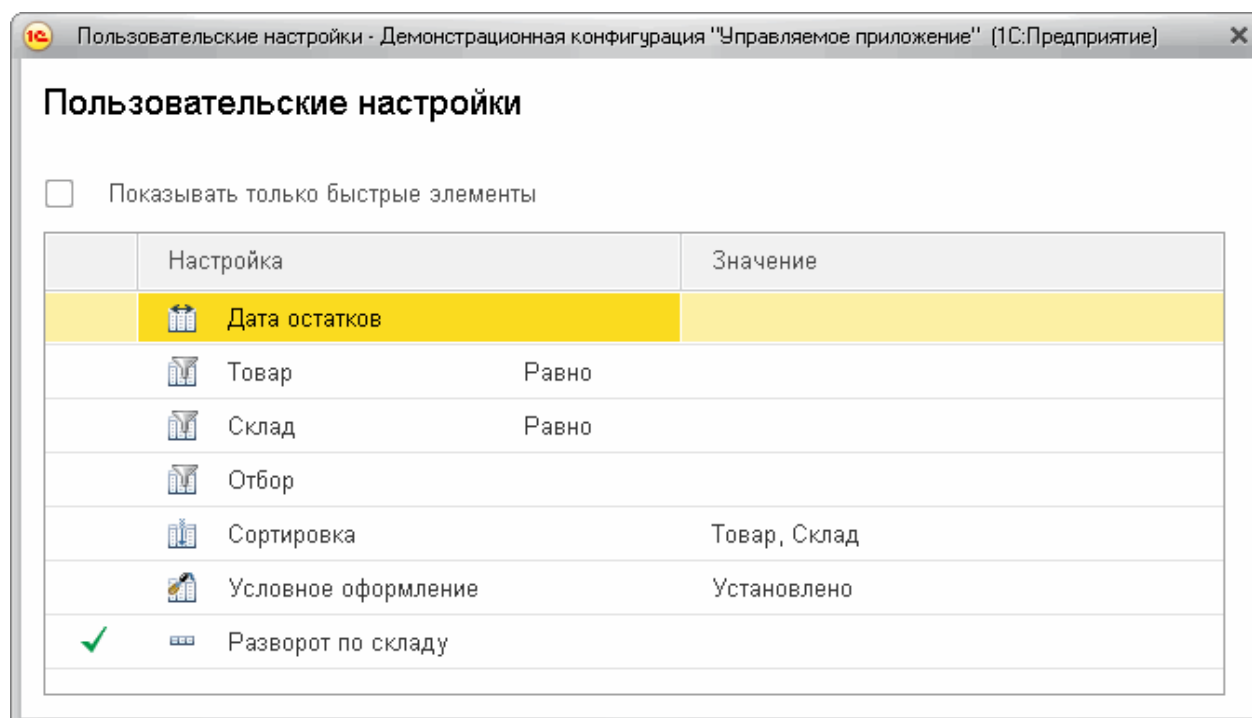
Объект	Настраиваемые элементы
Отчет	Выбранные поля, порядок, отбор, условное оформление, состав группировок
Группировка/группировка таблицы/группировка диаграммы	Группировка, выбранные поля, отбор, порядок, условное оформление, состав вложенных группировок
Диаграмма	Диаграмма, выбранные поля, условное оформление, состав группировок серий, состав группировок точек
Таблица	Таблица, выбранные поля, условное оформление, состав группировок строк, состав группировок колонок
Вложенная схема	Вложенный отчет, выбранные поля, отбор, порядок, условное оформление, состав группировок

В зависимости от того, где происходит вызов, команда [Свойства элемента пользовательских настроек](#) позволяет изменять различные настройки:

- список отбора – настройки для текущего элемента/групп отбора;
- список параметров вывода и параметров данных – настройки для текущего параметра;
- список условного оформления – настройки для текущего элемента условного оформления.

ПРИМЕЧАНИЕ. Если настройка включена в пользовательские настройки, то при выполнении отчета ее значение будет получаться из пользовательских настроек, а не из варианта отчета. При редактировании варианта отчета изменение настройки, значение которой было ранее сохранено в пользовательских настройках, не повлияет на результат отчета, поскольку значение настройки будет получено из ранее сохраненной пользовательской настройки.

Кроме того, вызванная из командной панели структуры отчета команда [Пользовательские настройки](#) позволяет открыть модальную форму (см. рисунок), в которой будут отображаться настройки со своими значениями по умолчанию.

**Рис. 159. Просмотр пользовательских настроек**

Глава 7. Сервисные возможности

Система «1С:Предприятие» предоставляет сервисные механизмы для навигации по приложению: механизм ссылок, история работы, избранное.

Также пользователю доступна возможность использования встроенных калькулятора, календаря, механизма для сравнения файлов и т. д.

Данная глава описывает работу с перечисленными механизмами.

7.1. Поиск в данных

Чтобы начать поиск данных, нужно нажать кнопку **Поиск** на панели инструментов.



Рис. 160. Кнопка поиска

Также для этого можно использовать сочетание клавиш **Ctrl + Shift + F**.

В рабочей области откроется форма поиска. Нужно ввести искомое слово или фразу в строку поиска и нажать кнопку **Найти** (или клавишу **Enter**). Для уточнения поиска используются операторы полнотекстового поиска в данных (подробнее см. главу «[Формат поисковых выражений](#)», [здесь](#)).

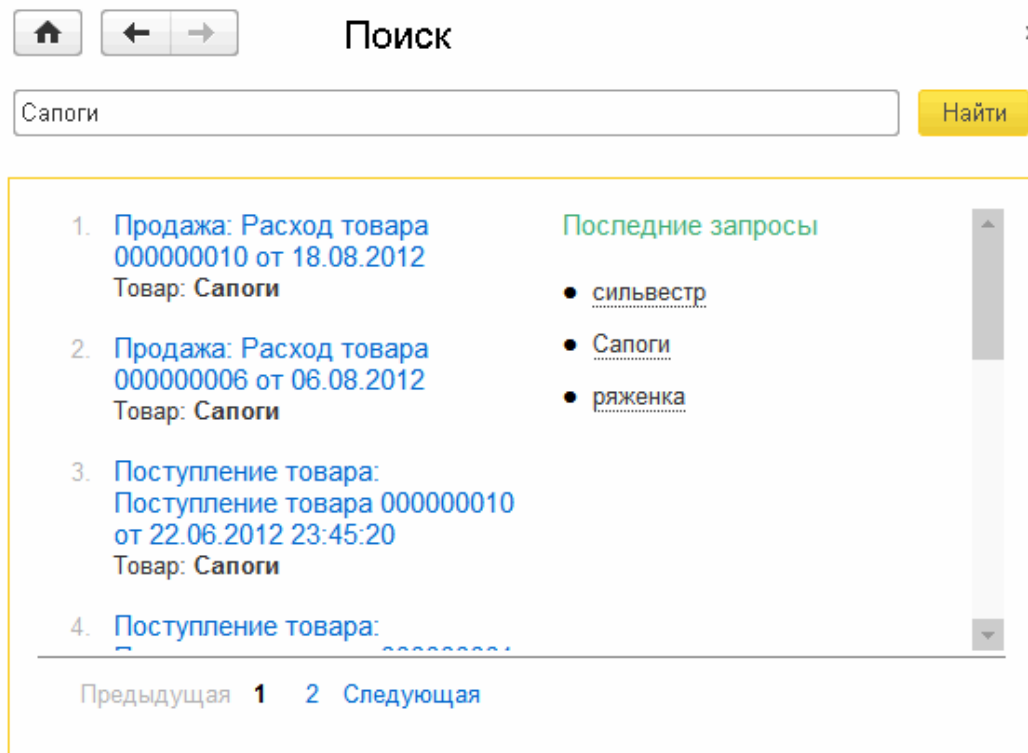


Рис. 161. Поиск данных

На форме отобразится список ссылок на объекты, содержащие указанную строку. Для более эффективной работы с поиском последние запросы отображаются отдельным списком.

Если в результате поиска объекты не были найдены, на экране отобразится подсказка по поиску.

Если в строке поиска ввести навигационную ссылку, то будет выполнен переход по ссылке.

7.2. Ссылки

В системе имеется возможность получить текстовую ссылку на любой раздел интерфейса программы, отчет, обработку и на объекты информационной базы (документы, элементы списков и т. д.).

ПРИМЕЧАНИЕ. Ссылки на разделы системой не выдаются.

Полученную ссылку можно сохранить и использовать в дальнейшем для перехода по ней. Так как ссылки являются текстовыми, то их можно отправлять другим пользователям, например, по электронной почте.

Для получения ссылки следует использовать пункт главного меню [Сервис – Получить ссылку](#) или нажать кнопку [Получить ссылку](#) области системных команд.

Рассмотрим пример получения ссылки на документ [Заказ 00000020](#):

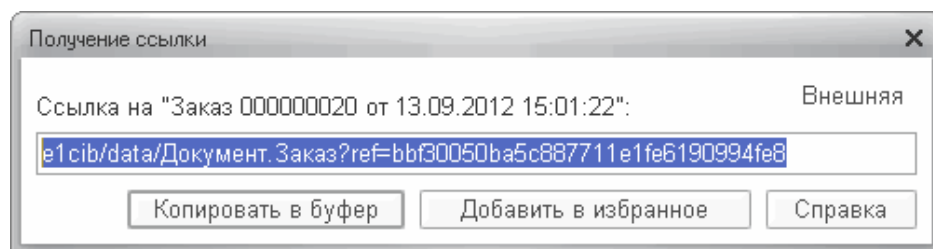


Рис. 162. Получение ссылки на документ «Заказ 0000020»

Чтобы получить ссылку для использования в адресной строке браузера (для веб-клиента), нужно нажать кнопку [Внешняя](#).

Для того чтобы добавить ссылку в буфер обмена, нажать кнопку [Копировать в буфер](#).

Чтобы добавить ссылку в список избранных, используется кнопка [Добавить в избранное](#).

Для перехода по ссылке следует использовать команду [Сервис – Перейти по ссылке](#), в открывшемся диалоге указать сохраненный ранее текст ссылки и нажать кнопку [Перейти](#) или клавишу [Enter](#). Для того чтобы отличать ссылки «1С:Предприятия» от других (например, [http](#) или [file](#)), используется идентификатор схемы [e1c](#). Если в диалог перехода по ссылке передается внешняя ссылка «1С:Предприятия», начало которой не совпадает с навигационной ссылкой информационной базы, то при переходе по ссылке выполняется попытка перейти по внутренней части, указанной после идентификатора [e1c](#). Если переход не удастся, то выполняется переход по переданной внешней ссылке.

При переходе по ссылке пробельные символы в начале и в конце строки не учитываются.

7.3. Избранное

Если панель избранного не используется, для вызова формы избранного нужно нажать кнопку [Избранное](#) панели инструментов.



Рис. 163. Кнопка «Избранное»

Панель избранного содержит ссылки на команды и данные, которые были добавлены в список избранных. Ссылки отображаются в порядке важности: важные элементы отображаются в начале списка.

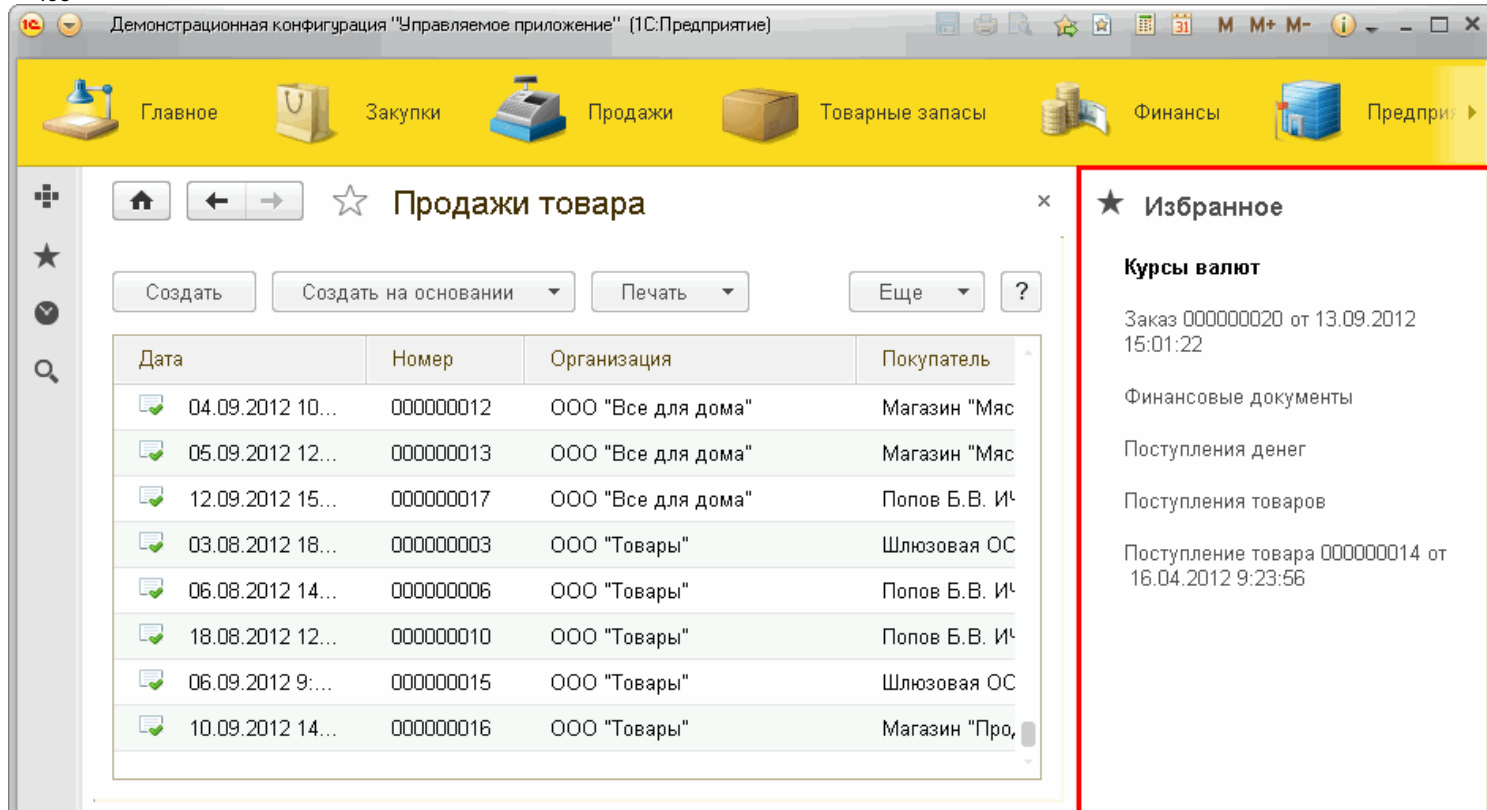


Рис. 164. Панель избранного

При выборе элемента откроется форма соответствующего объекта или выполнится команда. При нажатии на заголовок панели откроется форма избранного.

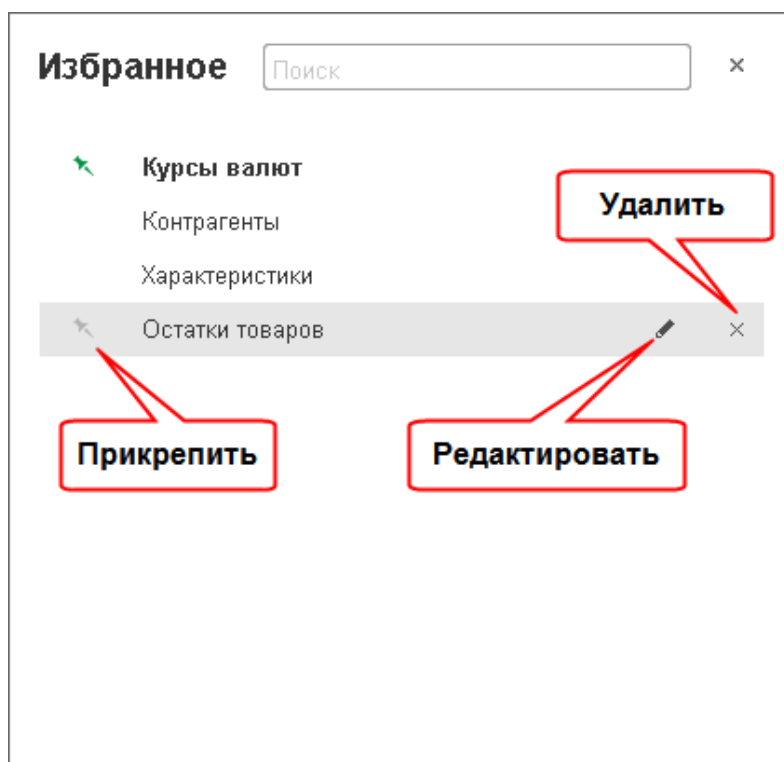


Рис. 165. Форма избранного

Переключатель слева от названия элемента используется для того, чтобы добавить (иконка активна) или удалить (иконка неактивна) элемент избранного из «прикрепленных».

Для переименования элемента в списке нужно нажать кнопку **Редактировать** (в виде карандаша) в строке элемента или клавишу **F2**.

Чтобы изменить порядок элементов, следует перетащить выбранный элемент избранного выше или ниже с помощью мыши.

Для удаления элемента из списка нажать кнопку [Удалить](#).

Все изменения в списке сохраняются при закрытии формы избранного.

Поиск в Избранном. Для поиска по списку избранного можно использовать поле поиска, или нажать **Ctrl + F**, или начать ввод названия искомого объекта (при этом программа установит фокус в поле поиска). При этом в списке избранного отображаются только те элементы, в представлении которых содержится введенная строка, найденные подстроки подсвечиваются. Для завершения поиска и возврата к полному списку избранного следует очистить строку поиска.

7.4. История

«1С:Предприятие» сохраняет историю работы пользователя, которую можно использовать для быстрого доступа к недавно созданным или отредактированным объектам информационной базы (документам, элементам списков и др.).

При интерактивном создании или изменении объектов информация об этом отображается в виде оповещения и попадает в историю (подробнее о механизме оповещений см. в разделе [«Оповещения»](#) данной главы).

История работы пользователя хранится в информационной базе. При этом в истории хранится одна запись на один объект информационной базы за каждый день (запись о последующем изменении замещает запись о предыдущем изменении данного объекта).

Панель истории содержит гиперссылки на формы, упорядоченные по времени открытия (кроме начальной страницы). При выборе элемента списка открывается форма объекта.

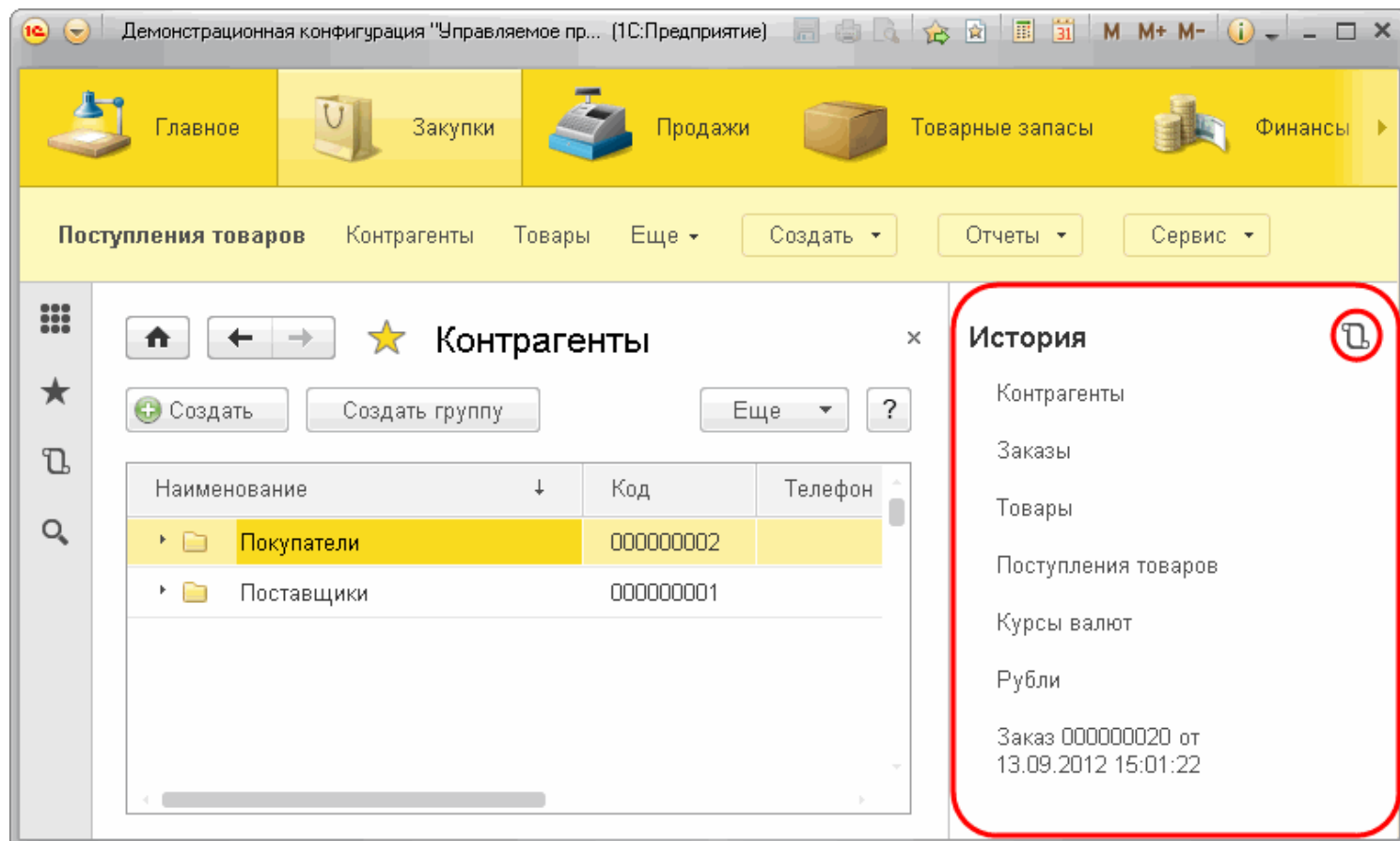


Рис. 166. Панель истории

При нажатии на заголовок панели или на иконку в заголовке откроется форма истории.

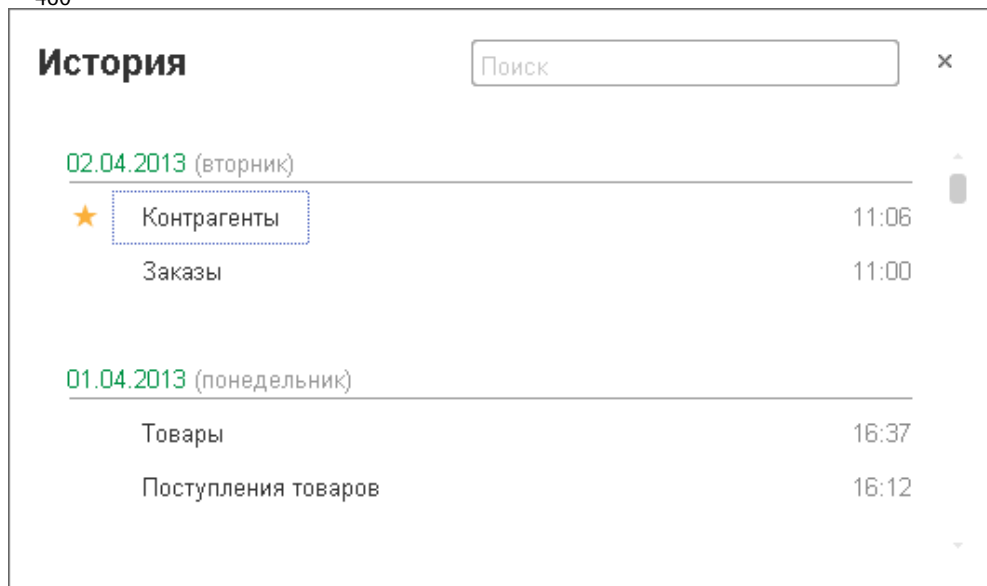


Рис. 167. Форма истории

Все элементы сгруппированы по датам. Если одна и та же форма была открыта в разные дни, то ссылка на нее будет присутствовать в каждом из этих дней. В пределах одного дня ссылки не повторяются, а форма отображается с последним временем открытия.

Если элемент истории содержится в списке избранного, рядом с ним отображается звездочка. Переключение звездочки в строке элемента добавляет/удаляет элемент из списка избранного.

Поиск в истории. Для поиска по истории можно использовать поле поиска, или нажать **Ctrl + F**, или начать ввод текста. Поиск по истории начинается при вводе в строке поиска. При этом в списке истории отображаются только те элементы, в представлении которых содержится введенная строка, найденные подстроки подсвечиваются. Во время поиска элементы в списке по дням не группируются, а рядом с элементами отображается время и дата открытия. Для завершения поиска и возврата к полному списку истории следует очистить строку поиска.

7.5. Все функции

Система предоставляет возможность открывать все доступные формы. Для этого используется список, отображающийся при вызове команды главного меню **Все функции**. Команда доступна, если установлена соответствующая настройка в параметрах системы и у пользователя есть право вызова данного режима (подробнее см. раздел [«Системные параметры»](#) [здесь](#)).

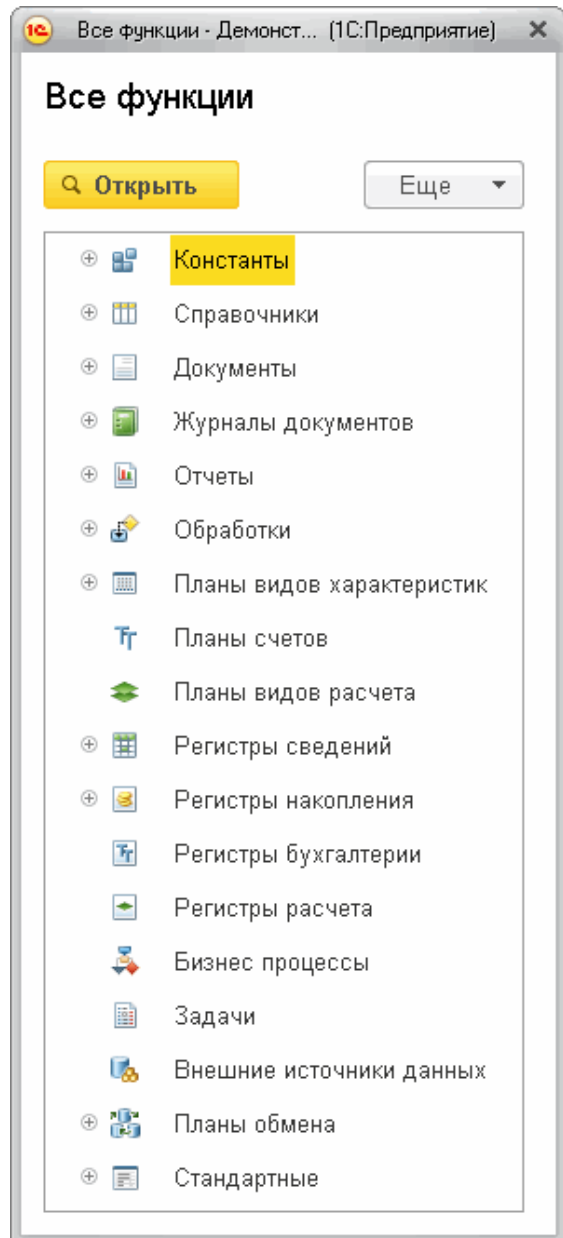


Рис. 168. Список всех функций системы

В список не включены объекты, на которые у пользователя нет права на просмотр.

В группе **Стандартные** отображаются стандартные функции (список активных пользователей, журнал регистрации и т. д.). Подробнее о стандартных функциях см. книгу «1С:Предприятие 8.3. Руководство администратора».

Чтобы перейти к объекту, нужно дважды щелкнуть на объекте в дереве объектов или выбрать объект в дереве и нажать кнопку **Открыть**.

Для быстрого перехода к нужному объекту в списке можно воспользоваться функцией поиска. При вводе текста с клавиатуры поиск будет выполняться по началам названий объектов. Также можно использовать сочетание клавиш **Ctrl + F** для поиска в любой части названия функции. При этом будет использован стандартный диалог поиска в списке.

7.6. Оповещения

Механизм оповещений предназначен для информирования пользователя о том, что система выполнила то или иное действие. Оповещения могут создаваться системой или разработчиком прикладного решения.

Оповещение отображается в окне, которое по умолчанию привязано к системному лотку панели задач операционной системы. Если с оповещением связана какая-либо ссылка, то пояснение будет одновременно являться гиперссылкой. При ее нажатии откроется объект, на который указывает ссылка. Пояснение также будет представлено гиперссылкой, если оповещение сформировано системой автоматически (при интерактивной записи/изменении объекта).

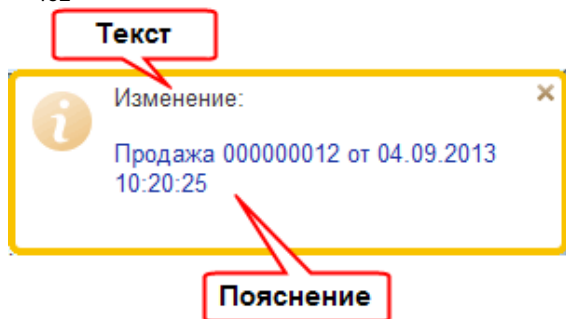


Рис. 169. Оповещение об изменении документа «Поступление денег»

- **Текст** – причина оповещения. Если было изменено несколько объектов, то в скобках будет указано количество измененных объектов;
- **Пояснение** – уточняющая информация.

Окно оповещения исчезает с экрана по истечении некоторого времени. Однако если навести на него курсор мыши, то окно будет отображаться до тех пор, пока его не закрыть или не отвести курсор мыши.

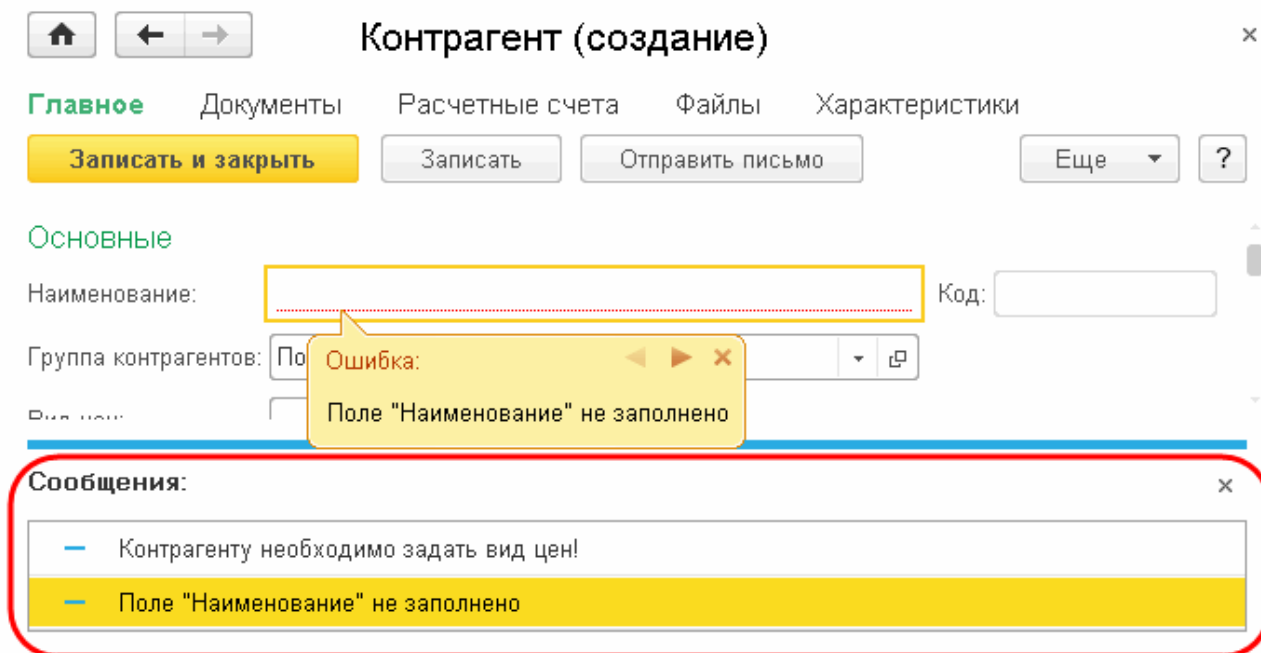
Если в процессе работы системой сгенерировано несколько разных оповещений, то в тексте оповещения в скобках будет указано количество измененных объектов, а в качестве пояснения будет использовано название списка.

При отображении в информационной панели список оповещений выравнивается по правому краю панели. Наиболее позднее оповещение будет отображаться у правого края панели. Если оповещений больше пяти, то отображаются только пять последних оповещений.

ПРИМЕЧАНИЕ. Список оповещений очищается при закрытии основного окна приложения.

7.6.1. Сообщения

При возникновении ситуаций, на которые следует обратить внимание, в нижней части окна отображается панель сообщений.



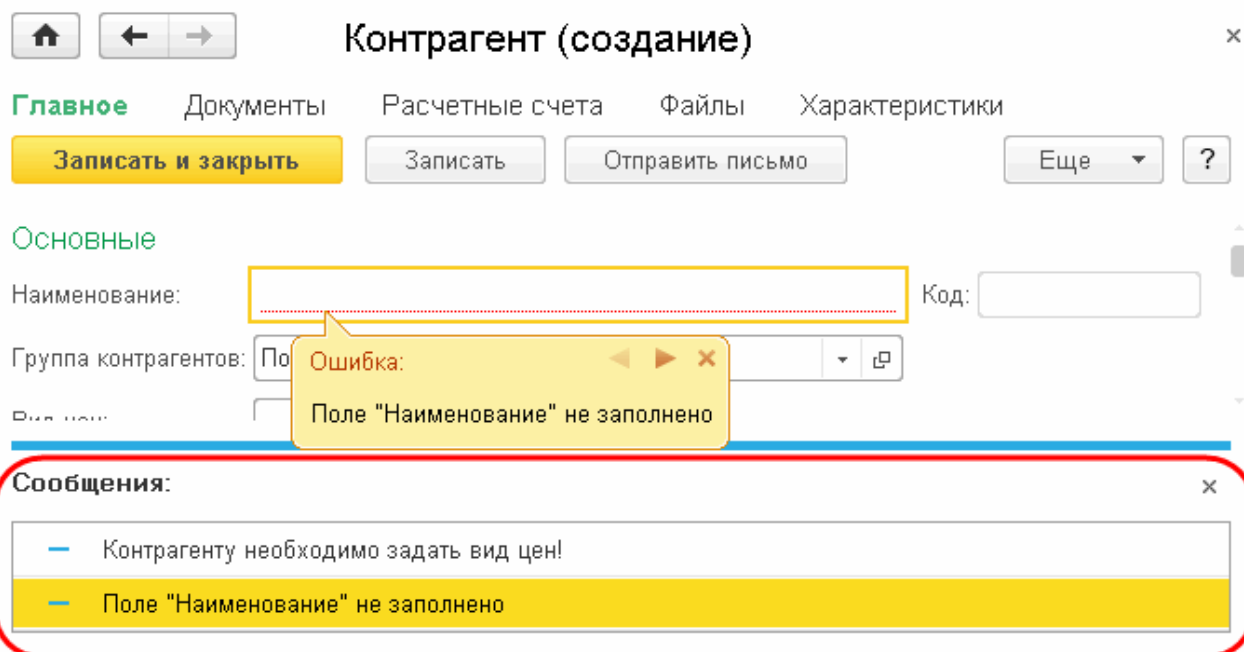


Рис. 170. Панель сообщений

Работа с сообщениями аналогична работе с сообщениями в стандартном интерфейсе. Одновременно могут отображаться до пяти сообщений. Чтобы увидеть не поместившиеся сообщения, следует прокрутить список с помощью клавиш [Стрелка вверх/Стрелка вниз](#) или с помощью мыши.

7.7. Состояние длительного процесса

Во время работы с программой могут возникать ситуации, когда системе требуется время для выполнения некоторого процесса. Например, при проведении большого количества документов, выполнении тяжелых расчетов. При этом на экране отображается сообщение, содержащее состояние длительного процесса, см. [рис. 171](#).

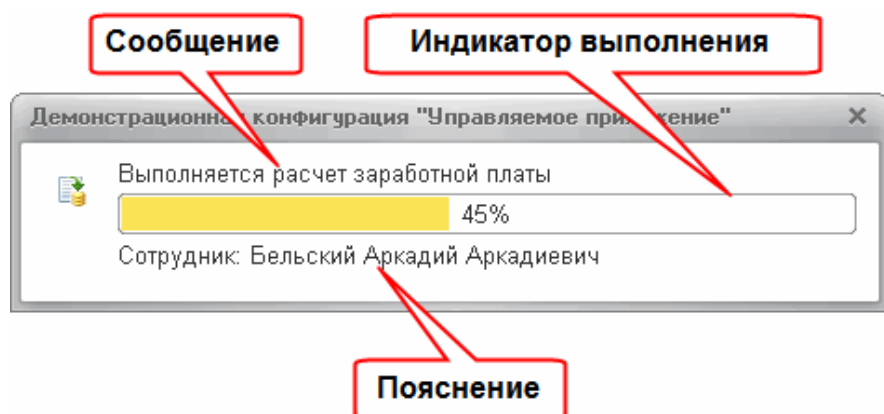


Рис. 171. Состояние выполнения процесса

Сообщение автоматически «гаснет» по истечении некоторого времени. При наведении курсора на сообщение оно снова отображается. В зависимости от настроек программы выполнение процесса может быть прервано по нажатию клавиш [Ctrl + Break](#).

7.8. Калькулятор

Для выполнения несложных текущих расчетов система «1С:Предприятие» имеет встроенный калькулятор. Его можно вызвать в любом режиме программы (кроме случаев, когда программа ждет ответа на запрос), выбрав пункт главного меню [Сервис – Калькулятор](#).

Калькулятор можно перетащить мышью в любое место экрана, а также изменить размер окна. При этом все элементы управления, расположенные в форме калькулятора, изменяют свой размер пропорционально изменению размеров окна.

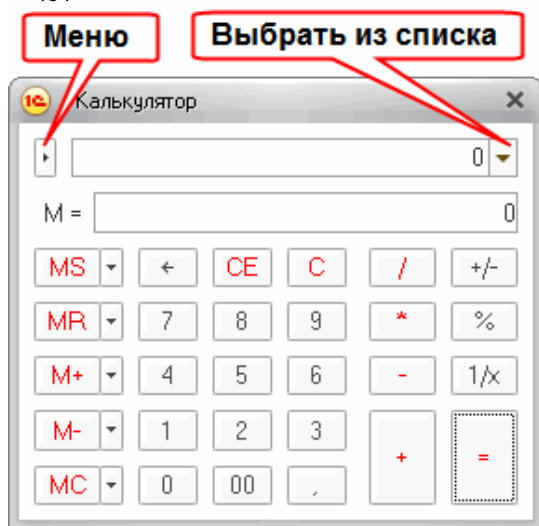


Рис. 172. Калькулятор

Калькулятор позволяет выполнять все арифметические действия, вычислять проценты и обратные значения, запоминать числа и выполнять сложение и вычитание с запомненными значениями. При вводе числа калькулятор позволяет редактировать введенное значение, удаляя неверно введенные разряды.

Для работы с калькулятором можно использовать как мышь, так и клавиатуру. По выполняемым функциям цифровой калькулятор напоминает «настоящие» калькуляторы.

Калькулятор запоминает последние 15 вычислений. Для просмотра результата любого вычисления нужно нажать кнопку **Выбрать из списка**, расположенную справа от поля ввода, и выбрать нужное вычисление.

Калькулятор имеет 10 регистров памяти для запоминания чисел и оперирования ими при вычислениях. С помощью кнопок **MR**, **M+**, **M-** и **MC** оперируют с первым регистром.

Для запоминания результата вычисления в первом регистре нажать кнопку **MS**.

В поле **M =** отображается значение буфера обмена, если в нем содержится число.

Нажатием кнопки **MR** запомненное значение читается из первого регистра в поле ввода. По кнопке **M+** запомненное в первом регистре значение прибавляется к результату вычислений, по кнопке **M-** запомненное в первом регистре значение вычитается из результата вычислений. Кнопка **MC** сбрасывает содержимое первого регистра.

Для запоминания числа в другие регистры число должно быть набрано в поле ввода (нужно ввести его или произвести расчеты). Затем нажать кнопку **Выбрать**, расположенную рядом с кнопкой **MS**.

На экран выводится список регистров.

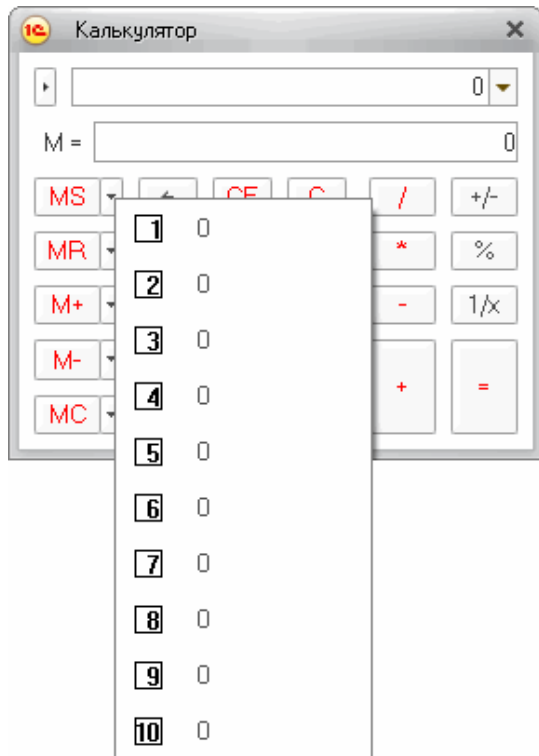


Рис. 173. Список доступных регистров

Для записи числа в регистр следует нажать кнопку соответствующего регистра.

Для извлечения числа из регистра нужно нажать кнопку, расположенную рядом с кнопками **MR**, **M+** или **M-**, в зависимости от вида действия, и выбрать регистр. Аналогично выполняется процедура для очистки регистра (кнопка **MC**).

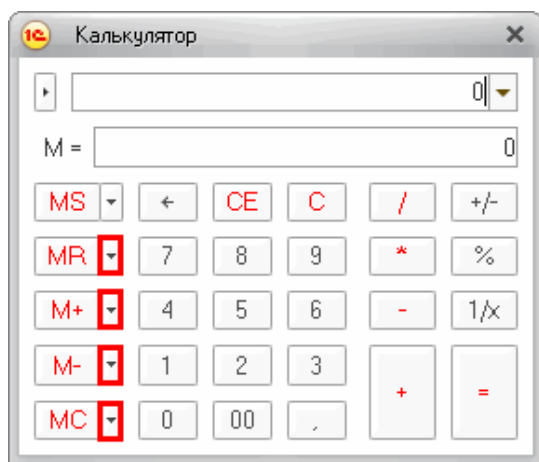


Рис. 174. Кнопки для работы с регистрами

Калькулятор может работать в различных режимах. Выбор режима производится нажатием кнопки **Меню** калькулятора (см. [рис. 172](#)) и выбором в контекстном меню нужного пункта. Пункты меню **Формульный калькулятор** и **Микрокалькулятор** позволяют включить или выключить соответствующий режим работы калькулятора.

Пункт контекстного меню **Настройка** предназначен для установки точности калькулятора. На экран выводится диалог настройки, в котором указывается число знаков после запятой для представления результата.

В режиме **Микрокалькулятор** все действия выполняются с клавиатуры. Калькулятор имеет упрощенную форму.



Рис. 175. Калькулятор в режиме «Микрокалькулятор»

Для ввода чисел используются цифровые кнопки клавиатуры. Для очистки поля ввода используется клавиша **C**.

Формульный калькулятор. В этом режиме все действия записываются в строку калькулятора последовательно. Для организации определенной последовательности действий и ввода сложных выражений допускается использование скобок.

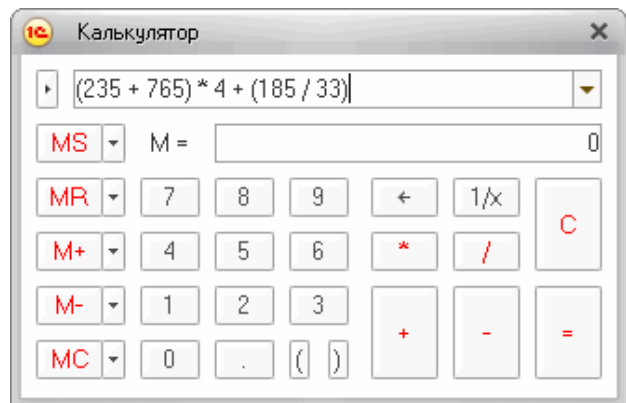


Рис. 176. Формульный калькулятор

Калькулятор вызывается для выбора значения реквизита типа **Число** в поле ввода нажатием кнопки выбора или по клавише **F4**. Калькулятор позволяет вводить большее число знаков после десятичной точки, чем определено для реквизита. При нажатии кнопки **OK** производится округление по установленным правилам. Для настройки точности калькулятора нужно нажать кнопку **Меню** и выбрать **Настройка**. На экран выводится диалог настройки, в котором указывается число знаков после запятой для представления результата.

7.8.1. Работа с буфером обмена

При работе с буфером обмена система «1С:Предприятие» предоставляет расширенный сервис работы с числами. Кроме запоминания числа в буфер число можно добавить к значению в буфере или вычесть из значения в буфере.

Для этого обычно используются кнопки стандартной командной панели **M**, **M+** и **M-**.

При нажатии кнопки **M** (можно использовать сочетание клавиш **Shift + Num***) выделенное значение помещается в буфер обмена.

Кнопка **M+** (сочетание клавиш **Shift + Num+**) позволяет добавить текущее выделенное значение к значению в буфере обмена.

Кнопка **M-** (сочетание клавиш **Shift + Num-**) позволяет вычесть текущее значение из буфера обмена.

Данные действия доступны во всех формах для числовых значений и при работе с таблицами. Кроме того, при работе с табличным документом существует возможность выполнять одновременные действия с несколькими ячейками: например, можно выделить несколько ячеек и, используя указанные действия, поместить или добавить в буфер обмена их сумму.

Полученные результаты в буфере обмена можно использовать в любых формах, например, вставляя значения с помощью клавиш **Shift + Insert** или соответствующих пунктов меню.

Числовое выражение текущего значения буфера обмена можно просмотреть в режиме формульного калькулятора в поле **M=**.

Если требуется вставить число из среды «1С:Предприятия» в калькулятор **Windows** или в ячейку **Microsoft Office Excel**, то для запоминания нужно использовать команды **M**, **M+**, **M-**, а не команды запоминания в буфер обмена.

Следует иметь в виду, что для формульного калькулятора и табло число сохраняется в соответствии с региональными установками информационной базы, но без разделителей групп и замены разделителя целой и дробной части на точку.

Для всех остальных приемников буфера обмена число сохраняется с региональными установками операционной системы и также без разделителей групп.

7.9. Календарь

Система «1С:Предприятие» содержит встроенный календарь. Его можно вызвать в любом режиме программы, выбрав пункт главного меню [Сервис – Календарь](#). Календарь также используется для ввода дат.

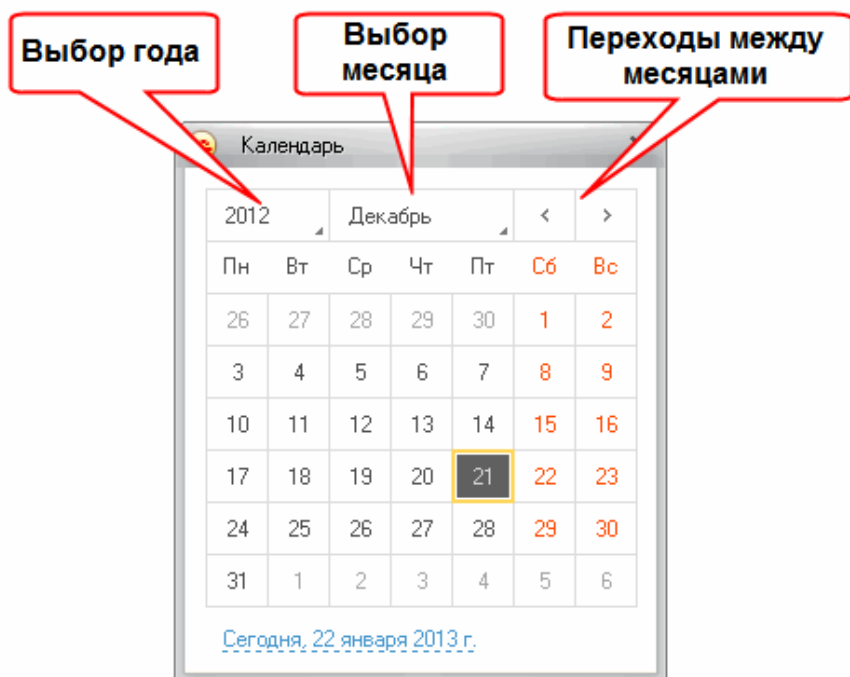


Рис. 177. Окно календаря

В окне выводится календарь на текущий месяц. Его наименование и год выводятся в верхней строке. Ниже расположена строка дней недели. Зеленой рамкой выделяется текущая дата.

Для смены года используется меню выбора года, вызываемое при нажатии на текущий номер года.

СОВЕТ. При выборе года между диапазонами лет можно перемещаться как с помощью кнопок меню, так и с помощью клавиш [PgUp](#) и [PgDown](#).

Для быстрого выбора текущей даты нажмите ссылку в нижней части окна календаря [Сегодня, <Дата>](#).

СОВЕТ. Для быстрого выбора даты начала и конца месяца можно использовать клавиши [Home](#) и [End](#).

Для перехода между месяцами используются кнопки, расположенные справа от заголовка. Для перехода к конкретному месяцу используется меню выбора месяца, вызываемое при нажатии левой кнопкой мыши на название месяца.

Для смены месяца следует выбрать нужную строку списка. Также для перехода между месяцами можно использовать клавиши [PgUp](#) и [PgDown](#).



Рис. 178. Календарь. Список месяцев

Календарь можно перетащить мышью в любое место экрана, а также изменить размер окна календаря.

7.10. Сравнение файлов

Режим сравнения файлов дает возможность пользователю сравнить два любых файла. Для сравнения файлов необходимо выбрать пункт главного меню [Файл – Сравнить файлы](#). На экран будет выдан диалог для выбора

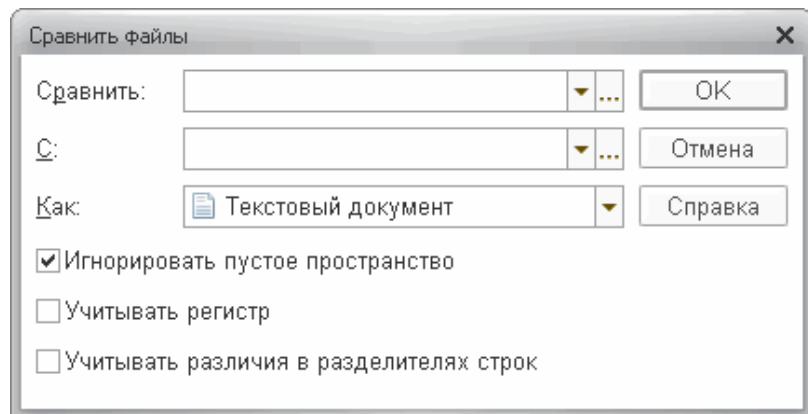


Рис. 179. Диалог выбора файлов для сравнения

В полях **Сравнить** и **С** необходимо указать имена сравниваемых файлов. Их можно ввести вручную, выбрать из выпадающего списка или нажатием кнопки **Выбрать** вызвать стандартный диалог открытия файла и выбрать файл в этом диалоге. В поле **Тип файлов** диалога можно указать любой тип исходного файла.

В поле **Как** указывается вид сравнения файлов. Текстовый и табличный документы сравниваются. Допускается двоичное сравнение. В этом случае результатом сравнения будет сообщение о совпадении или различии файлов.

Если в полях **Сравнить** и **С** указаны файлы различных типов, то выбранные файлы при сравнении будут приводиться к виду, указанному в поле **Как**. Затем выполнится сравнение, и на экран будет выведена информация о его результатах.

Если установлен флажок **Учитывать различия в разделителях строк**, то при сравнении будут учитываться символы, используемые в качестве разделителей строк.

Для начала сравнения нужно нажать кнопку **ОК**.

Если выбран вид сравнения **Текстовый документ**, то выбранные файлы будут интерпретироваться при сравнении как текстовые, несмотря на то, что в полях **Сравнить** и **С** могут быть выбраны нетекстовые (по расширению) файлы. Если указанные файлы или один из них не могут быть интерпретированы как текстовые, то производится двоичное сравнение.

7.10.1. Сравнение текстовых файлов

Для текстовых файлов установка флажка **Игнорировать пустое пространство** означает, что пробелы сравниваться не будут.

Флажок **Различать регистр** позволяет при сравнении различать прописные и строчные буквы. Если флажок установлен, одна и та же прописная и строчная буквы будут считаться разными символами.

Флажок **Учитывать разделители строк** означает, что при сравнении файлов будут учитываться разделители строк.

Перед сравнением программа последовательно выполняет попытку чтения файлов в кодировках UTF-8, UTF-16 и ANSI. Если файлы прочитаны и если при чтении в ANSI файлы не содержат 0, то программа сравнивает их как текстовые документы. Если попытки чтения были неудачными (или для ANSI содержится 0), то программа сравнивает их как двоичные данные.

Для текстовых файлов на экран будет выдано окно просмотра различий. В двух колонках окна просмотра различий будут помещены сравниваемые тексты. В текстах разными цветами выделены удаленные, измененные и добавленные (новые) строки. В окне имеется панель инструментов со стандартными для текстового редактора «1С:Предприятия» кнопками поиска и закладок, а также двумя кнопками, специфическими для данного окна.

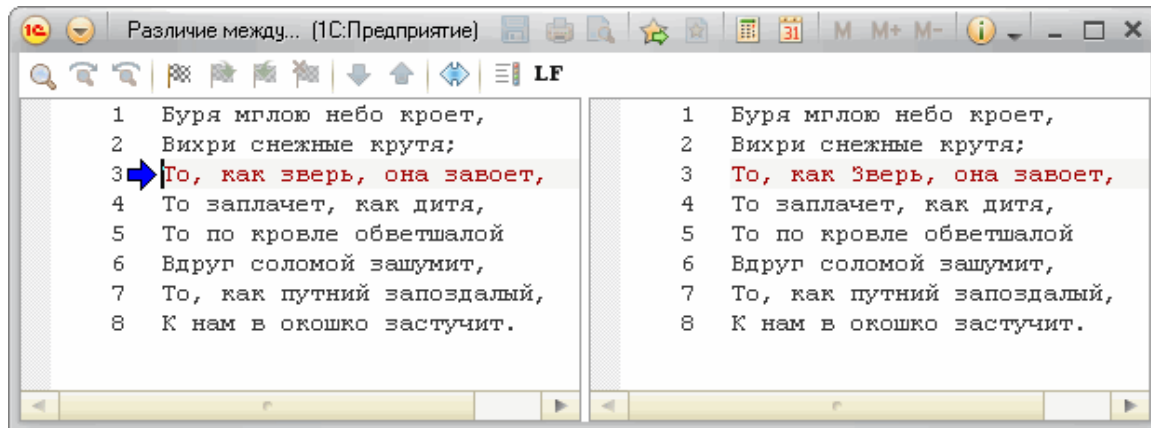


Рис. 180. Диалог сравнения текстовых файлов

Это большие стрелки, направленные одна вверх, другая вниз. Они служат для быстрого перехода на начало ближайшего предыдущего или последующего измененного текста.

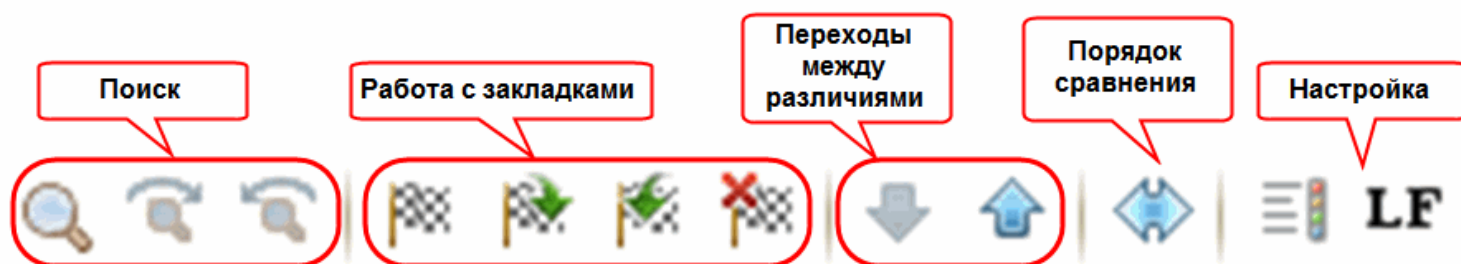


Рис. 181. Панель инструментов для сравнения файлов

При нажатии кнопки **LF** в текстах отображаются разделители строк в виде букв **LF** и **CR**.

Различия в сравниваемых текстах показываются с помощью цвета текста. Цвет текста можно настроить (по кнопке **Настройка** запускается окно настройки цветов). По умолчанию удаленный текст выделяется синим цветом, добавленный – зеленым, а измененный – красным.

В левой части окна показывается текст файла, указанного в поле **Сравнить**, в правой – указанного в поле **С**. Нажатием кнопки **Порядок сравнения** можно поменять местами сравниваемые файлы. При этом меняется и цвет пометки. Если первоначально в левой части был показан удаленный текст, то при смене порядка сравнения этот текст в правой части будет показан как добавленный.

7.10.2. Сравнение табличных документов

Если сравниваются табличные документы, то на экран будет выдано окно просмотра различий.

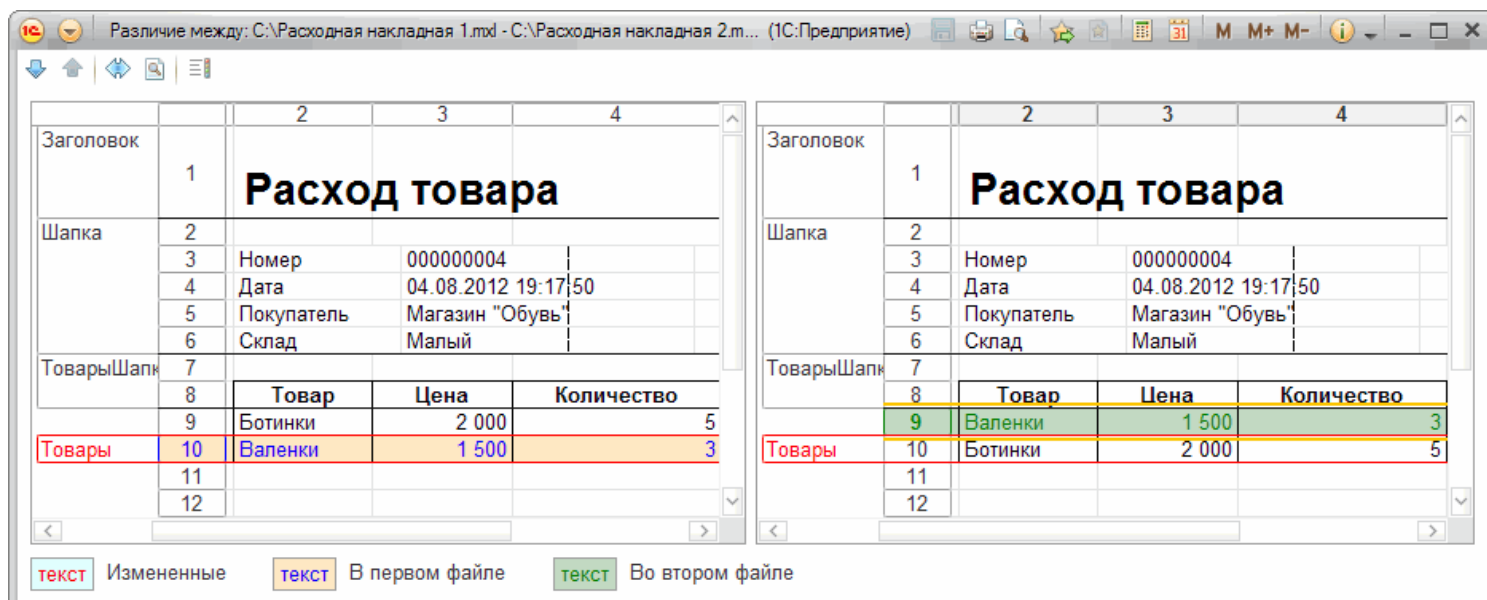


Рис. 182. Окно сравнения табличных документов

Окно состоит из двух частей (левой и правой). В каждую часть окна просмотра различий помещается исходный документ. С помощью кнопок командной панели производится управление просмотром и настройка цветов.

Кнопками [Следующее различие](#) и [Предыдущее различие](#) осуществляется переход к следующему или предыдущему различию.

Различия показываются с помощью цвета фона и текста. Обозначения цветов перечислены в легенде.

С помощью кнопки [Настройка параметров просмотра сравнения](#) открывается диалог настройки параметров показа различий.

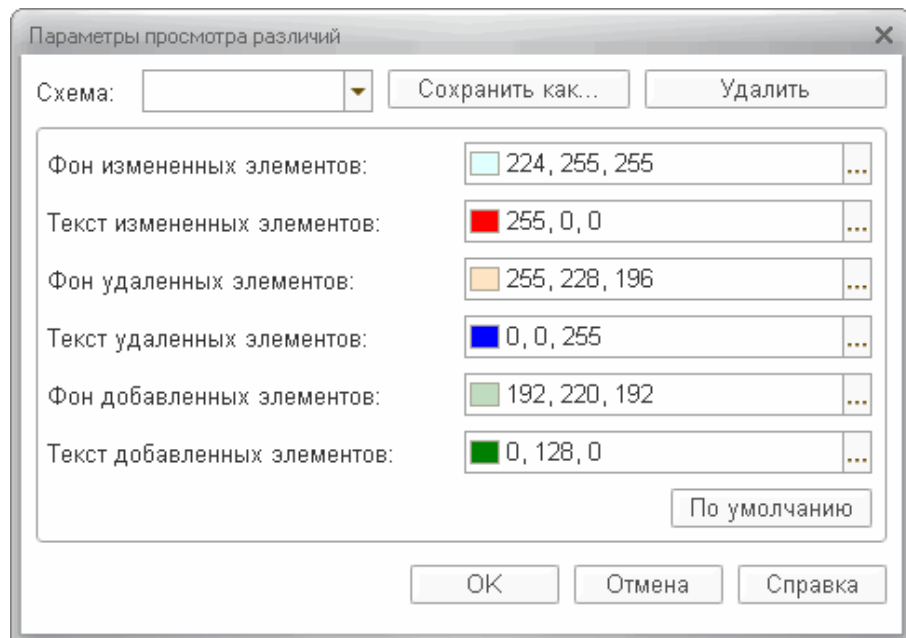


Рис. 183. Настройка параметров показа различий

Значения настроек можно сохранять как схемы. Для этого нужно нажать кнопку [Сохранить как](#), в открывшемся диалоге ввести наименование настройки и нажать кнопку [ОК](#).

Если есть несколько схем настроек, то выбор нужной осуществляется в поле [Схема диалога настройки](#).

Для восстановления стандартных настроек цветов нужно нажать кнопку [По умолчанию](#).

Различия в ячейках, строках, свойствах строк и колонок показываются цветом фона и цветом текста.

Различия в объединениях ячеек показываются обводящей рамкой цвета текста.

Различия в группировках и именованных областях обозначаются цветом текста.

Различия в рисунках показываются обводящей рамкой цвета текста, кроме прямоугольника, линии и овала, которые рисуются цветом текста.

В окне разными цветами выделены измененные, удаленные и добавленные (присутствуют в первом или втором файлах) строки.

Для выделенных ячеек можно получить подробный отчет о невидимых различиях (текст, шрифт, формат, объединение, имена и т. д.). Для этого нужно нажать кнопку [Подробнее...](#)

7.11. Сообщения об ошибках

В случае возникновения ошибки при выполнении программы на экран выводится предупреждение (может содержать разное число кнопок, назначение кнопок также может быть разным), например, следующего вида:

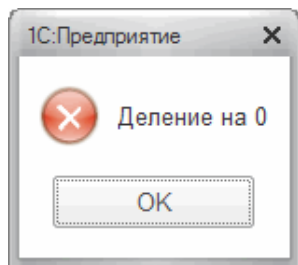


Рис. 184. Пример ошибки

Чтобы получить более подробную информацию об ошибке в окне "О программе" следует нажать ссылку [Информация для технической поддержки](#). Подробнее см. [здесь](#).

При аварийном завершении программы «1С:Предприятие» система отображает диалог с информацией о процессе записи дампа, который автоматически закрывается после завершения записи.

Глава 8. Настройка программы

Система «1С:Предприятие» позволяет настраивать интерфейс приложения, как глобальный, так и каждую форму в отдельности (если эта возможность есть в конфигурации), задавать системные параметры, управлять отображением текстовых и табличных документов и расположением окон.

8.1. Интерфейс

При работе с «1С:Предприятием» существует возможность настраивать интерфейс приложения, в частности:

- начальную страницу;
- панель разделов;
- панели навигации (как основного, так и вспомогательного окна);
- панель действий;
- командную панель вспомогательного окна.

Пользователь может выбирать команды из доступных и управлять их расположением.

Диалоги настройки вызываются из главного меню "1С:Предприятия" (Пункт главного меню [Вид](#)).

Также окно настройки для каждой панели пользователь может вызвать из ее контекстного меню любой из панелей.

8.1.1. Начальная страница

Система позволяет устанавливать состав форм, которые отображаются на начальной странице. При этом можно управлять только теми формами, которые расположены на начальной странице.

Для перехода к диалогу настройки начальной страницы в главном меню нужно выбрать команду [Вид – Настройка начальной страницы](#).

Настройка производится в диалоге, состоящем из трех списков: [Доступные формы](#) и группы колонок [Левая колонка](#), [Правая колонка](#). В списке [Доступные формы](#) отображаются формы, не используемые ни в левой, ни в правой колонке.

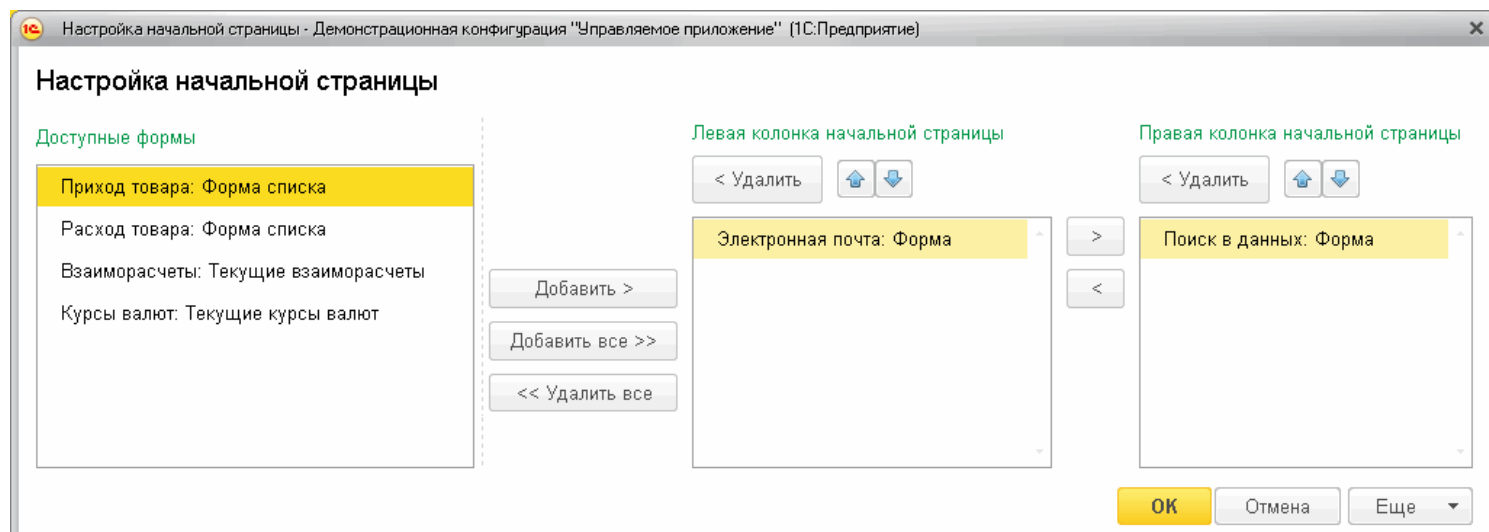


Рис. 185. Диалог настройки начальной страницы

Формы можно перетаскивать из одного списка в другой с помощью кнопок [Добавить >](#), [Добавить все >>](#), перемещать между колонками с помощью кнопок [>](#) и [<](#). Чтобы удалить форму из колонки (поместить ее обратно в список доступных форм), нужно выбрать ее в списке и нажать кнопку [Удалить <](#) или дважды щелкнуть по названию формы в списке. Чтобы удалить все формы с начальной страницы, нажать кнопку [<< Удалить все](#). Также можно изменять порядок форм в колонках с помощью кнопок [Вверх – Вниз](#). Формы выводятся на начальной странице в том же порядке, который установлен в настройках.

В процессе настройки можно отменить изменения, сделанные как в данном сеансе работы, так и ранее, и вернуться к настройкам, заданным в конфигурации. Для этого в меню **Еще** имеется команда **Установить стандартные настройки**. Эта команда не вызывает немедленного изменения командного интерфейса. После ее выполнения настройку можно продолжить. Применение настроек происходит по нажатию кнопки **OK**.

8.1.2. Настройка отображения панелей

Чтобы настроить или спрятать панели, нужно использовать соответствующие команды главного меню **Вид**.

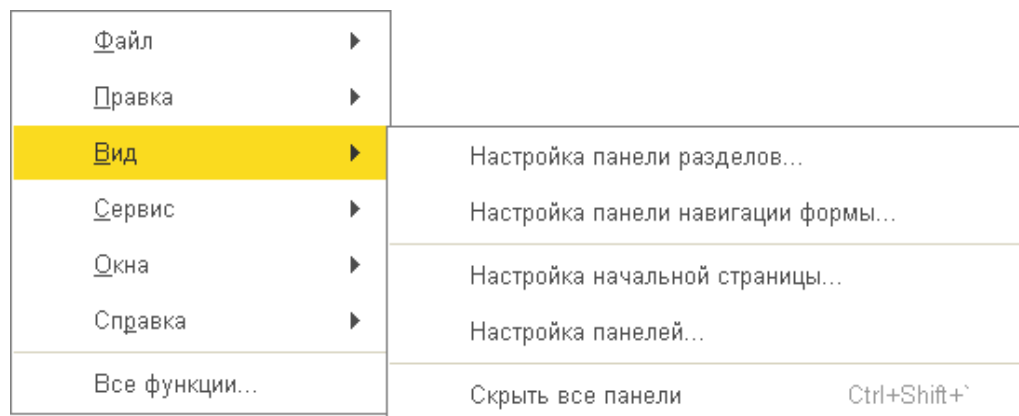


Рис. 186. Меню «Вид»

Данное подменю позволяет управлять видимостью панелей и содержимым панели разделов, панелей навигации формы и начальной страницы. Это дает возможность увеличить место под формы и уменьшить вероятность появления полос прокрутки. Чтобы спрятать все панели, используется команда главного меню **Вид – Скрыть все панели** или сочетание клавиш **Ctrl + Shift + `**.

8.1.3. Панель разделов

При настройке панели разделов пользователь устанавливает список команд перехода к разделам.

Диалог настройки панели разделов открывается командой главного меню **Вид – Настройка панели разделов...**

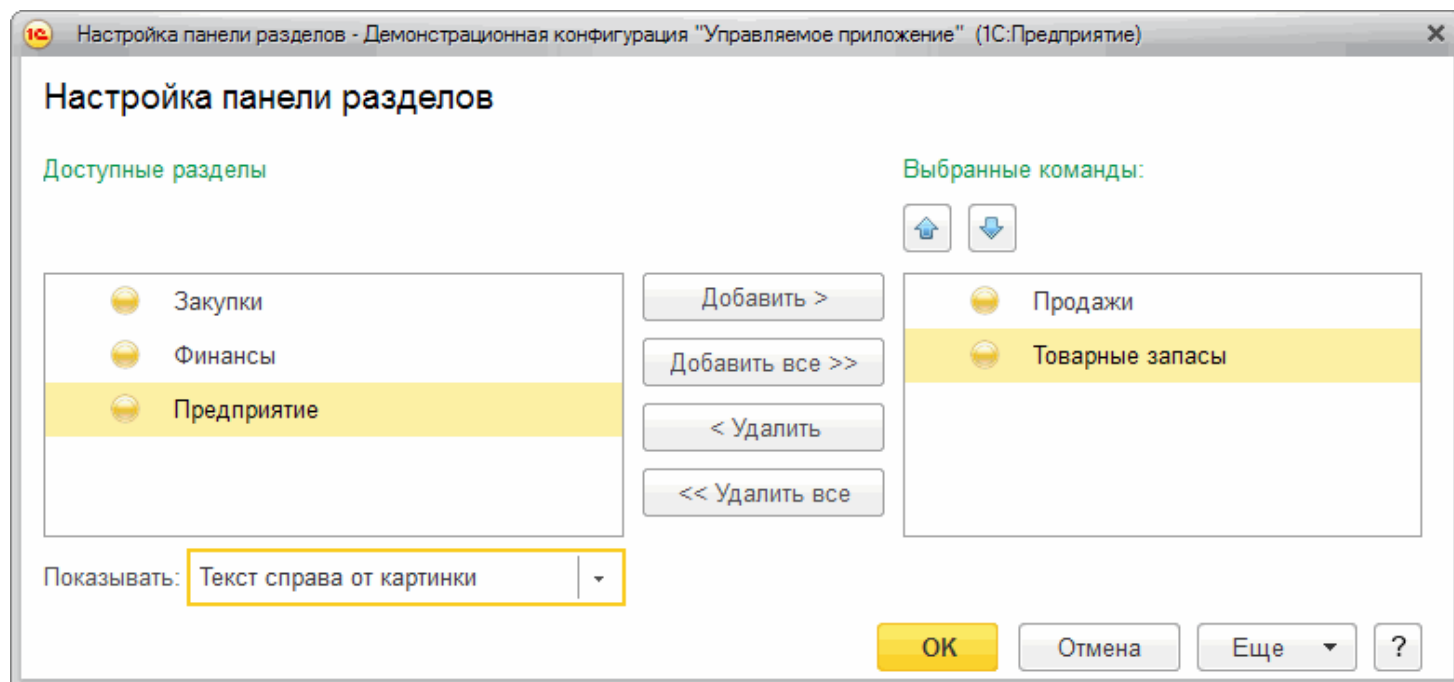


Рис. 187. Форма настройки панели разделов

Чтобы добавить раздел на панель, нужно выбрать его в списке доступных разделов и нажать кнопку **Добавить >**. В диалоге настройки поддерживается перемещение элементов списка с помощью мыши и выбор нескольких строк.

Чтобы добавить на панель все доступные разделы, нужно нажать кнопку **Добавить все >>**. Также можно перемещать разделы между списками двойным нажатием левой кнопки мыши по разделу в списке.

Чтобы удалить команды из панели, следует использовать кнопки [Удалить](#) и [Удалить все](#).

Последовательность разделов на панели регулируется с помощью кнопок командной панели списка выбранных (см. [рис. 187](#)).

Для любого раздела можно настроить представление. Для этого нужно выбрать пункт и установить требуемый способ отображения в списке [Показывать](#):

- [Картинка](#) – каждый раздел отображается только картинкой;
- [Текст](#) – каждый раздел отображается только названием;
- [Текст справа от картинки](#) – раздел отображается названием и картинкой (текст справа);
- [Текст под картинкой](#) – раздел отображается названием и картинкой (текст слева).

Чтобы применить выбранные настройки и закрыть форму, нужно нажать кнопку [OK](#). При этом происходит сохранение настроек в информационной базе.

В процессе настройки можно отменить изменения, сделанные как в данном сеансе работы, так и ранее, и вернуться к настройкам, заданным в конфигурации. Для этого нужно выбрать пункт [Еще – Установить стандартные настройки](#). Исполнение этой команды не вызывает немедленного изменения командного интерфейса. После ее выполнения можно продолжить настройку.

8.1.4. Панель навигации

Настройка расположения команд на панели навигации производится в диалоге [Настройка панели навигации](#).

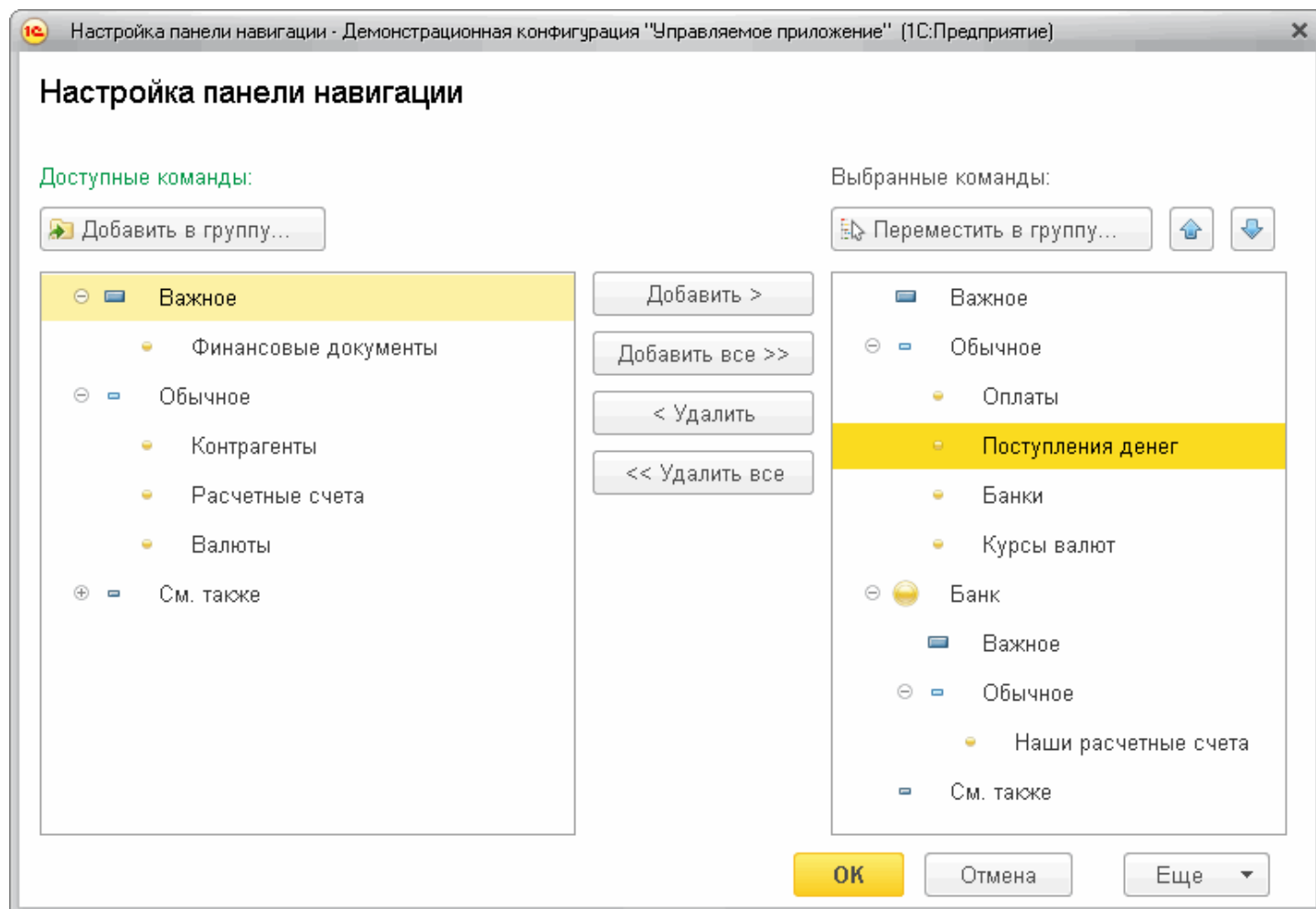


Рис. 188. Настройка панели навигации

- команды перехода к формам списков, например, [Банки](#), [Контрагенты](#);
- подчиненные разделы (например, на [рис. 188](#) текущему разделу подчинен раздел [Банк](#), для которого определена команда [Наши расчетные счета](#)).

Все команды распределены по группам:

- [Важное](#),
- [Обычное](#),
- [См. также](#).

Чтобы добавить команду на панель, нужно выбрать ее в списке доступных команд и нажать кнопку [Добавить >](#).

Чтобы добавить на панель все доступные команды, нажать кнопку [Добавить все >>](#). Также можно перемещать команды между списками двойным нажатием левой кнопки мыши по разделу в списке.

Для удаления команд из панели используются кнопки [Удалить](#) и [Удалить все](#).

Последовательность команд регулируется с помощью стрелок на командной панели списка выбранных команд (см. [рис. 188](#)).

Чтобы добавить команду из списка доступных в группу выбранных команд, нужно выбрать ее и нажать кнопку [Добавить в группу](#).

Для перемещения команды в необходимую группу в списке выбранных команд нужно выбрать команду и нажать кнопку [Переместить в группу](#) или перетащить команду в требуемую группу с помощью мыши. Если при этом выбрана группа или подраздел, то операция выполняется для всех команд в группе/подразделе.

Диалоги настройки поддерживают перетаскивание команд между списками и множественный выбор.

ПРИМЕЧАНИЕ. Если некоторая команда относится к подразделу, то она может быть перемещена в другую группу только внутри подраздела, к которому она относится. Например, на [рис. 188](#) команда [Наши расчетные счета](#) может быть перемещена только в группу [Важное](#) подраздела [Банк](#).

Исключение составляет группа [См. также](#), которая может содержать любые команды.

Чтобы применить выбранные настройки и закрыть форму, нужно нажать кнопку [ОК](#).

Заметим, что порядок может быть настроен только для команд.

8.1.5. Панель действий

Настройка расположения команд на панели действий производится в диалоге [Настройка панели действий](#).

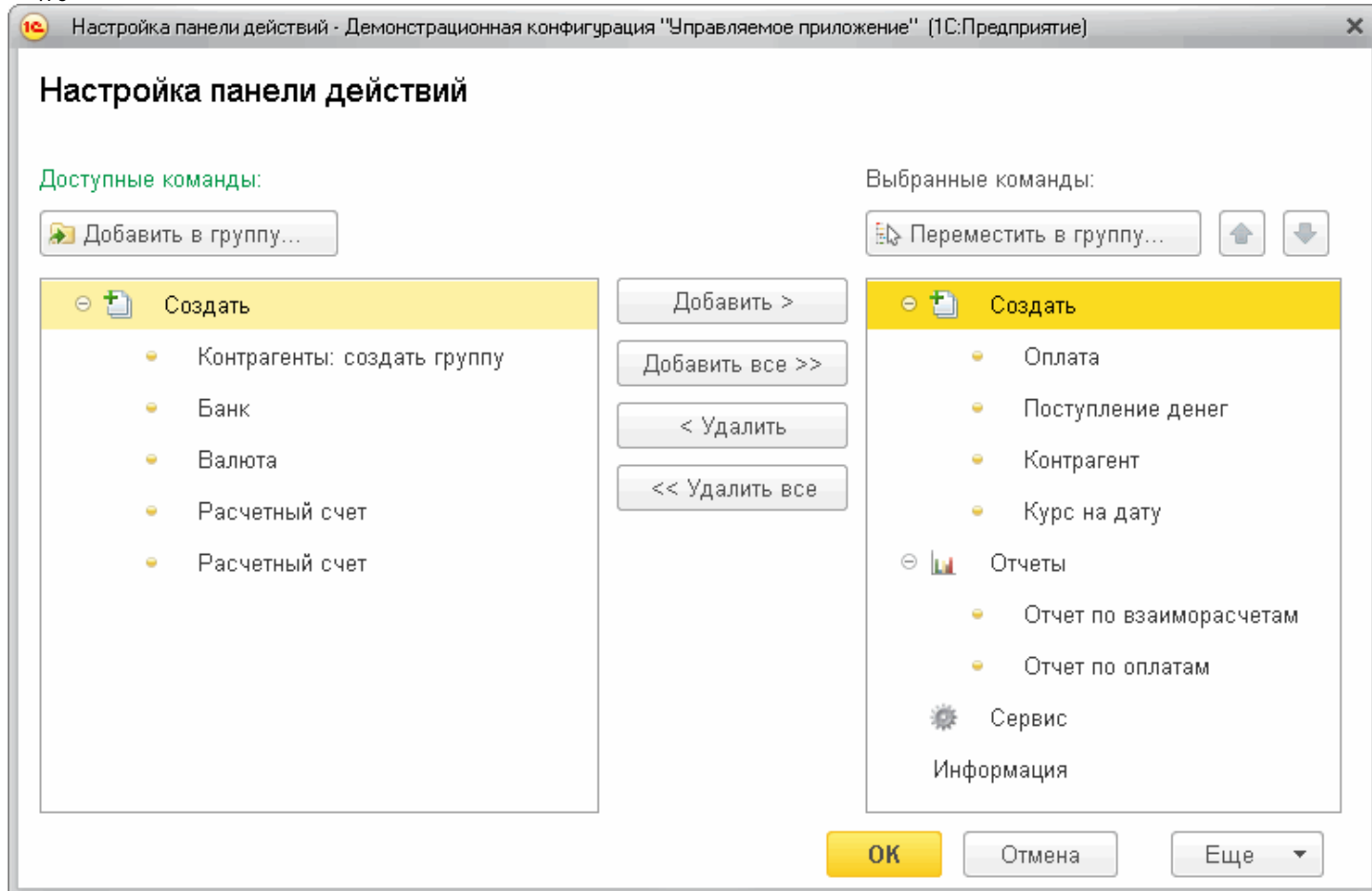


Рис. 189. Настройка панели действий

- команды создания новых объектов (например, [Контрагент](#)) – в стандартной группе [Создать](#);
- команды открытия отчетов (например, [Отчет по взаиморасчетам](#)) – в стандартной группе [Отчеты](#);
- команды вызова обработок – в стандартной группе [Сервис](#);
- другие команды, созданные в конфигурации.

Настройка панели действий производится аналогично настройке панели навигации (см. выше).

8.1.6. Редактор панелей

Чтобы открыть редактор панелей, следует выполнить команду главного меню [Вид – Настройка панелей](#).

На экране отобразится форма редактора панелей.

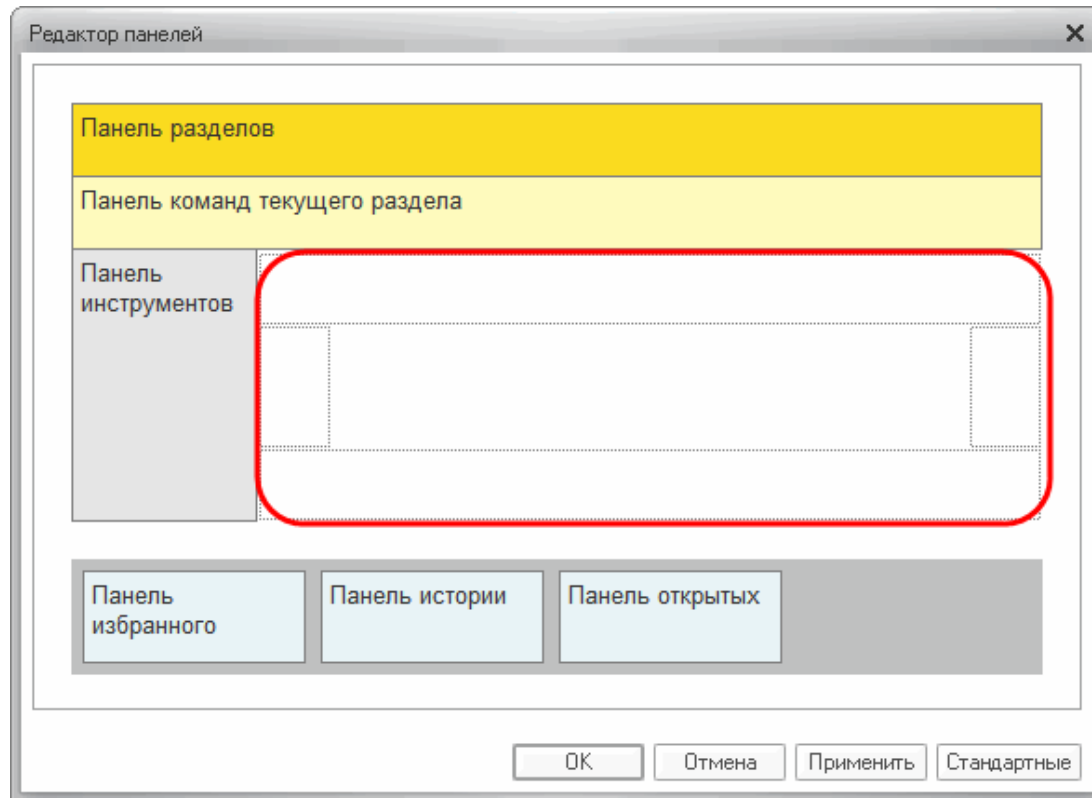


Рис. 190. Редактор панелей

В нижней (серой) части окна находятся панели, которые на данный момент не отображаются в окне программы, но могут быть добавлены в интерфейс.

В верхней части окна пунктирными рамками отображены области, в которые можно перетащить панели.

Чтобы добавить панель в интерфейс, нужно перетащить ее мышью в область доступных. Если в требуемой области уже отображается другая панель, то новая панель будет отображаться рядом с отображаемой. С помощью мыши можно изменять область отображения уже отображаемых панелей.

Чтобы убрать панель из интерфейса, нужно перетащить ее с помощью мыши из области отображаемых панелей в область неиспользуемых. Чтобы вернуть расположение панелей по умолчанию, нажать кнопку [Стандартные](#).

Для просмотра результата изменения расположения панелей нажать кнопку [Применить](#).

Для завершения редактирования отображения панелей следует нажать кнопку [ОК](#).

СОВЕТ. Чтобы скрыть сразу все отображаемые панели, в главном меню нужно выбрать [Вид – Скрыть все панели](#) или нажать [Ctrl + Shift + `](#). При этом скроются все панели, кроме панели инструментов. Чтобы вернуть панели, повторно выбрать команду [Скрыть все панели](#) или использовать комбинацию клавиш.

8.1.7. Область системных команд основного и вспомогательного окна

Чтобы добавить или удалить кнопки на стандартной командной панели основного или вспомогательного окна, следует нажать кнопку [Другие кнопки](#), в появившемся меню навести курсор мыши на пункт [Добавить или удалить кнопки](#) и в отобразившемся меню установить или снять пометку напротив названия команды.

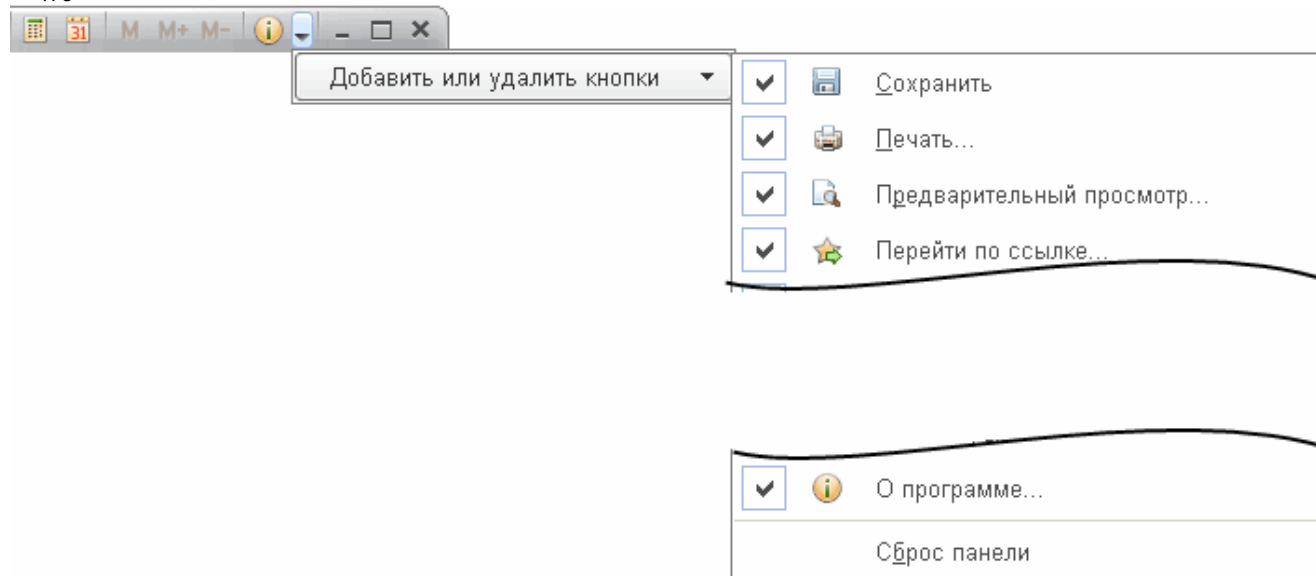


Рис. 191. Настройка командной панели окна

Команда, для которой пометка установлена, отобразится в области системных команд окна.

Команда, для которой пометка снята, скроется из области.

Для возврата панели первоначального вида используется пункт [Сброс панели](#).

8.2. Настройка формы

При работе в системе «1С:Предприятие» может быть доступна пользовательская настройка формы. Если иного не было установлено в конфигурации, то при вызове команды [Еще – Изменить форму](#) на командной панели формы откроется окно редактора.

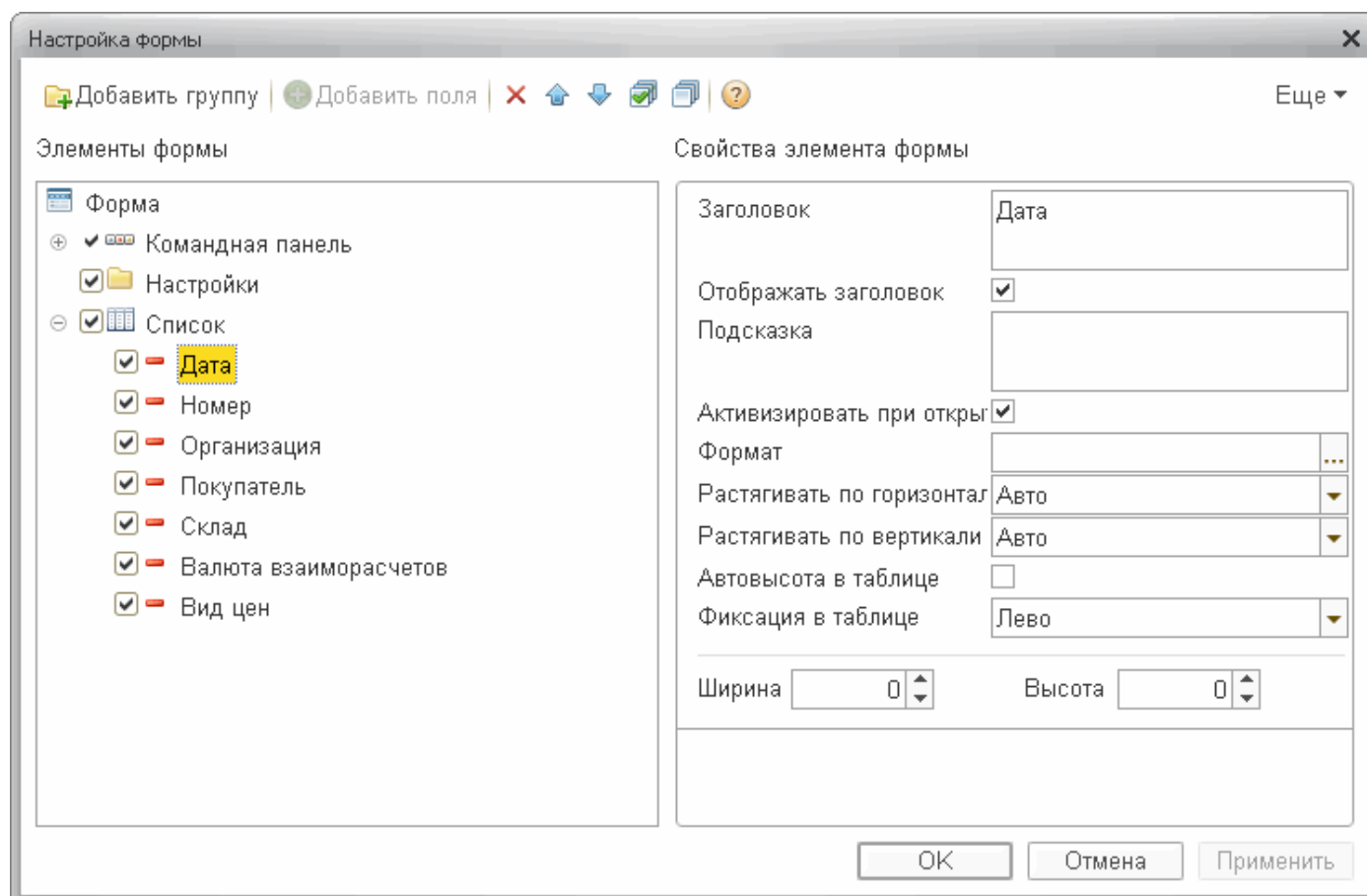


Рис. 192. Настройки формы

В левой части окна в виде дерева отображается структура элементов формы. В правой части – список свойств выбранного элемента формы.

Содержание и внешний вид формы определяются набором составляющих форму элементов нескольких видов.

- группы (их несколько видов),
- таблицы,
- поля (нескольких видов),
- кнопки.

Эти элементы обладают свойствами, влияющими на способ их отображения на форме. Иерархическая организация элементов определяет их взаимное расположение на форме.

Пользователь может изменять состав, взаимное расположение, группировку и свойства элементов.

Перестановка элементов. Чтобы изменить положение элементов в иерархии, следует использовать кнопки [Вверх](#) и [Вниз](#) или перетащить элемент на нужное место с помощью мыши.

Свойства элемента. Для каждого элемента свойства настраиваются в свойствах элемента формы. Описание каждого свойства отображается в нижней части панели при выборе свойства.

Настройка видимости. Для изменения видимости элементов на форме (скрыть или показать) используются флажки напротив элементов. Следует отметить, что нельзя управлять видимостью корневого элемента [Форма](#) и кнопки настройки формы. Также не может быть изменена видимость командной панели, на которой расположена кнопка [Изменить форму...](#)

Создание групп. Чтобы добавить группу на форму, нужно выбрать в дереве структуры корневой элемент [Форма](#) и нажать кнопку [Добавить группу](#). В свойствах группы установить требуемый вид ([Обычная группа](#), [Страницы](#), [Командная панель](#)).

Добавление полей. Для некоторых полей может быть доступна кнопка [Добавить поля](#). Это означает, что у данных, отображаемых в поле/колонке, есть реквизиты, которые также могут быть отображены в поле/колонке. Для того чтобы добавить на форму реквизиты полей, следует выбрать элемент формы, нажать кнопку [Добавить поля](#) и в появившемся окне выбрать требуемые реквизиты.

Применение настроек. Применение настроек происходит при выходе из редактора настройки с помощью кнопки [ОК](#) или по команде [Применить](#). При этом настройки сохраняются в информационной базе, чтобы впоследствии их можно было применить при открытии формы.

В процессе настройки можно отменить изменения, сделанные как в данном сеансе работы, так и ранее, и вернуться к настройкам формы, заданным в конфигурации. Для этого используется команда [Установить стандартные настройки](#) меню [Еще](#). Эта команда не вызывает немедленного изменения формы. После ее выполнения настройку можно продолжить. Применение настроек происходит только при нажатии кнопки [ОК](#) или [Применить](#).

8.2.1. Пример настройки формы

Рассмотрим процесс настройки формы на примере формы документа [Продажа](#).

🏠
⬅️ ➡️
☆ Продажа 000000012 от 04.09.2012 10:20:25
✕

Главное
Регистр взаиморасчетов с контрагентами
Регистр продаж
Регистр товарных запасов

Провести и закрыть
Записать
Провести
Еще ▾
?

Номер:

Дата: 📅

Организация: ▾ 🗑

Покупатель: ▾ 🗑

Склад: ▾ 🗑

Вид цен: ▾ 🗑

Валюта взаиморасчетов: ▾ 🗑

Товары (2)
Прочее

Добавить
Подбор
Еще ▾

N	Товар	Цена	Количество	Сумма
1	Колбаса	250,00	10	2 500,00
2	Сметана	50,00	11	550,00
Количество (итог): 21		Сумма (итог): 3 050,00		

Рис. 193. Форма документа «Продажа»

Предположим, что в таблице **Товары** требуется отобразить артикул товара и поставщика, от которого был получен товар. Для этого на форме документа вызовем пункт **Еще – Изменить форму**.

В форме настроек раскроем группу **Группа табличные части – Товары**, выберем поле **Товар** и нажмем кнопку **Добавить поля**, как показано на рисунке ниже.

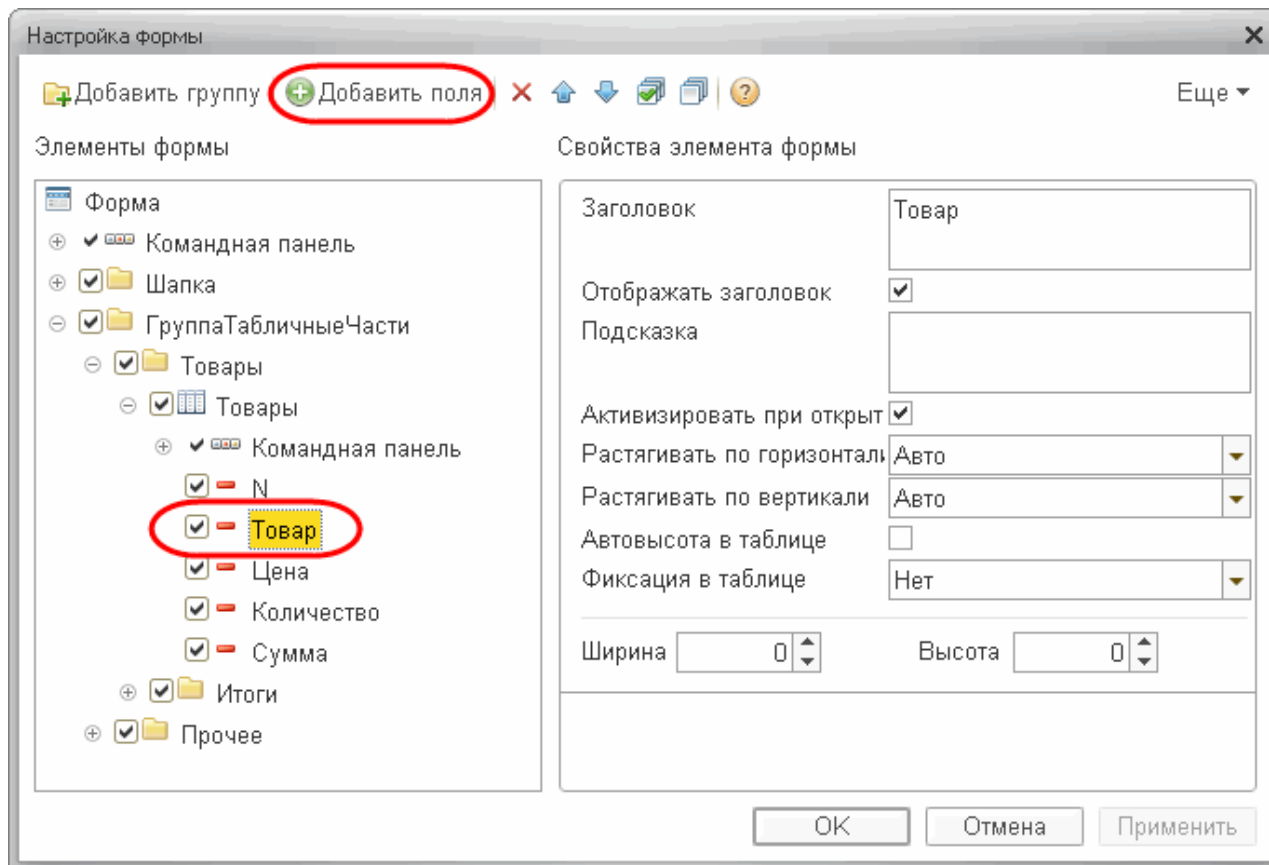


Рис. 194. Добавление полей в настройках формы

В появившемся окне выберем поля [Артикул](#) и [Поставщик](#), как показано на рисунке ниже.

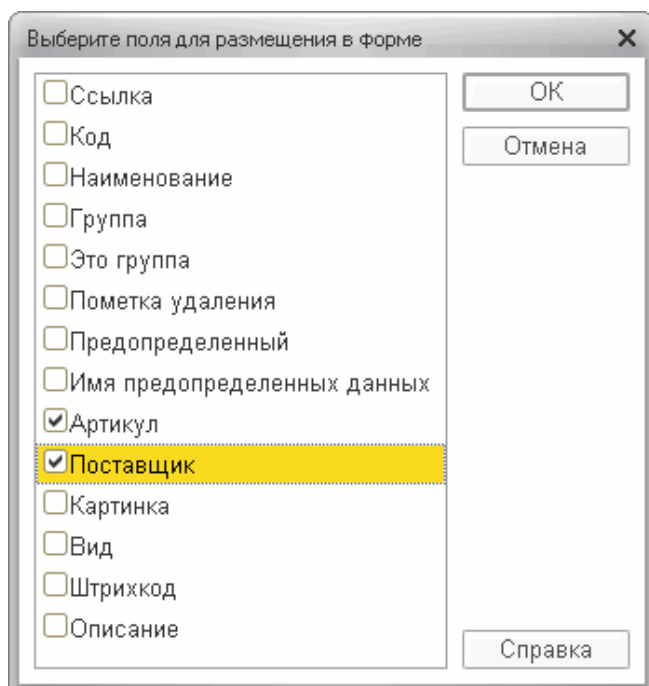


Рис. 195. Выбор полей

И нажмем **OK**. В диалоге настройки формы также нажмем **OK** или **Применить**. Форма документа [Продажа](#) примет вид:

☆ Продажа 000000012 от 04.09.2012 10:20:25 ×

[Главное](#) Регистр взаиморасчетов с контрагентами Регистр продаж Регистр товарных запасов

Номер: Вид цен:

Дата: Валюта взаиморасчетов:

Организация:

Покупатель:

Склад:

Товары (2) Прочее

N	Товар	Артикул	Поставщик	Цена	Количество	Сумма
1	Колбаса	Кол67	Животноводство ООО	250,00	10	2 500,00
2	Сметана	ПРД-0001	Животноводство ООО	50,00	11	550,00
Количество (итог): 21						Сумма (итог): 3 050,00

Рис. 196. Форма документа «Продажа»

Предположим, что требуется отображать, из какого региона поставщик товара. Для этого в настройках формы выберем поле **Поставщик** и нажмем кнопку **Добавить поля**. В открывшемся окне выберем поле **Регион** и нажмем **ОК**.

Структура таблицы примет следующий вид:

- ☐ Группа табличные части
 - ☐ Товары
 - ☐ Товары
 - + Командная панель
 - N
 - Товар
 - Артикул (Товар)
 - Поставщик (Товар)
 - Регион (Товар, Поставщик)
 - Цена
 - Количество
 - Сумма

Рис. 197. Структура таблицы в документе «Продажа»

Сгруппируем колонки таблицы. Для этого в таблице **Товары** создадим две группы (нажатием кнопки **Добавить группу**). В одной группе будет отображаться информация о товаре: поля **Товар** и **Артикул**. В другой – информация о поставщике: поля **Поставщик** и **Регион**. Установим порядок отображения полей в колонках с помощью стрелок панели формы настройки.

Структура таблицы документа примет вид:

- ⊖ Группа Табличные Части
 - ⊖ Товары
 - ⊖ Товары
 - ⊕ Командная панель
 - N
 - ⊖ Группа
 - Товар
 - Артикул (Товар)
 - ⊖ Группа1
 - Поставщик (Товар)
 - Регион (Товар, Поставщик)
 - Цена
 - Количество
 - Сумма
 - ⊕ Итоги

Рис. 198. Структура таблицы в документе «Продажа»

А измененная форма будет выглядеть, как показано на рисунке ниже.

🏠
⏪ ⏩
☆ Продажа 000000012 от 04.09.2012 10:20:25
×

Главное
Регистр взаиморасчетов с контрагентами
Регистр продаж
Регистр товарных запасов

Провести и закрыть
Записать
Провести
Создать на основании ▾
Печать ▾
Еще ▾
?

Номер:

Дата:

Организация:

Покупатель:

Склад:

Вид цен:

Валюта взаиморасчетов:

Товары (2)
Прочее

Добавить
Подбор
Еще ▾

N	Товар	Поставщик	Цена	Количество	Сумма
	Артикул	Регион			
1	Колбаса	Животноводство ООО	250,00	10	2 500,00
	Kol67	Алтай			
2	Сметана	Животноводство ООО	50,00	11	550,00
	ПРД-0001	Алтай			

Количество (итог): 21
Сумма (итог): 3 050,00

Рис. 199. Форма документа «Продажа»

Упростим представление формы, скрыв редко используемые команды.

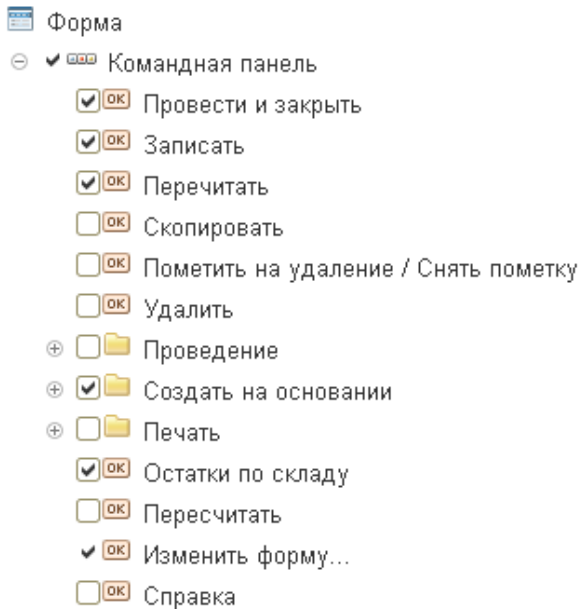


Рис. 200. Установка невидимости для кнопки «Пересчитать»

Командная панель формы документа примет следующий вид:



Рис. 201. Команды документа «Продажа»

По документам **Продажа** проходят в основном неделимые предметы. И значит, точность до двух знаков после запятой в колонке **Количество (Итог)** нам не нужна.

В настройках формы выберем поле **Количество (итог)** и в свойстве **Формат** нажмем кнопку выбора, как показано на рисунке.

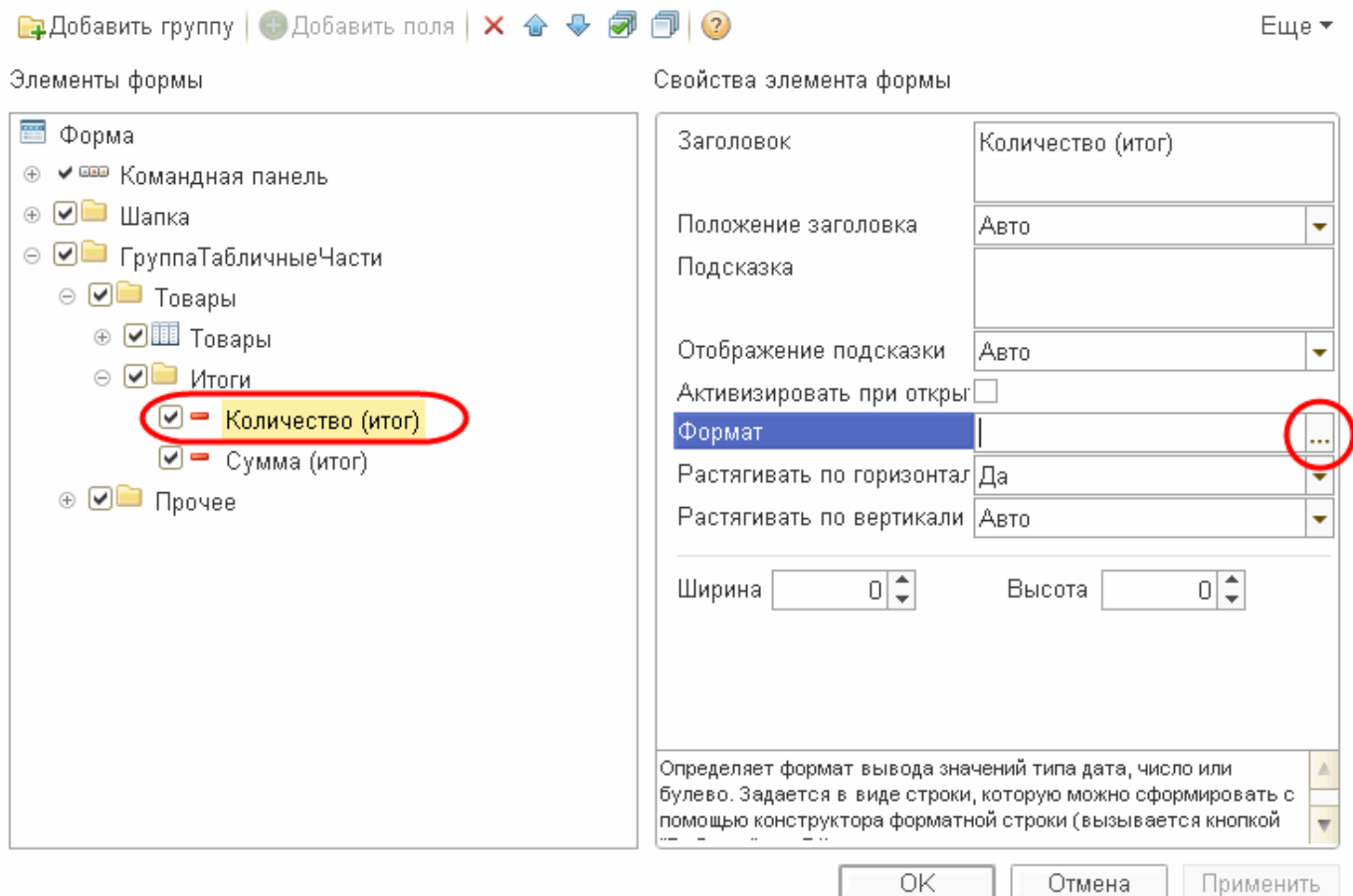


Рис. 202. Вызов конструктора форматной строки

В открывшемся окне конструктора форматной строки установим флажок напротив параметра [Точность](#) и нажмем [OK](#). Применим настройки формы.

После применения всех перечисленных выше настроек документ [Продажа](#) будет выглядеть следующим образом:

Продажа 000000012 от 04.09.2012 10:20:25

Главное Регистр взаиморасчетов с контрагентами Регистр продаж Регистр товарных запасов

Провести и закрыть Записать Создать на основании Остатки по складу Еще

Номер: 000000012 Вид цен: Мелкооптовая

Дата: 04.09.2012 10:20:25 Валюта взаиморасчетов: Рубли

Организация: ООО "Все для дома"

Покупатель: Магазин "Мясная лавка"

Склад: Большой

Товары (2) Прочее

Добавить Подбор Еще

N	Товар	Поставщик	Цена	Количество	Сумма
	Артикул	Регион			
1	Колбаса	Животноводство ООО	250,00	10	2 500,00
	Кол67	Алтай			
2	Сметана	Животноводство ООО	50,00	11	550,00
	ПРД-0001	Алтай			

Количество (итог): 21 Сумма (итог): 3 050,00

Рис. 203. Результирующая форма документа «Продажа»

Таким образом, в режиме [1С:Предприятие](#) можно группировать связанные по смыслу элементы, располагать их в удобном для себя порядке, отключать ненужные для работы элементы, изменять способ отображения элементов на форме.

8.3. Системные параметры

Диалог установки системных параметров предназначен для выбора внешнего вида программы и настройки режимов отладки разработчиком или администратором.

Для вызова диалога настройки параметров системы «1С:Предприятие» используется команда главного меню [Сервис – Параметры](#).

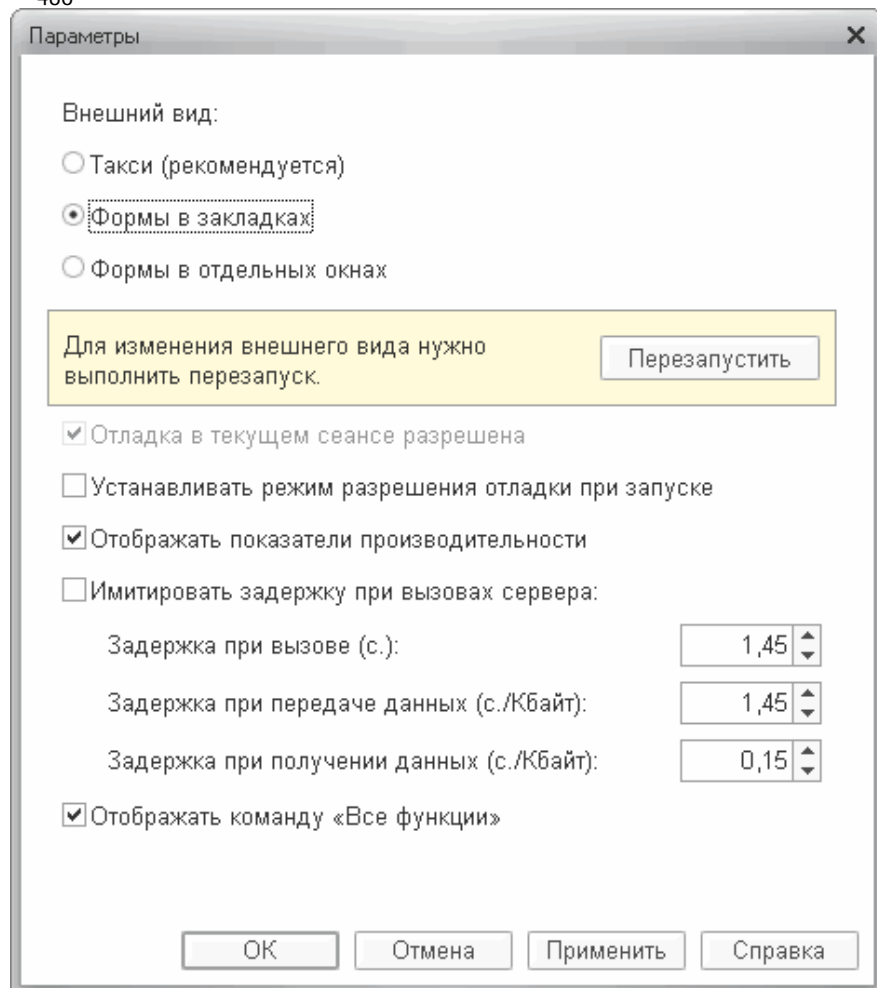


Рис. 204. Диалог настройки параметров

В диалоге возможна настройка следующих параметров:

- **Внешний вид** – чтобы параметр вступил в силу, требуется перезапуск программы. Использование данного параметра зависит от настроек программы. Доступны следующие варианты:
 - **Такси (рекомендуется)** – доступность данного внешнего вида зависит от настроек программы;
 - **Формы в закладках** – подробнее о данном режиме открытия форм см. книгу «1С:Предприятие 8.3. Руководство пользователя»;
 - **Формы в отдельных окнах** – подробнее о данном режиме открытия форм см. книгу «1С:Предприятие 8.3. Руководство пользователя»;
- **Отладка в текущем сеансе разрешена** – если флажок установлен, то в исполняемом экземпляре «1С:Предприятия» разрешена отладка;
- **Устанавливать режим разрешения отладки при запуске** – если флажок установлен, то отладка будет разрешена при последующем запуске экземпляров приложения;
- **Отображать показатели производительности** – если установлен флажок, то в отдельном специальном окне, которое по умолчанию расположено в левом нижнем углу экрана, будут отображаться показатели производительности;
- **Имитировать задержку при вызовах сервера** – данный параметр используется для проверки работы конфигурации в условиях соединения по медленным каналам;
- **Отображать команду "Все функции"** – если данный флажок установлен, то в меню добавляется команда **Все функции**, с помощью которой возможно выполнение различных действий. В список включены все объекты конфигурации независимо от того, включены ли они в интерфейс системы или нет. Данная настройка доступна, если пользователь имеет право на использование режима **Все функции**.

Подробнее о влиянии значений системных параметров см. в книге «1С:Предприятие 8.3. Руководство разработчика».

8.4. Управление окнами

В процессе работы помимо основного окна может быть открыто несколько вспомогательных. Платформа предоставляет механизм управления окнами и перехода между ними.

Для вызова диалога управления окнами нужно выбрать пункт **Окна – Все окна** главного меню. В диалоге отображается список окон, открытых к настоящему моменту.

Чтобы перейти к требуемому окну, нужно выбрать его в списке и нажать кнопку **Перейти**. Диалог **Окна** закроется автоматически.

Чтобы закрыть несколько окон, следует выбрать их с помощью мыши, удерживая клавишу **Shift**, и нажать кнопку **Закрыть окна**. Основное окно не может быть закрыто с помощью этого диалога.

Также для переключения между окнами одного сеанса используются сочетания клавиш **Ctrl + Tab**, **Ctrl + Shift + Tab**, **Ctrl + Shift + F6**. Для перехода к следующему активному окну следует нажать клавиши **Ctrl + Tab** или **Ctrl + F6**, для перехода к предыдущему окну – **Ctrl + Shift + Tab** или **Ctrl + Shift + F6**. При этом учитываются открытые модальные и блокирующие окна – окно, перекрытое блокирующим окном, активировано не будет.

8.4.1. Окна текстовых и табличных документов

Система «1С:Предприятие» помимо различных форм, предназначенных для учетных целей, предоставляет возможность работы с текстовыми и табличными документами (отчеты и созданные пользователем табличные документы). Для этих типов окон становятся доступными пункты меню, предоставляющие дополнительный сервис при просмотре и редактировании.

Пункты **Горизонтально разделить окно** и **Вертикально разделить окно** позволяют в пределах одного окна просматривать различные части данного окна. Пункты меню доступны, только если активным является окно с текстовым или табличным документом. Выбранное окно разделяется на две одинаковые части. Чтобы изменить соотношение частей, нужно подвести указатель мыши к разделителю окна. В зависимости от выбранного вида деления указатель мыши сменит свою форму на раздвоенную вертикальную или горизонтальную стрелку. Нажав левую клавишу и перемещая мышью, изменить размеры подокон.

Отмена деления окна редактора текстов. Чтобы отменить деление окна редактора текстов, нужно выбрать пункт **Окна – Удалить деление окна** или перетащить мышью разделительную линию к одному из краев окна редактора.

8.4.2. Восстановление положения окна

Положение, размер и состояние окна можно изменять. При закрытии последние параметры отображения сохраняются, и при повторном открытии окно открывается с последними значениями параметров показа.

Для восстановления первоначального положения, размера и состояния окна следует в главном меню выбрать пункт **Окна – Восстановить положение окна**. При этом запомненные значения параметров показа сбрасываются и восстанавливаются первоначальные, какие были при первом открытии.

Глава 9. Получение справочной информации

Информацию по работе с системой «1С:Предприятие» и по конкретным объектам и режимам можно получить, используя систему справочной информации (пункт [Справка](#) или клавиша **F1**). В главном меню для вызова помощи существует подменю [Справка](#).

При вызове справки из какого-либо режима отображается раздел справочной информации, соответствующий тому режиму, в котором сейчас работает пользователь. Пункт меню [Содержание](#) и клавиши **Shift + F1** позволяют обратиться к общему содержанию справочной информации.

9.1. Окно справки

Окно справочной информации состоит из двух частей. Вверху расположена командная панель, с помощью которой осуществляется навигация и поиск информации. Ниже командной панели отображается содержание справки.

Если текст не помещается в окне целиком, справа и внизу формы появляются линейки прокрутки, которые можно использовать для вывода на экран невидимой части.

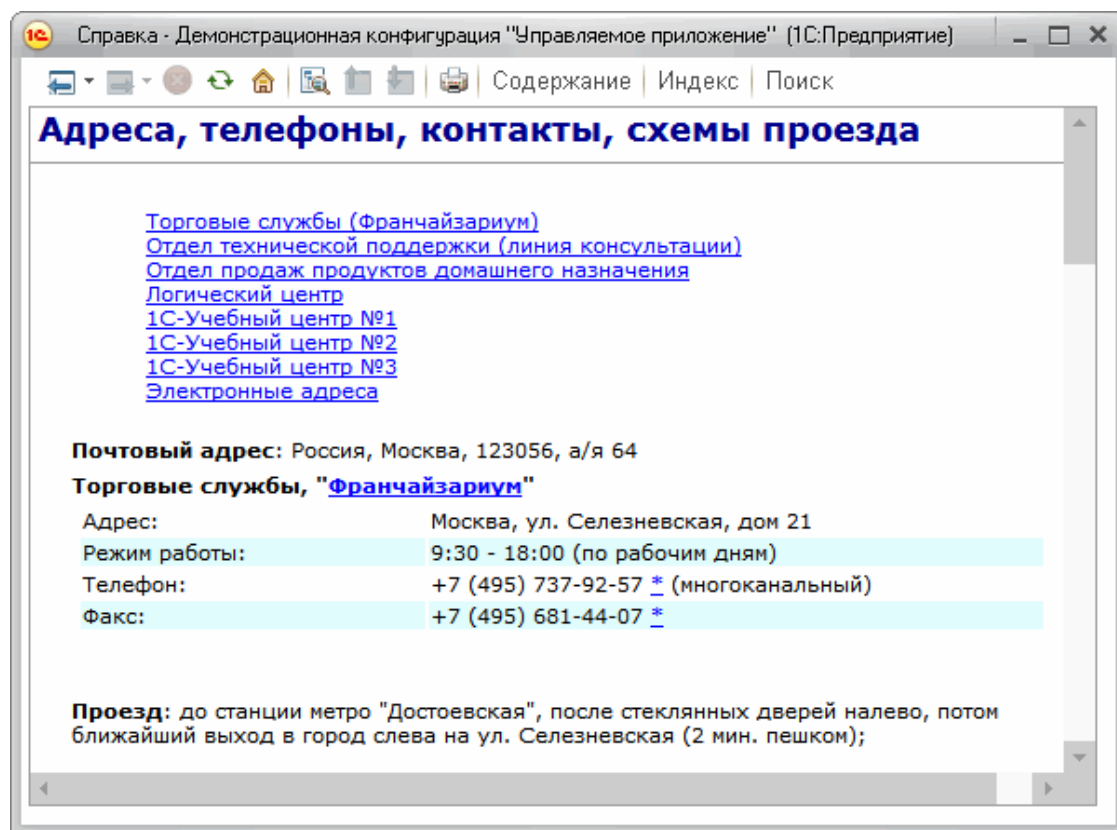


Рис. 205. Окно справочной информации

9.1.1. Просмотр справочной информации

Для поиска нужной главы можно воспользоваться содержанием. Для этого в форме [Справка](#) нужно нажать кнопку [Содержание](#).

В левой части окна появится дерево глав содержания.

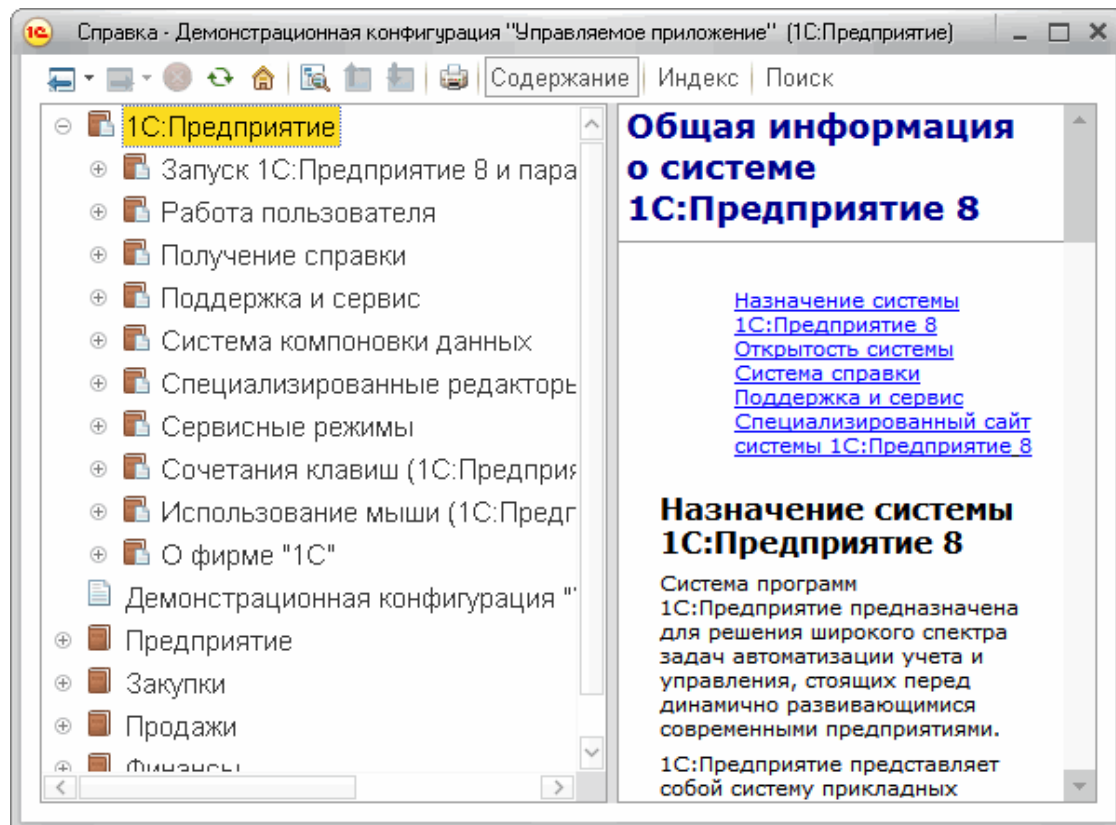


Рис. 206. Содержание справки

Дерево глав строится автоматически и включает в себя системную справку и справку по использованию конфигурации. Также справка может содержать информацию по работе с конкретными разделами программы.

Для удобства просмотра все дерево или отдельные ветви можно развернуть и свернуть.

Пиктограмма + (плюс) в узле ветви указывает, что эту ветвь можно развернуть. Для этого достаточно нажать пиктограмму или дважды нажать мышью наименование главы, расположенное возле пиктограммы. Ветвь откроет для просмотра следующий уровень, а пиктограмма + (плюс) изменится на – (минус). Свернуть ветвь дерева можно щелчком мыши на пиктограмме – (минус).

Чтобы развернуть сразу все ветви дерева, нужно выбрать клавишами или мышью наименование главы корневой ветки дерева и нажать клавишу * (звездочка на цифровой клавиатуре), которая расположена на цифровой клавиатуре.

На первом уровне в дереве располагаются названия разделов.

Последовательно раскрывая ветви дерева глав, можно перейти к описанию конкретного элемента данных.

Для перемещения по главам описания можно использовать дерево глав или кнопки панели управления окна [Справка](#).

Чтобы просмотреть текст интересующей главы в дереве, следует выбрать наименование главы мышью или с помощью клавиш управления курсором и нажать клавишу [Enter](#). В окне просмотра отобразится текст выбранной главы.

Для перемещения по тексту описания можно использовать клавиши управления курсором и клавиши [PgDn](#) и [PgUp](#).

При значительном объеме текста можно использовать линейки прокрутки, расположенные справа и внизу окна, для вывода на экран невидимой части описания.

Кнопки командной панели окна [Справка](#) позволяют быстро перемещаться от одной главы к другой.

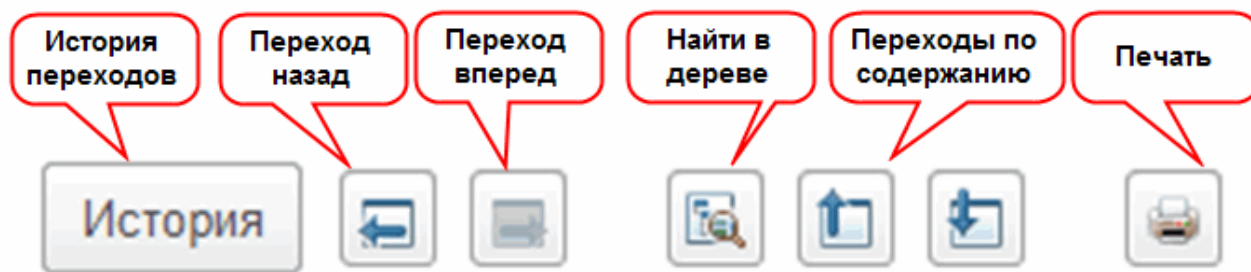


Рис. 207. Командная панель окна справочной информации

Кнопкой [Переход к следующему](#) осуществляется переход к следующей по порядку главе. В дереве глав при необходимости будет открыта соответствующая ветвь, а выделение будет перемещаться по названиям глав, показывая текущее положение в дереве. Для перехода к предыдущей по порядку главе используется кнопка [Переход к предыдущему](#).

После того как глава описания найдена и окно справки открыто, для поиска положения главы в дереве описаний используется кнопка командной панели [Найти текущий элемент в дереве](#).

Если в процессе просмотра выбирались несколько страниц, то с помощью команд [Переход вперед](#), [Переход назад](#) и [История](#) можно вернуться к просмотренным страницам.

Программа сохраняет историю просмотра справочных глав. Для возврата на предыдущую просмотренную главу или для перехода на следующую просмотренную главу в истории используются кнопки истории переходов.

9.1.2. Поиск справочной информации

Для поиска нужной информации в описании существует несколько способов, которые рассмотрены ниже.

Поиск по индексу. Окно [Справка](#) предоставляет поиск справочной информации по индексу. На командной панели нужно выбрать команду [Индекс](#). В левой части окна отобразится индекс разделов справки.

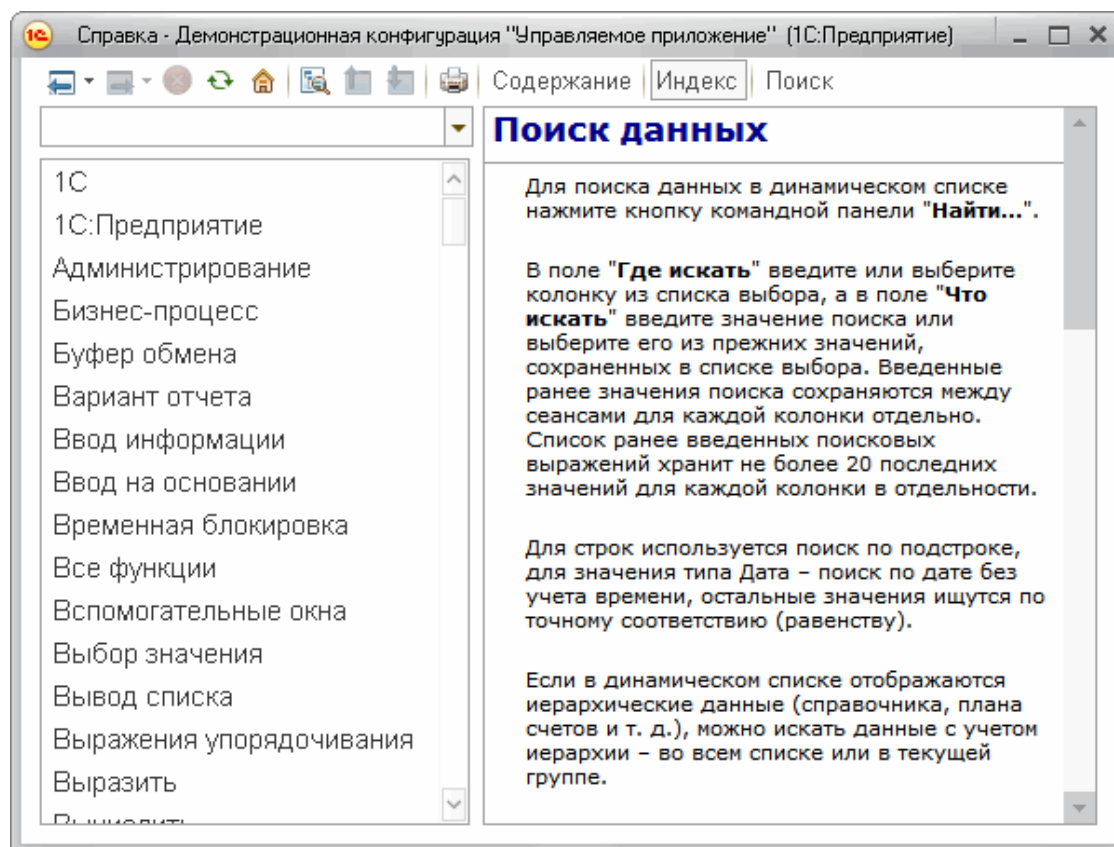


Рис. 208. Вызов индекса справки

В верхней строке находится поле ввода, предназначенное для написания наименования главы. Ниже расположен отсортированный список глав. Если начать набирать название главы, то по мере набора символов будет выделяться глава, первые символы названия которой совпадают с вводимыми.

Если нужная глава найдена, следует нажать клавишу [Enter](#). В окне [Справка](#) будет выведено содержимое справки

по данной главе.

Если по данной главе в содержании имеется несколько описаний, то их список выводится в окно [Выбор главы](#).

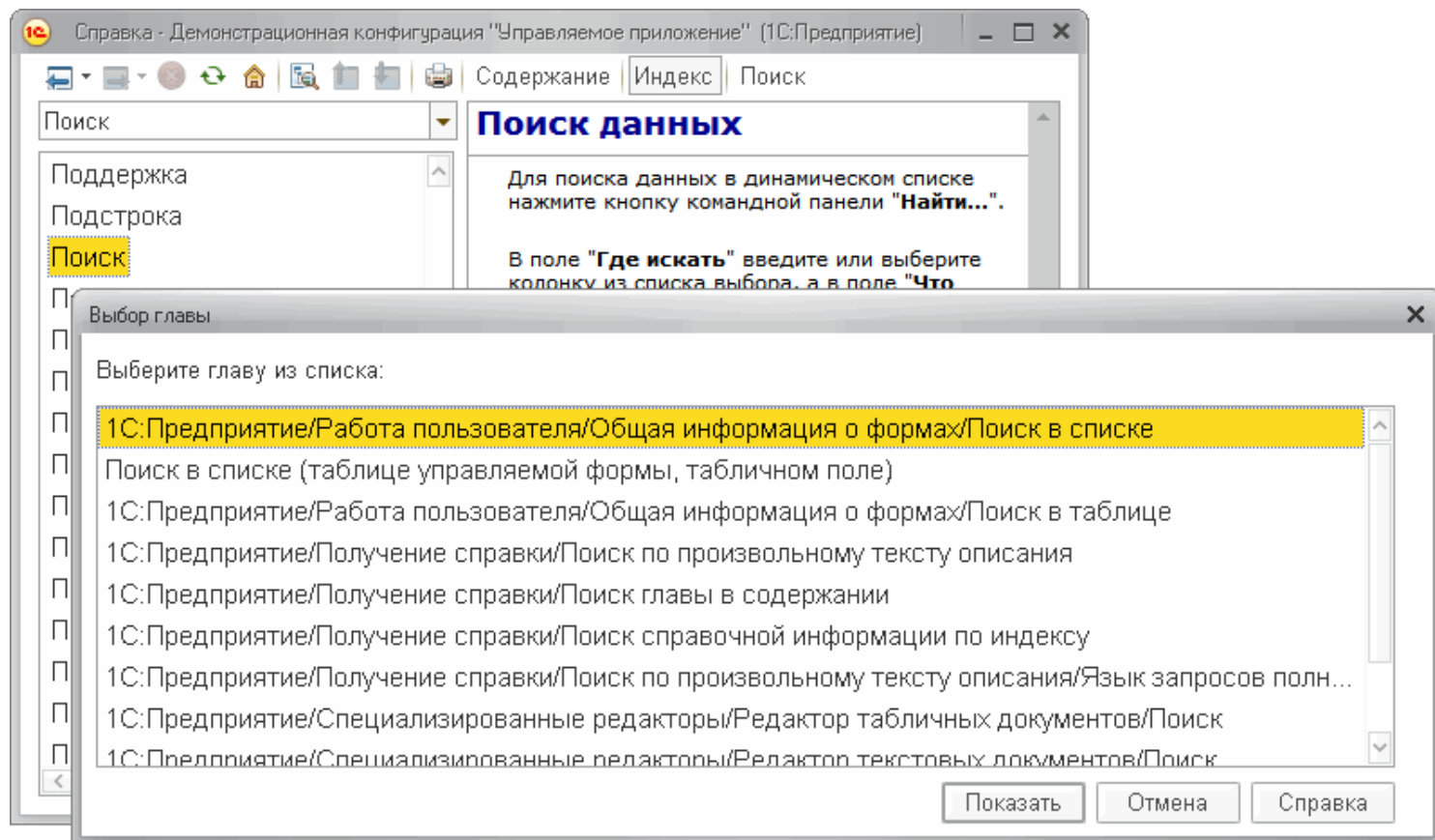


Рис. 209. Выбор главы справки

Следует указать нужную главу и нажать кнопку [Показать](#).

СОВЕТ. Также для вызова индекса справки используется сочетание клавиш [Alt + Shift + F1](#).

Поиск по справке. Для поиска глав, в которых встречается указанная строка, нужно выбрать пункт [Справка – Поиск по справке](#). На экран выводится окно, в котором выполняется поиск по произвольному тексту описания. В верхней части содержится поле для ввода строки поиска и поле списка найденных глав описаний.

Для начала поиска следует ввести текст и нажать кнопку [Найти](#) (или клавишу [Enter](#)). Программа выполняет поиск глав, в которых встречается введенный текст. Регистр ввода может быть любым, слова текста учитываются целиком (если не использован оператор *), с учетом морфологии. Допускается использование поисковых операторов (см. главу [«Формат поисковых выражений»](#), [здесь](#)).

В процессе ввода программа выводит список этих глав. Если введенный текст нигде не встречается, то ниже поля ввода программа выводит об этом сообщение.

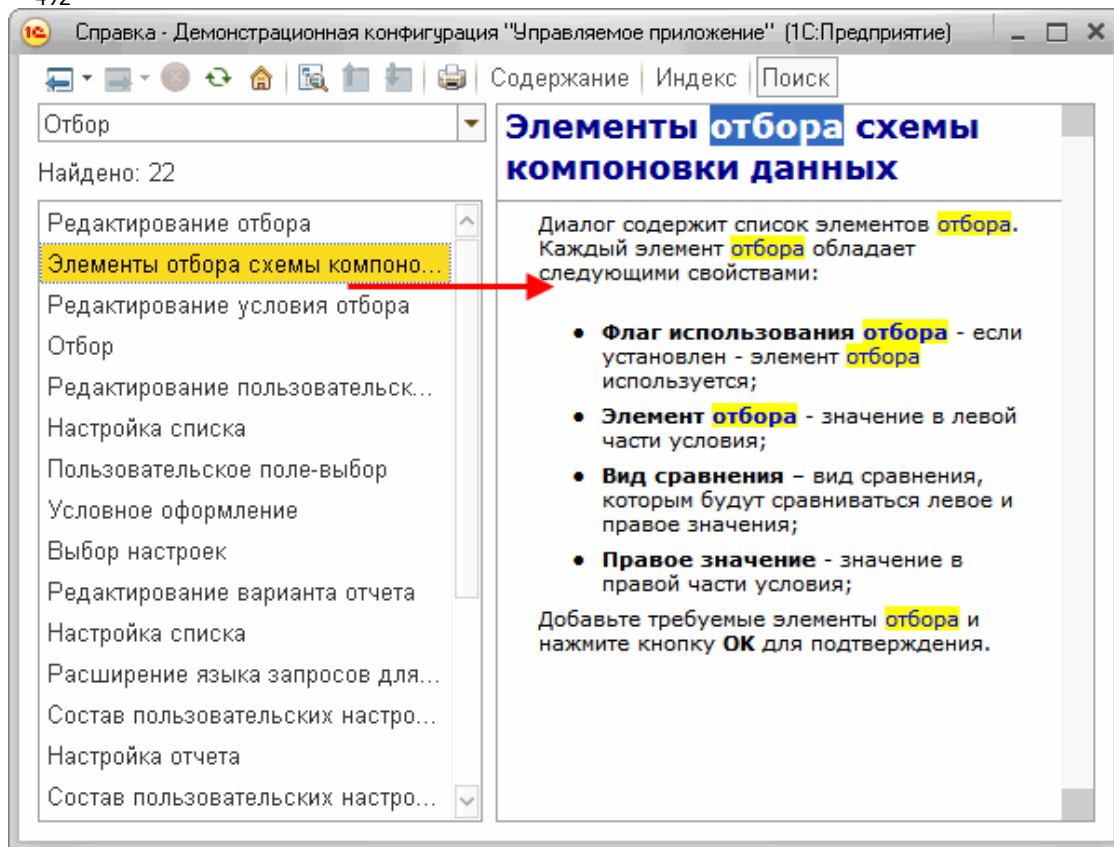


Рис. 210. Поиск по справке

При открытии главы программа покажет описание таким образом, чтобы было видно первое вхождение указанного текста.

Для просмотра главы нужно выбрать ее в списке и нажать клавишу **Enter**. Описание выбранной главы показывается в соседнем поле.

СОВЕТ. Также для вызова поиска по справке можно использовать сочетание клавиш **Alt + F1**.

Список операторов, используемых при поиске по справке, приведен в главе [«Формат поисковых выражений»](#) [здесь](#).

9.1.3. Печать описания

Для вывода любой главы описания на печать необходимо выполнить следующие действия:

1. В окне описания выбрать главу для печати и нажать в область с текстом, где отобразится текстовый курсор.
2. Выделить нужный текст и запомнить его в буфере обмена.
3. Вставить запомненную часть в любой текстовый редактор и распечатать.

9.2. { Окно «О программе»

Окно **О программе** вызывается выбором пункта **Справка – О программе** в главном меню или нажатием кнопки **i** области системных команд.



Рис. 211. Вызов окна «О программе»

Окно предоставляет общую информацию о системе «1С:Предприятие» и условиях конкретного запуска системы.

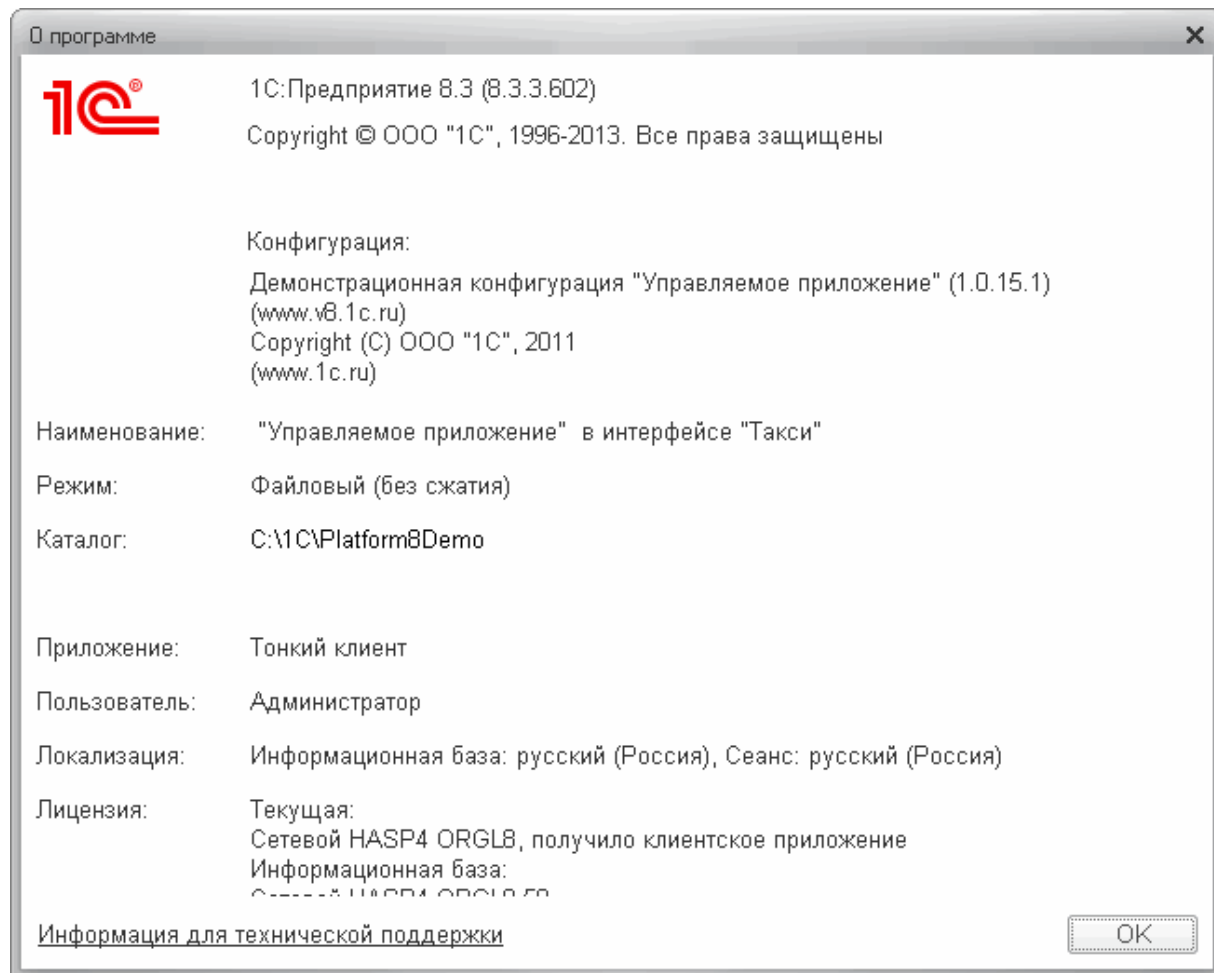


Рис. 212. Окно «О программе»

В окне отображаются следующие данные:

- информация о текущей версии системы «1С:Предприятие»;
- информация о конфигурации, с которой в настоящий момент работает пользователь (наименование и версия конфигурации);
- сведения об информационной базе, с которой в настоящий момент работает пользователь:
 - имя информационной базы, как оно задано в списке информационных баз диалога запуска «1С:Предприятия»;
 - формат базы данных (формат хранения информации) – файловый или клиент-сервер;
 - каталог, в котором располагается информационная база, и сведения о региональных установках информационной базы;
 - приложение (режим запуска: [Толстый клиент](#), [Тонкий клиент](#) и [Веб-клиент](#)). Также указывается скорость соединения и режим сжатия данных ([без сжатия](#), [обычное](#), [усиленное](#));
 - имя пользователя, который работает с программой в данный момент. Если задано полное имя текущего пользователя и оно не совпадает с кратким именем, имя текущего пользователя показывается в формате [Краткое имя](#); [Полное имя](#);
 - данные о локализации информационной базы и сеанса;
 - данные о текущей лицензии клиента и сервера «1С:Предприятия», а также данные обо всех лицензиях, работавших с данной информационной базой и изменявших данную конфигурацию;
- гиперссылка [Информация для технической поддержки](#), при нажатии которой в новом окне отображается информация о программе и журнал ошибок.

Приведенную в окне информацию можно копировать в буфер обмена с помощью команды контекстного меню.

Глава 10. Формат поисковых выражений

Поиск может осуществляться по нескольким словам, с использованием поисковых операторов и поиском по точной фразе.

По умолчанию поиск с учетом синонимов и нечеткий поиск не производятся. Для выполнения поисков этих видов следует использовать соответствующие операторы.

В данной таблице приведены поисковые операторы, которые можно использовать как при поиске по справке, так и при полнотекстовом поиске в данных. Некоторые операторы можно использовать только для одного вида поиска (см. описание конкретных операторов в таблице).

Оператор	Пример	Описание
Пробел И AND &	запись документ запись И документ запись AND документ запись & документ	Будут найдены все разделы, содержащие и «запись», и «документ» с учетом морфологии
ИЛИ OR ,	запись ИЛИ документ запись OR документ запись документ запись , документ	Будут найдены все разделы, содержащие хотя бы одно из слов «запись» или «документ» с учетом морфологии
НЕ NOT ~	закрытие НЕ месяц закрытие NOT месяц закрытие ~ месяц	Будут найдены все разделы, содержащие слово «закрытие», но не содержащие слова «месяц» с учетом морфологии. Использование знака ~ в начале строки не допускается
РЯДОМ/[±]n NEAR/[±]n	Пример 1: право РЯДОМ/3 пользователь Пример 2: право РЯДОМ/+3 пользователь Пример 3: право РЯДОМ/-3 пользователь	Поиск раздела, содержащего указанные слова с учетом морфологии на расстоянии n слов между словами. Знак указывает, в каком направлении от первого слова будет искаться второе слово («+» – после первого; «-» – до первого слова). Если знак не указан, то будет найден текст, содержащий указанные слова на дистанции n слов друг от друга. Порядок слов не имеет значения. В примере 1 будут найдены разделы, в которых слово «пользователь» находится не более трех слов до или после слова «право». В примере 2 будут найдены разделы, в которых слово «пользователь» находится не более трех слов после слова «право». В примере 3 будут найдены разделы, в которых слово «пользователь» находится не более трех слов перед словом «право»
РЯДОМ NEAR	Библиотека РЯДОМ имени РЯДОМ Достоевского	Краткая форма. Запрос в короткой форме найдет разделы, в которых слова встречаются не больше чем на 8 слов друг от друга в любую сторону
«»	«проведение документа»	Поиск точной фразы с учетом морфологии (эквивалентно «проведение РЯДОМ/+1 документ»)
()	(проведение выписка) & (счета, документа)	Группировка слов (сколько угодно уровней вложенности)

*	доку*	<p>Поиск с использованием группового символа (замена окончания слова). Должно быть введено более одного значащего символа. Запрос «доку*» найдет «документ», «документировать», «документальный» и др. (Если в индексе поиска окажется более 300 слов, начинающихся с «доку», система попросит уточнить запрос.) В поиске по справке знак * может быть в любом месте слова и сколько угодно раз, например «*ОК*нт*».</p> <p>При полнотекстовом поиске поддерживается использование только одного символа «*», и он должен быть в конце слова. То есть запрос «арх*» найдет «архив», «археология»</p>
#	#Система Система#2	<p>Нечеткий поиск слов с заданным количеством отличий от указанного в строке поиска.</p> <p>Запрос «#Система» (эквивалентно запросу «Система#1») найдет «систама», «сивтема».</p> <p>Запрос «Система#2» найдет «ситтама», «сеттема».</p> <p>Данный оператор используется при полнотекстовом поиске и не используется при поиске по справке</p>
!	! красный кафель	<p>Поиск с учетом синонимов русского, английского и украинского языков. Оператор «!» ставится перед соответствующим словом. Пример: поиск «!красный кафель», найдет еще и «алый кафель», и «коралловый кафель».</p> <p>Данный оператор используется при полнотекстовом поиске и не используется при поиске по справке</p>

Если не указано никаких операторов (слова набраны через пробел), то программа осуществляет поиск всех слов из запроса с использованием оператора И.

ПРИМЕЧАНИЕ 1. Написание операторов **И (AND)**, **ИЛИ (OR)**, **НЕ (NOT)**, **РЯДОМ (NEAR)** допускается только в верхнем регистре.

ПРИМЕЧАНИЕ 2. Операторы не используются как унарные (в начале строки поиска). Например, нельзя сделать выбор всех глав, в которых отсутствует указанный текст.

ПРИМЕЧАНИЕ 3. Все символы в поле поиска, кроме символов поисковых операторов, букв и цифр, игнорируются. Для поиска спецсимволов, используемых в тексте, их следует заключать в кавычки. Например: **"РЯДОМ"**.

1С:Предприятие 8.3.6

Руководство пользователя

83.004.06 от 12-02-2015

Руководство пользователя (02)

Введение (06)

Глава 1. О системе (01)

Глава 2. Установка и обновление системы (07)

Глава 3. Начало работы (06)

Глава 4. Интерфейс системы (11)

Глава 5. Работа в формах (24)

Глава 6. Списки (20)

Глава 7. Работа с данными различных видов (08)

Глава 8. Отчеты (15)

Глава 9. Изменение варианта отчета (18)

Глава 10. Сервисные возможности (13)

Глава 11. Настройка программы (15)

Глава 12. Получение справочной информации (07)

Глава 13. Формат поисковых выражений (02)

Глава 14. Работа с веб-клиентом (11)

Глава 15. Режим низкой скорости соединения (01)

Глава 16. Режим обычного приложения (17)

Приложение 1. Язык выражений системы компоновки данных (21)

Приложение 2. Текстовый редактор (06)

Приложение 3. Редактор табличных документов (36)

Приложение 4. Редактор HTML документов (04)

Приложение 5. Редактор графической схемы (09)

Приложение 6. Редактор географической схемы (09)

Руководство пользователя

1С:ПРЕДПРИЯТИЕ 8.3

Руководство

пользователя

Москва

Фирма «1С»

2015

ПРАВО ТИРАЖИРОВАНИЯ
ПРОГРАММНЫХ СРЕДСТВ И ДОКУМЕНТАЦИИ ПРИНАДЛЕЖИТ ФИРМЕ «1С»

Приобретая систему «1С:Предприятие»,
вы тем самым даете согласие не допускать
копирования программ и документации
без письменного разрешения фирмы «1С»

© ООО «1С», 2014

Фирма «1С», Москва, 123056, а/я 64.

Отдел продаж: Селезневская ул., 21,

телефон: (495) 737-92-57,

факс: (495) 681-44-07.

E-mail: 1c@1c.ru

URL: www.1c.ru, www.v8.1c.ru

Группа разработки программ – А. Алексеев, А. Безбородов, Д. Бескоровайнов, П. Василец, А. Виноградов, А. Волков, И. Гольштейн, Е. Горностаев, Г. Дамье, А. Даровских, О. Дерут, Н. Евграфов, Б. Евтифеев, Д. Зарецкий, Д. Ивашов, С. Копиенко, Н. Корсаков, С. Кравченко, В. Кудрявцев, П. Кукушкин, А. Лакутин, М. Лейбович, Г. Леонтьев, А. Лехан, А. Макеев, А. Медведев, Е. Митрошкин, С. Мурзин, С. Нуралиев, Л. Онучин, М. Отставнов, Д. Павленко, А. Плякин, А. Рукин, Д. Русанов, Е. Силин, Д. Службин, А. Смирнов, В. Соколов, П. Солодкий, В. Сосновский, Д. Сысоенков, А. Топорков, В. Тунегов, А. Трубкин, В. Филиппов, А. Цилябин, В. Черемисинов, П. Чиков, А. Чичерин, А. Шевченко, А. Щербинин.

Документация – В. Байдаков, В. Дранищев, Е. Королькова, А. Краюшкин, И. Кузнецов, М. Лавров, А. Монищев, А. Плякин, М. Радченко.

Группа тестирования – А. Галочкин, Б. Зиатдинов, А. Лапин, Е. Медведев, С. Потапов.

Группа консультационной поддержки – С. Алексеева, О. Багрова, О. Баклушина, М. Белоконь, Э. Гарифуллина, В. Давыдова, О. Дмитренко, Л. Ермакова, М. Ершова, Ю. Жестков, С. Жолудев, М. Жолудева, О. Завальская, Н. Заявлиная, О. Колоскова, Г. Коробка, Ю. Лаврова, С. Лепешкина, С. Мазурин, С. Марков, Ю. Мисан, А. Павликов, И. Панин, О. Пехтерева, С. Постнова, А. Прокуровский, Г. Степаненко, Н. Степанов, Т. Токарева, Г. Ястребова.

Корректурa – Е. Семененко.

Верстка – О. Шестакова.

Наименование книги:	1С:Предприятие 8.3. Руководство пользователя

Номер издания:	83.004.06
Дата выхода:	12 февраля 2015 года

ЛИНИЯ КОНСУЛЬТАЦИЙ

Для пользователей системы «1С:Предприятие 8» услуги линии консультаций предоставляются по линии информационно-технологического сопровождения (ИТС).

Получение консультаций возможно только после регистрации программного продукта (для чего необходимо заполнить регистрационную анкету и выслать ее в фирму «1С») и оформления подписки на ИТС.

В стоимость комплекта поставки программного продукта включено обслуживание по линии ИТС в течение периода, указанного в вашей регистрационной анкете. Купон на бесплатное обслуживание по линии ИТС, а также конверт с маркой и адресом входят в состав приобретенного вами комплекта программ.

По истечении периода бесплатного обслуживания получение консультаций возможно только по договору на ИТС. Для заключения договора обращайтесь к партнерам фирмы «1С».

Телефон и адрес электронной почты для консультаций указаны на диске ИТС.

Линия консультаций работает с 9:30 до 17:30 по московскому времени, кроме суббот, воскресений и праздничных дней (по расписанию московских предприятий).

Обращаясь к линии консультаций, следует находиться рядом со своим компьютером, иметь под рукой настоящее руководство и свою половину регистрационной анкеты. Желательно заранее уточнить типы используемого компьютера и принтера.

Набрав телефон линии консультаций, вы должны услышать ответ дежурного сотрудника. После этого сообщите наименование вашей организации, номер вашего экземпляра системы (он обозначен на CD-ROM комплекта поставки и на вашей половине регистрационной анкеты) и, возможно, другую регистрационную информацию по запросу сотрудника линии консультаций. Названная вами информация будет проверена по данным, указанным в отосланной в фирму «1С» половине регистрационной анкеты.

Отвечая на возникшие у вас вопросы, дежурный сотрудник может воспроизвести возникшую ситуацию на своем компьютере. Он даст вам рекомендации сразу же или после обсуждения с разработчиками. Вам не нужно просить к телефону конкретных специалистов: мы отвечаем за работу всего персонала. Работа линии консультаций регистрируется, поэтому при повторных обращениях по сложным проблемам вы можете сослаться на дату и время предыдущего разговора.

МЫ ВСЕГДА РАДЫ ВАМ ПОМОЧЬ!

Введение

Настоящая книга является Руководством по использованию системы «1С:Предприятие» (далее – Руководство).

Данное руководство предназначено как для пользователей «1С:Предприятия», ответственных за работу с системой, – бухгалтеров, расчетчиков, так и для пользователей, работающих с ограниченным набором функций программы, – операторов, менеджеров, руководителей.

Так как работа системы во многом определяется используемой конфигурацией, для ознакомления с особенностями работы конкретных документов, отчетов и других элементов конфигурации необходимо изучить описание конфигурации, которая используется при работе с «1С:Предприятием».

Специалистам, занимающимся конфигурированием и администрированием системы, необходимо обращаться к книгам «1С:Предприятие 8.3. Руководство разработчика» и «1С:Предприятие 8.3. Руководство администратора».

Структура Руководства

В главе 1 излагается общий порядок работы с системой «1С:Предприятие».

В главе 2 кратко описывается процесс установки и обновления системы.

В главе 3 рассказывается о запуске приложения «1С:Предприятия».

В главе 4 описываются приемы навигации по программе.

В главе 5 содержатся сведения о работе с формами.

В главе 6 приводится описание порядка работы со списками в системе «1С:Предприятие».

В главе 7 содержится описание типовых приемов работы с различными видами данных.

В главах 8 и 9 излагаются приемы работы с отчетами и вариантами отчетов.

В главе 10 описываются сервисные возможности системы: использование ссылок и избранного, работа с историей, использование встроенного калькулятора и т. д.

В главе 11 рассматриваются возможности пользовательской настройки интерфейса, форм и параметров системы.

В главе 12 содержится информация по работе со справочной системой.

В главе 13 описывается формат поисковых выражений, используемый при поиске в программе.

В главе 14 содержатся сведения об особенностях при работе с веб-клиентом.

В главе 15 перечислены особенности работы с системой в режиме низкой скорости.

В главе 16 описываются особенности работы с системой в режиме «обычного» приложения.

Приложение 1 содержит описание языка выражений системы компоновки, который используется для создания пользовательских полей отчетов.

Также в приложениях вы найдете описание встроенных редакторов системы «1С:Предприятие»: в Приложении 2 – текстового редактора; в Приложении 3 – табличного редактора, который используется для просмотра, редактирования и печати выходных форм документов и отчетов; в Приложении 4 – редактора HTML-документов; в Приложении 5 – редактора графической схемы; в Приложении 6 – редактора географической схемы.

Приложения 1– 6 поставляются только в электронном виде в составе документации, публикуемой на

ИТС.

Данная книга описывает работу в интерфейсе [Формы в закладках](#) и [Формы в отдельных окнах](#).

Описание работы в интерфейсе [Такси](#) приводится в книге «1С:Предприятие 8.3. Руководство пользователя. Интерфейс «Такси».

Описание работы в мобильном приложении приводится в книге «1С:Предприятие 8.3. Руководство пользователя мобильного приложения».

Примечания

В тексте Руководства встречаются примечания нескольких видов:

- Совет – содержит альтернативный метод действия;
- Примечание – содержит дополнительную информацию;
- Внимание! – содержит информацию, необходимую для правильной работы системы.

Обозначения

Обозначения клавиш. Клавиши, например [Enter](#), [Esc](#), [Del](#), выделяются в тексте, как показано выше.

Для ссылок на клавиши управления курсором (клавиши со стрелками) будет использоваться фраза [клавиши управления курсором](#). Если необходимо упомянуть эти клавиши по отдельности, будут использоваться словосочетания [Стрелка вверх](#), [Стрелка вниз](#), [Стрелка вправо](#) и [Стрелка влево](#).

Комбинации клавиш. Когда для выполнения какого-либо действия необходимо нажать комбинацию клавиш, в тексте это обозначается, например, как [Ctrl + F3](#).

Обозначения кнопок. Наименования кнопок в диалогах и формах, закладках, ввода-редактирования данных будут даваться их названиями без кавычек, например [ОК](#), [Отмена](#), [Удалить](#) и так далее.

Описание действий. В Руководстве описание действия (открытие журнала документов, ввод документа, построение отчета и т. д.) производится выбором пункта меню (основного или активного окна, а также контекстного меню). Аналогичные действия в большинстве случаев можно выполнить с помощью команд командной панели. При выборе пункта меню следует обращать внимание на пиктограмму, располагающуюся слева от наименования пункта. Эта пиктограмма также размещена на кнопке командной панели с аналогичным действием. Изображения пиктограмм в тексте Руководства обычно не приводятся. Помимо пиктограммы следует обращать внимание на сочетание клавиш, с помощью которых также выбирается данное действие. Сочетание указывается справа от наименования пункта меню.

Выбор элемента обычно осуществляется двойным или одинарным нажатием левой кнопки мыши.

Что вы должны знать

Характер изложения данного Руководства предполагает, что вы знакомы с операционной системой компьютера, на котором работает система «1С:Предприятие» (ОС семейства [Microsoft Windows](#) и ОС семейства [Linux](#)), и владеете базовыми навыками работы в ней.

Вам должны быть знакомы следующие понятия и навыки (список приведен для операционной системы [Windows](#)):

- использование меню [Пуск \(Start\)](#) для вызова программ,
- приемы работы с окнами,
- работа с меню,
- использование управляющих элементов диалогов,

- стандартные диалоги,
- понятие буфера обмена операционной системы и приемы работы с ним,
- настройка операционной системы с помощью панели управления (**Control Panel**).

Если вы недостаточно хорошо владеете перечисленными выше понятиями и навыками, рекомендуем обратиться к документации по операционной системе.

При описании мобильного приложения предполагается, что вы знакомы с операционной системой мобильного устройства, на котором работает система «1С:Предприятие» (iOS, Android, далее – операционная система или ОС), и владеете базовыми навыками работы в ней.

Вам должны быть знакомы следующие понятия и навыки:

- установка и обновление приложения с помощью магазина приложений ОС,
- удаление приложения с мобильного устройства,
- приемы работы с пользовательским интерфейсом ОС,
- способы резервного копирования данных пользователя.

Книги документации

В состав документации входят книги по технологической платформе «1С:Предприятие 8.3»:

- «1С:Предприятие 8.3. Руководство пользователя». Книга содержит описание общих приемов работы с программными продуктами, созданными на базе платформы «1С:Предприятие 8.3». Данная книга не поставляется в составе продукта «1С:Предприятие 8.3. Версия для обучения программированию».
- «1С:Предприятие 8.3. Руководство разработчика». Книга необходима для изменения и настройки конфигурации под особенности учета конкретной организации, а также для разработки новых конфигураций.
- «1С:Предприятие 8.3. Руководство администратора». Книга описывает администрирование системы «1С:Предприятие 8.3», включая информацию об особенностях построения клиент-серверных систем. Данная книга не поставляется в составе продукта «1С:Предприятие 8.3. Версия для обучения программированию».
- Синтаксис встроенного языка и языка запросов представлен в книге «1С:Предприятие 8.3. Руководство разработчика». Описание объектной модели полностью включено в поставку в электронном виде (в разделах справки конфигуратора и синтакс-помощнике). Описание объектной модели также содержится в книге «1С:Предприятие 8.3. Описание встроенного языка», которая распространяется отдельно.

Внимание! Комплект поставки конкретного продукта может включать лишь некоторые из перечисленных книг документации.

Сопроводительные файлы комплекта поставки «1С:Предприятия 8.3»

В процессе установки платформы «1С:Предприятие 8.3» выполняется копирование на жесткий диск ряда сопроводительных файлов, содержащих описание изменений, реализованных в данной версии платформы, и инструкции по переходу с предыдущих версий.

Все сопроводительные файлы располагаются в каталоге установочных файлов конкретной версии системы «1С:Предприятие 8», в директории `\docs\ru`. Если при установке системы использовался каталог, предложенный по умолчанию, то эти файлы будут располагаться в директории `C:\Program Files\1cv8\НомерВерсии\docs\ru`. Здесь `НомерВерсии` означает номер установленной версии. Так, для версии 8.3.9.300 каталог будет иметь следующий вид: `C:\Program Files\1cv8\8.3.9.300\docs\ru`.

В файле `V8Update.htm` содержится список отличий текущей версии платформы от предыдущих версий и

особенности перехода на новую версию.

Директория [C:\Program Files\1cv8\НомерВерсии\licenses\](#) содержит лицензионное соглашение на «1С:Предприятие 8.3»:

- [1CEnterprise_ru.htm](#) – на русском языке,
- [1CEnterprise_en.htm](#) – на английском языке.

В директории [C:\Program Files\1cv8\НомерВерсии\licenses\3rd_party](#) содержатся лицензии на использованные компоненты.

ИТС – информационно-технологическое сопровождение

Фирма «1С» осуществляет платную методическую поддержку пользователей в рамках информационно-технологического сопровождения (ИТС) программ системы «1С:Предприятие».

Ежемесячные выпуски ИТС содержат большое количество постоянно обновляемой информации, позволяющей более эффективно использовать продукты системы «1С:Предприятие». Отметим наиболее важные составляющие ИТС:

- обновления технологической платформы «1С:Предприятие 8» и прикладных решений;
- методические материалы по технологической платформе «1С:Предприятие 8»;
- методические материалы по типовым прикладным решениям фирмы «1С»;
- конфигурация «Конвертация данных» для настройки правил обмена между информационными базами «1С:Предприятия», имеющими различную конфигурацию;
- система стандартов и методик разработки конфигураций для платформы «1С:Предприятие» (предназначена для ознакомления партнеров и пользователей фирмы «1С» с техническими и проектными решениями, используемыми при разработке типовых конфигураций на платформе «1С:Предприятие»);
- советы линии консультаций, помогающие пользователям «1С:Предприятия» получить ответы на наиболее часто встречающиеся вопросы и избежать типичных ошибок;
- информация по обучению работе с платформой «1С:Предприятие» и прикладными решениями фирмы «1С»;
- демонстрационные ролики программных продуктов, позволяющие получить первое представление о возможностях прикладных решений;
- справочники по заполнению налоговых деклараций в прикладных решениях фирмы «1С»;
- обширная подборка бухгалтерской периодики, включая текущие выпуски журналов и архивы;
- база аналитических обзоров законодательства и арбитражной практики;
- рекомендации по составлению квартальной и годовой отчетности в прикладных решениях фирмы «1С»;
- справочник типовых хозяйственных операций хозрасчетного предприятия;
- справочник по оформлению расчетов с персоналом по оплате труда и правовым аспектам трудовых отношений;
- справочная правовая система «Гарант» – полный набор нормативных документов законодательства Российской Федерации, в том числе по бухгалтерскому учету, налогам и предпринимательству.

Более подробно о проекте ИТС можно прочитать на сайте фирмы «1С»:

<http://www.1c.ru/rus/support/its/its.htm>.

Оставить заявку на демонстрацию дисков и приобретение подписки на ИТС можно на сайте фирмы «1С»:
<http://www.1c.ru/rus/support/its/zajavka.jsp>.

Оформить подписку на ИТС можно у партнеров фирмы «1С». Список партнеров, имеющих опыт массового обслуживания пользователей в рамках проектов ИТС, опубликован на сайте фирмы «1С»:
<http://www.1c.ru/rus/partners/service.jsp>.

Информация по «1С:Предприятию 8»

Адрес сайта: <http://v8.1c.ru/AllInfo>.

Данный сайт содержит набор ссылок на часто используемую информацию для следующих категорий пользователей:

- пользователи прикладных решений,
- разработчики прикладных решений,
- партнеры фирмы «1С».

Сайт системы программ «1С:Предприятие 8»

Адрес сайта: <http://v8.1c.ru>.

Сайт содержит информацию по технологической платформе системы «1С:Предприятие 8» и по типовым прикладным решениям, выпущенным фирмой «1С» на ее основе.

Пользовательский сайт

Адрес сайта: <http://users.v8.1c.ru>.

На сайте поддержки пользователей системы «1С:Предприятие 8» представлена информация о номерах версий платформы и конфигураций, дате их выхода, выпусках ИТС, на которых опубликовано обновление. По каждой версии представлена следующая информация:

- для технологической платформы:
 - отличия данной версии от предыдущих и особенности перехода;
 - ошибки, исправленные при выпуске данной версии;
 - дистрибутив обновления;
 - файл [readme.htm](#);
- для прикладных решений:
 - новое в релизе;
 - полный список изменений;
 - список основных изменений;
 - порядок обновления;
 - дистрибутив обновления;
 - номер версии платформы, необходимой для использования релиза конфигурации.

Также на пользовательском сайте публикуются рекомендации по администрированию системы «1С:Предприятие 8».

Основное преимущество, которое дает пользователям «1С:Предприятия 8» поддержка на данном сайте, – это возможность обновления технологической платформы и прикладных решений через Интернет до получения дисков ИТС.

Также на сайте публикуются дополнительные компоненты, используемые системой «1С:Предприятие 8» (например, СУБД PostgreSQL), и тестовые версии платформы и прикладных решений.

ВНИМАНИЕ! К сайту поддержки пользователей <http://users.v8.1c.ru/> имеют право доступа зарегистрированные пользователи программных продуктов системы «1С:Предприятие 8».

При этом пользователи продуктов, для которых предусмотрено обслуживание по линии информационно-технологического сопровождения (ИТС), должны иметь действующую подписку на ИТС.

Зарегистрироваться на сайте поддержки пользователей можно самостоятельно либо обратиться к партнеру фирмы «1С».

Для самостоятельной регистрации необходимо с главной страницы сайта перейти по ссылке [Самостоятельная регистрация пользователей по PIN-коду](#) и выполнить указанные на сайте действия.

В процессе регистрации потребуется PIN-код, входящий в комплект поставки программного продукта.

Если в комплекте поставки отсутствует PIN-код для регистрации на сайте, следует обратиться к партнеру фирмы «1С».

Для регистрации программного продукта необходимо заполнить регистрационную анкету на программный продукт (она является частью регистрационной карточки) и отправить ее в фирму «1С» по почте или факсу.

Для оформления подписки на ИТС можно обратиться к любому из сервис-партнеров фирмы «1С». Список партнеров, имеющих опыт массового обслуживания пользователей в рамках проектов ИТС, опубликован на сайте фирмы «1С»: <http://www.1c.ru/rus/partners/service.jsp>.

Глава 1. О системе

«1С:Предприятие» является универсальной проблемно-ориентированной системой для автоматизации задач учета, планирования и управления на предприятиях, а также решения персональных задач.

Функционирование системы «1С:Предприятие» делится на два разделенных во времени процесса: настройку (**конфигурирование**) и непосредственную работу пользователя по ведению учета или выполнению различных расчетов.

Конфигурирование. На этапе конфигурирования системы «1С:Предприятие» выполняется разработка прикладного решения в соответствии с особенностями конкретного предприятия. При этом определяются структуры объектов, способы их отображения, для пользователей разного уровня создаются роли, описывается интерфейс.

Возможности конфигурирования подробно описаны в книге «1С:Предприятие 8.3. Руководство разработчика».

Кроме того, на данном этапе могут выполняться административные действия, такие как ведение списка пользователей, установка параметров информационной базы, настройка журнала регистрации, обновление конфигурации и другое.

Работа пользователя. Работа пользователя с информационной базой осуществляется при запуске системы в режиме **1С:Предприятие**. При этом выполняется собственно функционирование системы в предметной области: осуществляется заполнение данных, формирование различных отчетов, выполнение различных регламентных расчетов и т. д.

Пользователь работает с данными, структура которых описана в конфигурации. При этом он использует алгоритмы, созданные на этапе конфигурирования.

В настоящем Руководстве содержится общее описание порядка работы пользователя при работе с системой «1С:Предприятие» в различных режимах: отображение и ввод данных, получение отчетов, пользовательская настройка интерфейса и другое. Пояснения особенностей работы с конкретными объектами, которые зависят от конфигурации, могут содержаться в описании самой конфигурации. В этом случае их можно получить, обратившись к режиму справки (см. главу «[Получение справочной информации](#)», [здесь](#)).

Глава 2. Установка и обновление системы

2.1. Установка системы

Система «1С:Предприятие» состоит из компонентов, предназначенных для разработки и использования решений (конфигураций) по ведению учета и автоматизации хозяйственной деятельности предприятий и персональных задач.

Установка «1С:Предприятия» может выполняться первично, когда производится полная установка выбранных компонентов, или повторно, когда есть необходимость изменения, переустановки или удаления некоторых или всех компонентов системы.

ВНИМАНИЕ! При установке тонкого клиента «1С:Предприятия» из самораспаковывающегося архива [1CEClientSetup.exe](#), полученного из сети Интернет, следует обратить внимание на отображаемую операционной системой информацию о цифровой подписи файла: в оригинальном файле будет отображаться информация от фирмы «1С» или доверенного провайдера.

При вставке диска в устройство чтения автоматически запустится программа установки. Если функция автозапуска отключена на конкретном компьютере, то следует запустить файл [autorun.exe](#) (в операционной системе Linux – [autorun](#)). Программа предлагает выбрать способ установки: быстрая установка и запуск или выборочная установка.

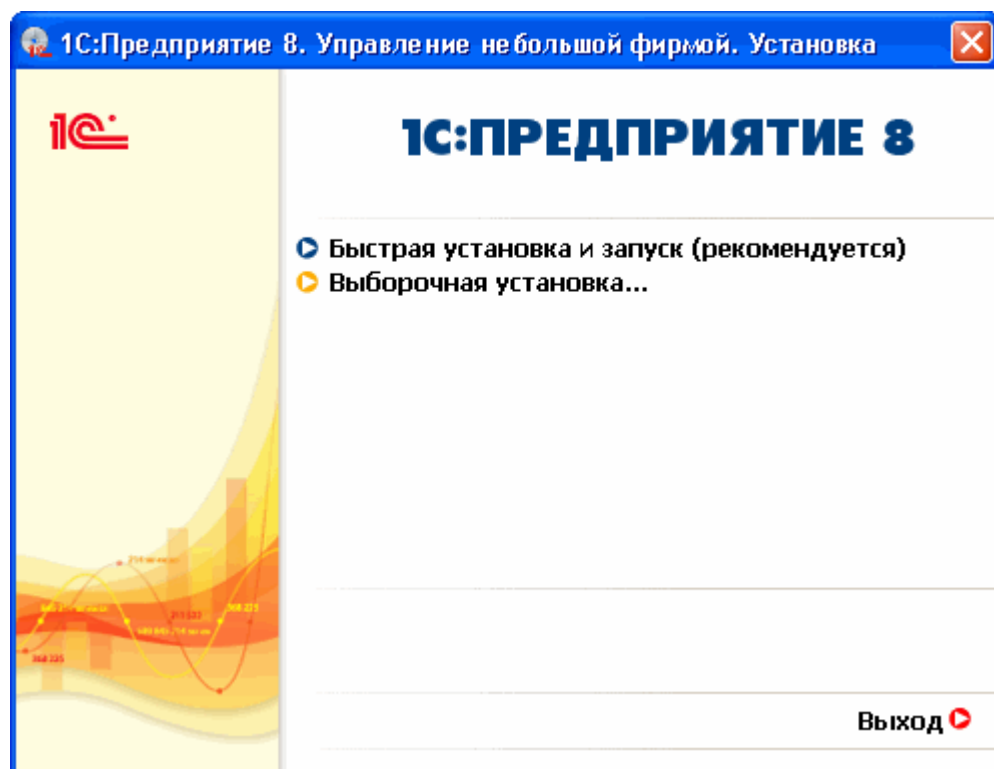


Рис. 1. Программа установки

При выборе пункта [Быстрая установка и запуск \(рекомендуется\)](#) программа установит платформу «1С:Предприятие» и конфигурацию, поставляемую на диске, в каталоги, используемые по умолчанию.

Если выбрать пункт [Выборочная установка](#) или если быстрая установка не предусмотрена для конкретной поставки, программа предложит указать компонент, который требуется установить.

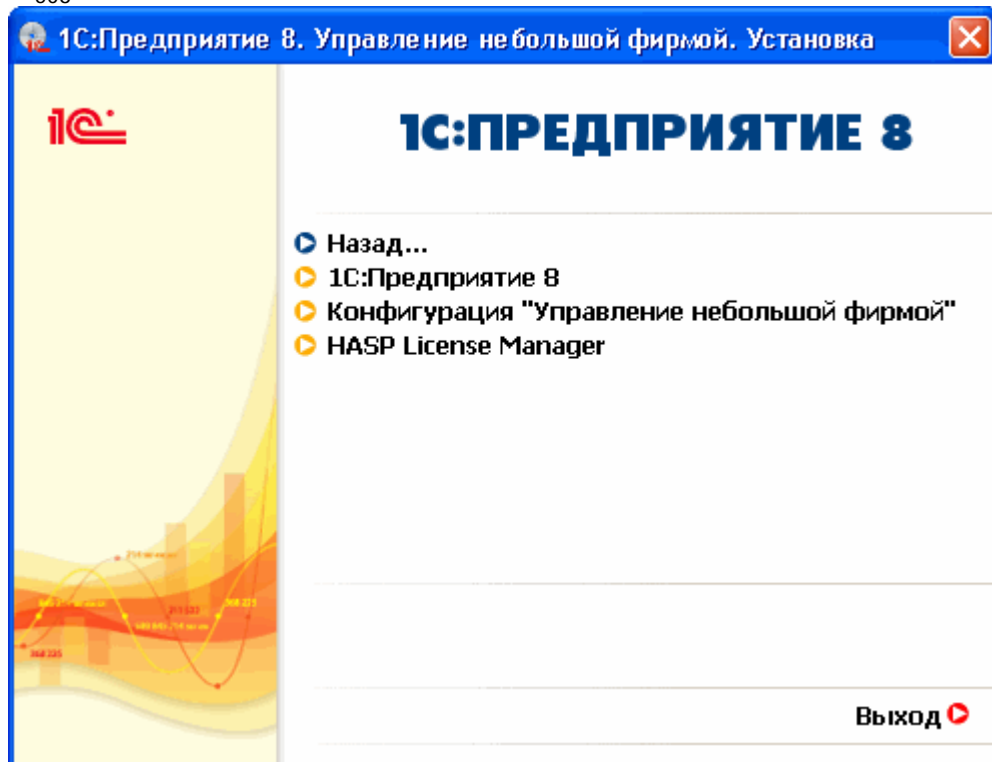


Рис. 2. Меню выборочной установки

2.1.1. Установка платформы

После выполнения предварительных действий на экране отобразится окно приветствия программы установки.

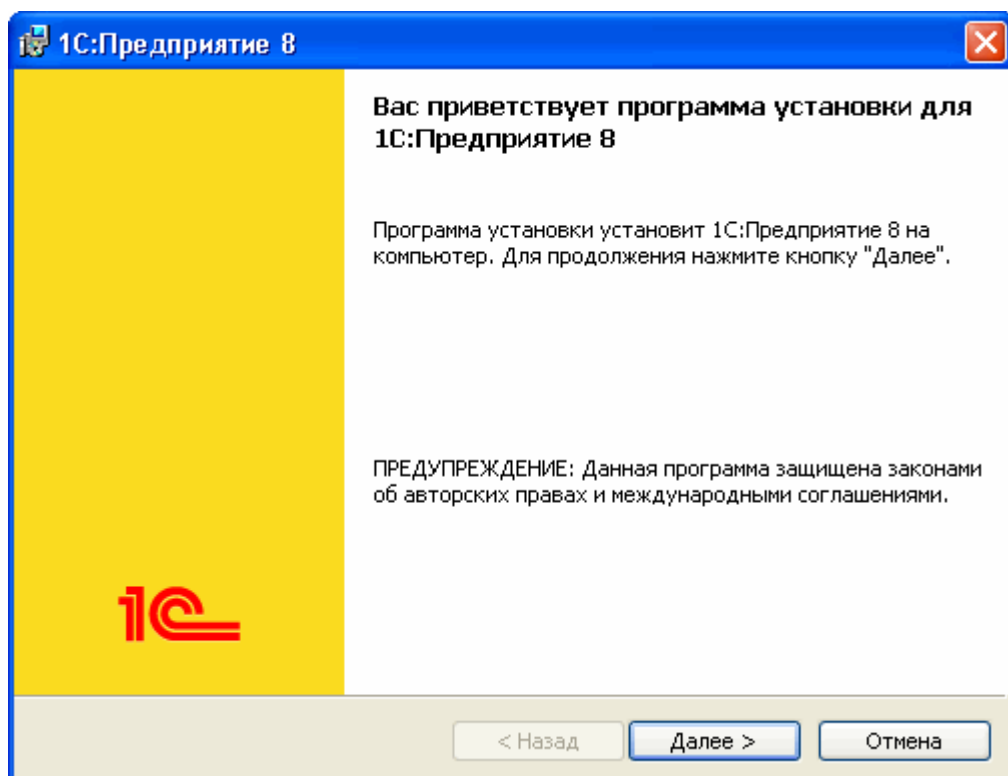


Рис. 3. Приветствие программы установки

Для продолжения установки следует нажать [Далее >](#).

В зависимости от того, была ли ранее система «1С:Предприятие» установлена, или это начальная установка, программа установки продолжает работу по-разному.

Программа установки предлагает выбрать компоненты для установки «1С:Предприятия».

В поле **Папка**: необходимо указать каталог установки, в который будут скопированы выбранные компоненты.

Можно использовать предложенное программой имя каталога установки или указать другое имя, нажав кнопку **Изменить**.

После выбора компонентов и указания имени каталога для продолжения установки следует нажать кнопку **Далее >**.

По умолчанию при первой установке программа предлагает установить компонент **1С:Предприятие**, который включает в себя полный набор функциональности платформы для работы на компьютере пользователя – толстый и тонкий клиенты, а также средства конфигурирования и администрирования.

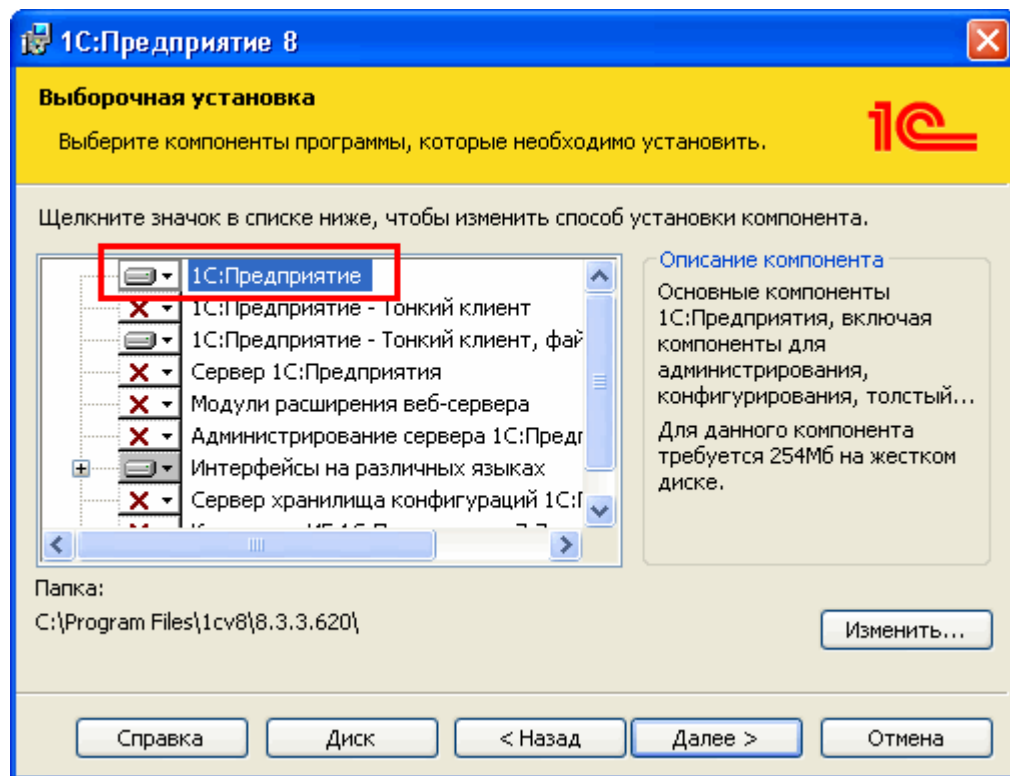


Рис. 4. Выбор установки основных компонентов «1С:Предприятия»

Для продолжения установки следует нажать кнопку **Далее >**, если не требуется особой настройки установки.

Чтобы установить только тонкий клиент для подключения к информационной базе, находящейся на сервере или на веб-сервере, выберите пункт **1С:Предприятие – Тонкий клиент**.

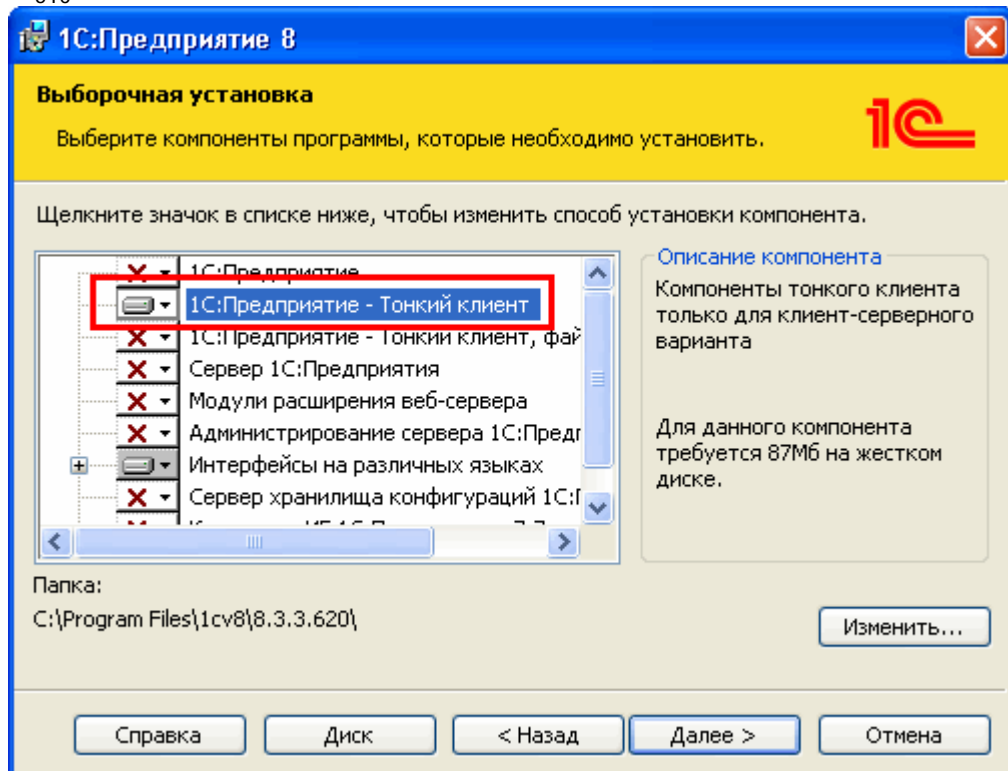


Рис. 5. Выбор установки компонентов тонкого клиента для клиент-серверного варианта

Для продолжения установки следует нажать кнопку [Далее >](#).

Если требуется установить только тонкий клиент для работы с информационной базой в файловом варианте, в списке компонентов следует выбрать [1С:Предприятие – Тонкий клиент, файловый вариант](#).

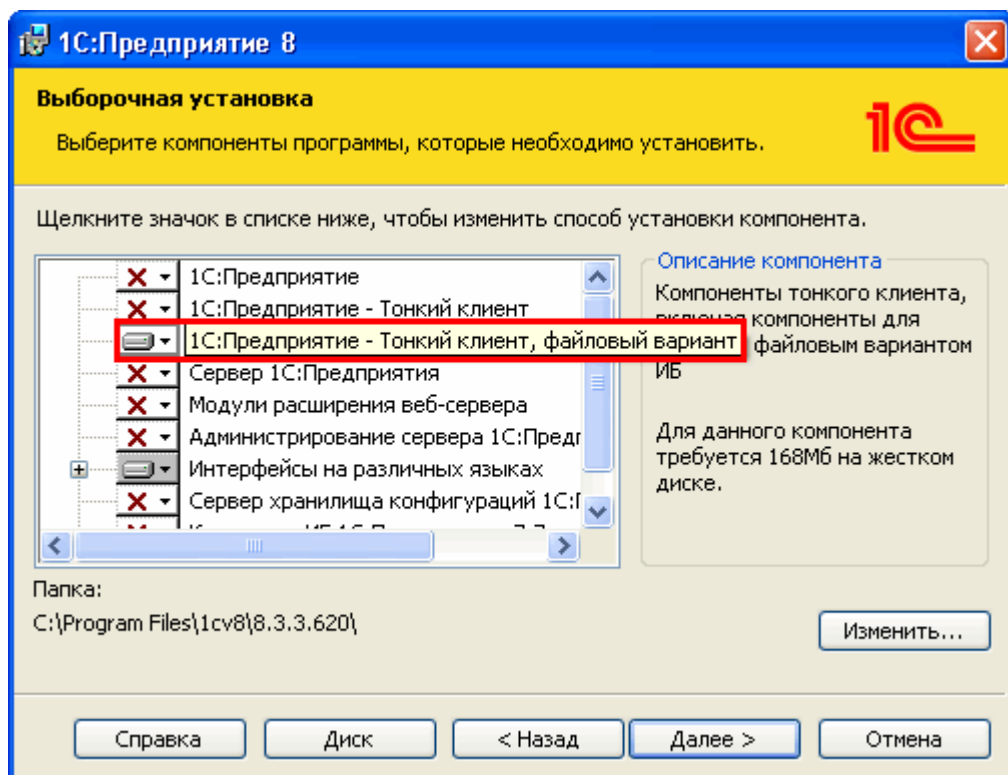


Рис. 6. Установка тонкого клиента для файлового варианта работы

2.1.2. Выбор языка интерфейса

На следующем шаге программа установки предложит выбрать язык интерфейса по умолчанию.

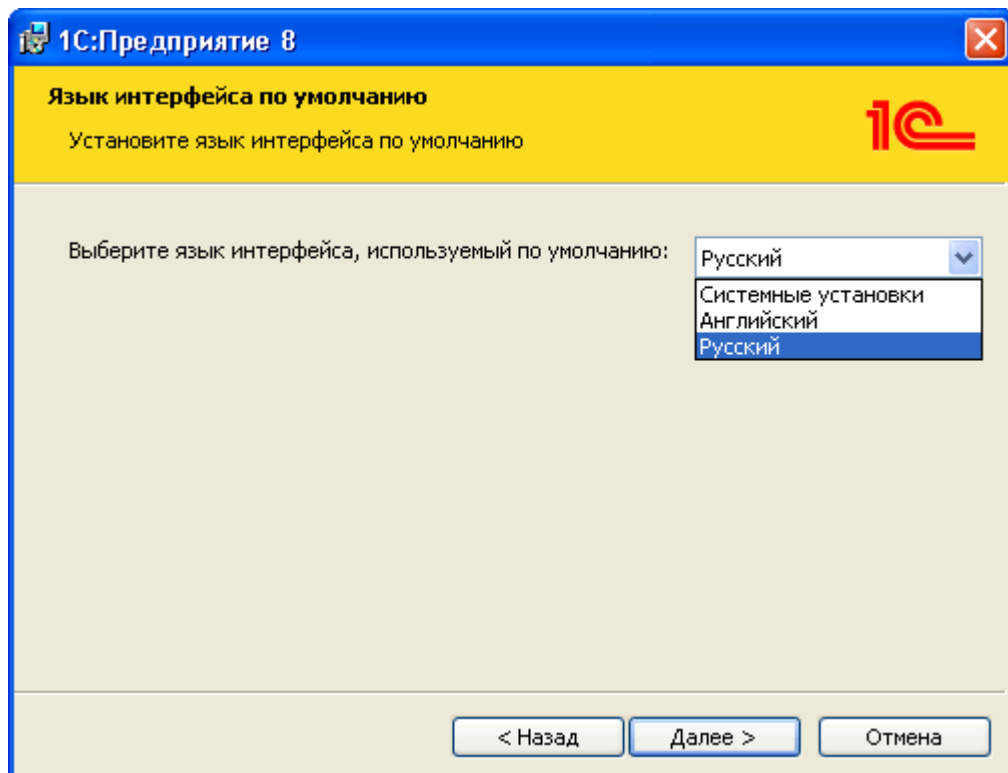


Рис. 7. Выбор языка интерфейса, используемого по умолчанию

Следует указать один из интерфейсов в качестве интерфейса по умолчанию.

Для продолжения установки следует нажать [Далее >](#).

На следующем шаге нажать кнопку [Установить](#).

Программа установки производит копирование файлов, создает программные группы и ярлыки, после чего предлагает произвести установку драйвера защиты от несанкционированного использования HASP Device Driver.

ВНИМАНИЕ! Установка драйвера требуется, если к USB-порту данного компьютера будет присоединен аппаратный ключ защиты.

После нажатия кнопки [Далее >](#) на экран выводится финальное состояние программы установки. Нажатие кнопки [Готово](#) завершает установку «1С:Предприятия».

2.2. Установка конфигурации

Для самостоятельной установки конфигурации необходимо выполнить следующие действия:

1. Запустить программу установки конфигурации (в ОС Windows – [setup.exe](#), в ОС Linux – [setup](#)).
2. В диалоге выбора каталога шаблонов указать каталог для установки конфигурации.
3. Нажать кнопку [Далее >](#).
4. Когда программа установки выполнит копирование файлов, нажать кнопку [Готово](#).

Выбранная конфигурация установлена.

2.3. Активация лицензии

В зависимости от конкретной поставки пользователю предоставляется лицензия в виде аппаратного ключа защиты программы или пинкода на получение электронной лицензии программы.

При активации программной лицензии важно учитывать, что она зависит от параметров компьютера, таких как наименование, версия, серийный номер и дата установки операционной системы, имя материнской платы, количество процессоров.

Изменение параметров компьютера может потребовать активации новой программной лицензии платформы с новым пинкодом.

Подробнее о защите от несанкционированного использования и опrogramмной лицензии «1С:Предприятия 8», а также полный список ключевых параметров см. в книге «1С:Предприятие 8.3. Руководство администратора», главу 9 «Защита от несанкционированного использования: особенности и настройка».

2.4. Особенности установки системы в ОС Linux

Использование шрифтов. Для корректной работы программы необходимо установить шрифты из состава Microsoft Core Fonts.

Существует несколько способов установки данных шрифтов:

- некоторые дистрибутивы включают в свой состав пакеты, осуществляющие скачивание и установку данных шрифтов. Например, [ttf-mscorefonts-installer](#) в Debian, Ubuntu и Mint;
- для RPM-дистрибутивов существует проект спец-файла, при помощи которого можно произвести установку данных шрифтов.

Подробная информация по установке: <http://corefonts.sourceforge.net/>;

- если первые два способа не подходят, следует вручную скачать шрифты (по ссылке <http://sourceforge.net/projects/corefonts/files/the%20fonts/final/>) и разместить их в каталоге `$HOME/.fonts` пользователя, под именем которого запускается «1С:Предприятие 8».

Внешние библиотеки. Для работы «1С:Предприятия 8» в Linux могут потребоваться следующие библиотеки:

Библиотека	Имя загружаемой библиотеки	Версия	Назначение
ImageMagick	libMagickWand , или libWand , или libMagickWand-6.Q16	6.2.8 и выше, если не планируется работать с географической схемой; 6.3.2 и выше, если планируется работать с географической схемой	Для отображения диаграмм, диаграмм Ганта, дендрограмм, сводных диаграмм
FreeType	libfreetype	2.1.9 и выше	Для отображения диаграмм, диаграмм Ганта, дендрограмм, сводных диаграмм; Для сохранения файлов в формате PDF
Libgsf	libgsf-1	1.10.1 и выше	Для экспорта и импорта файлов в формат XLS
Glib	libglib-2.0	2.12.4 и выше	Для работы с внешними источниками данных
UnixOdbc	libodbc	2.2.11 и выше	
Kerberos	libkrb5	1.4.2 и выше	Для аутентификации средствами ОС
GSS-API Kerberos	libgssapi_krb5	1.4.2 и выше	

Удаленное соединение. При использовании удаленного соединения необходимо использовать программу TightVNC.

Глава 3. Начало работы

Данная глава описывает процесс запуска клиентского приложения «1С:Предприятия».

3.1. Запуск приложения

В операционной системе Windows для запуска приложения «1С:Предприятия» на панели задач следует выбрать [Пуск – Все программы – 1С Предприятие 8 – 1С:Предприятие](#).

В операционной системе Linux для запуска приложения используется меню приложений Linux: категория – [Office:Finance](#), ярлык – [1С:Предприятие](#).

В появившемся окне следует выбрать информационную базу и нажать кнопку [1С:Предприятие](#).

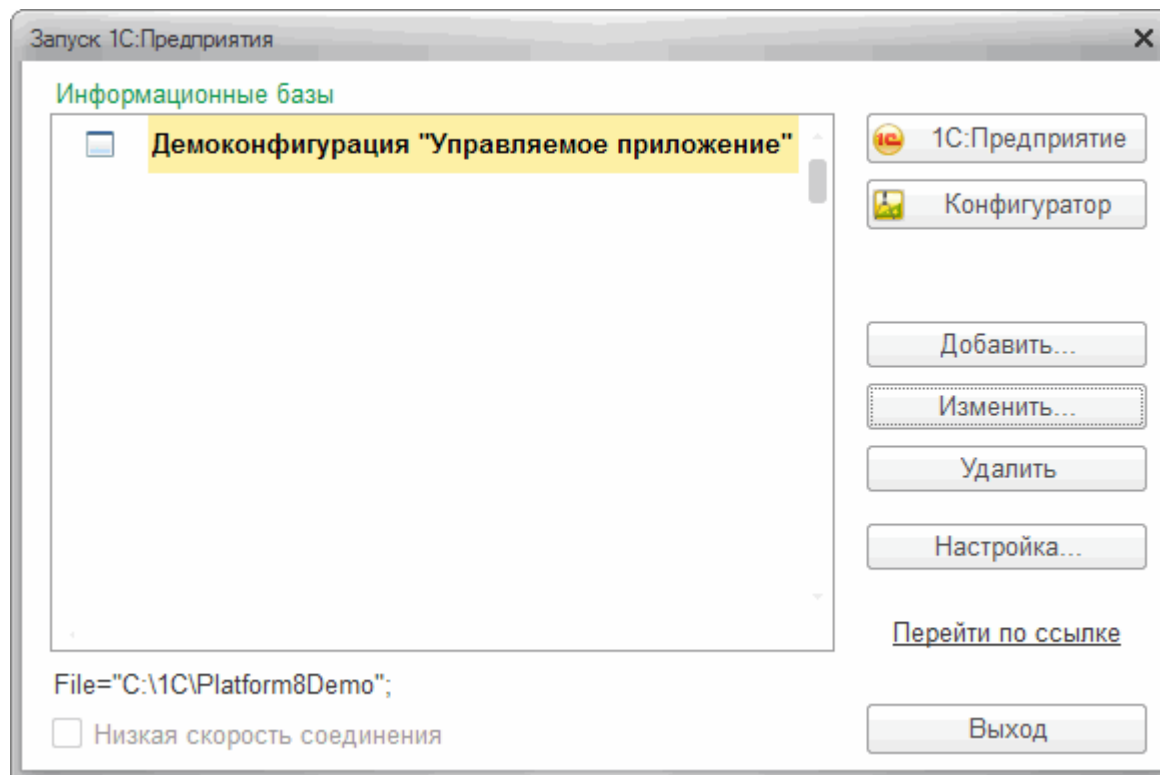


Рис. 8. Диалог запуска «1С:Предприятия»

Чтобы добавить информационную базу в список, следует нажать кнопку [Добавить](#). Подробнее о добавлении и создании информационных баз см. раздел «[Добавление информационной базы](#)» [здесь](#).

Для изменения параметров информационной базы следует нажать кнопку [Изменить](#).

Чтобы перейти по навигационной ссылке следует нажать [Перейти по ссылке](#), в открывшемся диалоге ввести адрес ссылки и нажать кнопку [Перейти](#) или клавишу [Enter](#).

3.2. Вход в систему

При запуске информационной базы система проверяет наличие прав доступа пользователя к информационной базе. Другими словами, выполняется [аутентификация](#).

Если в информационной базе не задан список пользователей, будет выполнен вход в систему.

Если в информационной базе существует список пользователей, то в зависимости от конфигурации системы параметры аутентификации пользователя могут быть связаны с настройками операционной системы.

Если такой настройки установлено не было, система предложит ввести имя пользователя и пароль.

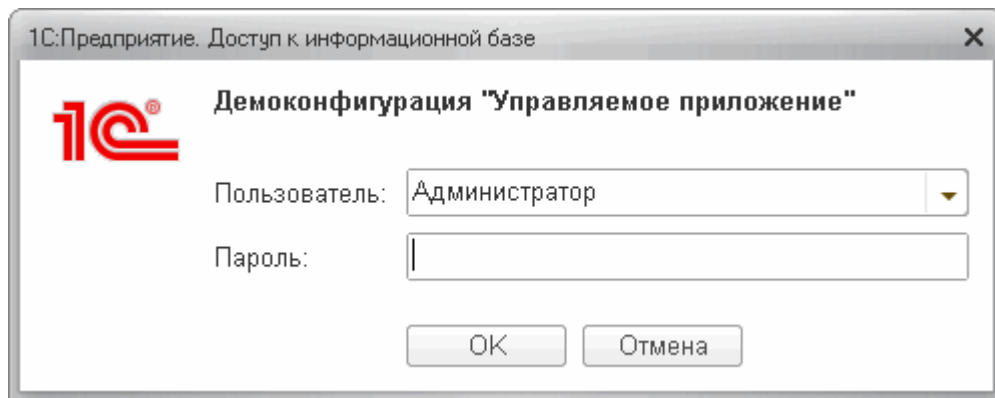


Рис. 9. Вход в систему

В этом диалоге необходимо указать имя пользователя. Для этого следует в поле **Пользователь** выбрать пользователя из списка или набрать имя с клавиатуры и ввести пароль (если он был установлен).

При нажатии кнопки **ОК** будет выполнен вход в систему.

Нажатие клавиши **Отмена** позволяет отказаться от запуска программы.

3.2.1. Запуск веб-клиента

Для начала работы с системой «1С:Предприятие» в режиме веб-клиента введите URL-адрес базы в адресную строку браузера. При этом браузер должен быть настроен соответствующим образом. О настройке браузера см. разделы «Настройка Microsoft Internet Explorer» на стр. 251, «Настройка Mozilla Firefox» [здесь](#), «Настройка Google Chrome» [здесь](#), и «Настройка Safari» [здесь](#).

3.2.2. Установка скорости соединения

Если в параметрах информационной базы установлена возможность выбирать режим скорости соединения, то в диалоге запуска будет доступен параметр **Низкая скорость соединения**:

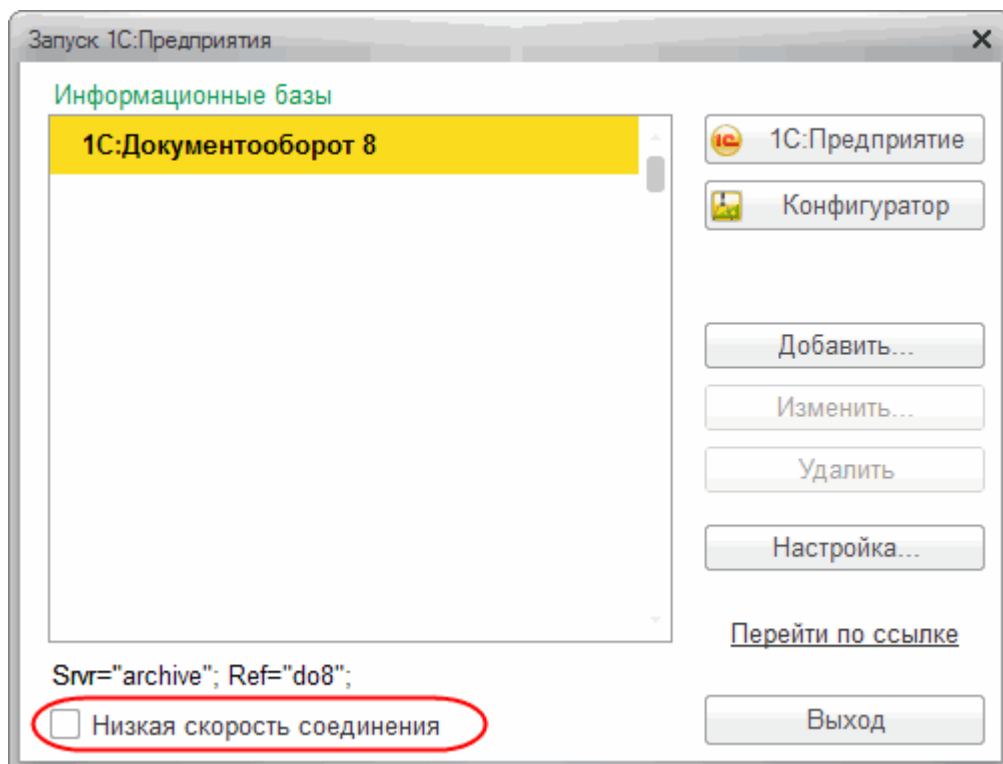


Рис. 10. Параметр информационной базы

При установленном флажке **Низкая скорость соединения** работа системы «1С:Предприятие» будет сопровождаться некоторыми особенностями, которые перечислены в главе [«Режим низкой скорости соединения»](#). Флажок рекомендуется устанавливать при подключении к информационной базе через

3.3. Добавление информационной базы

Чтобы добавить новую информационную базу в список, в диалоге списка информационных баз следует нажать кнопку **Добавить**. На экране отобразится диалог выбора режима добавления.

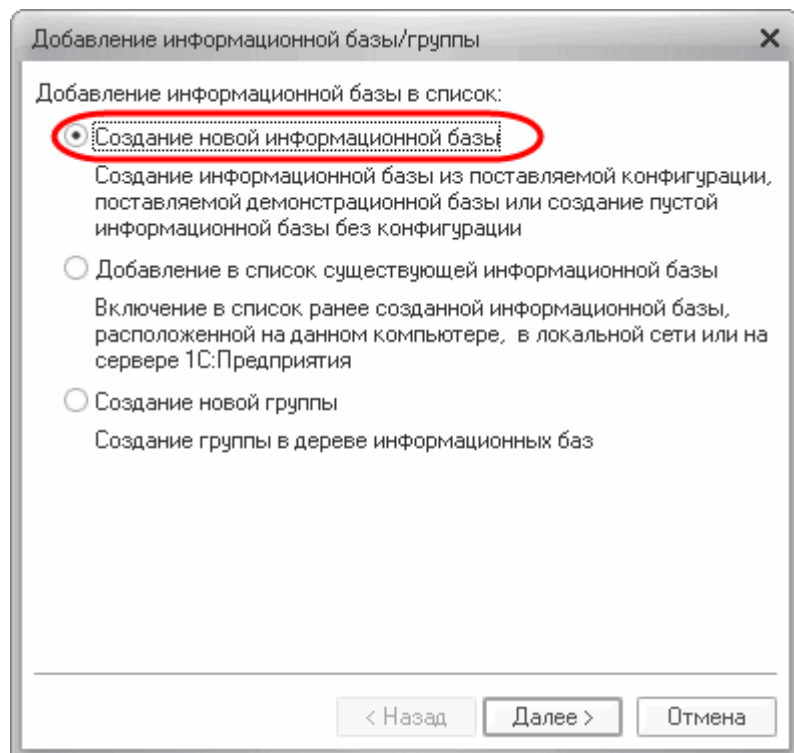


Рис. 11. Выбор режима добавления информационной базы

3.3.1. Создание информационной базы

Если выбран режим **Создание информационной базы**, то система «1С:Предприятие» предоставляет возможность создать информационную базу на основании базы-шаблона или пустую базу.

После нажатия кнопки **Далее >** на экране отобразится диалог выбора.

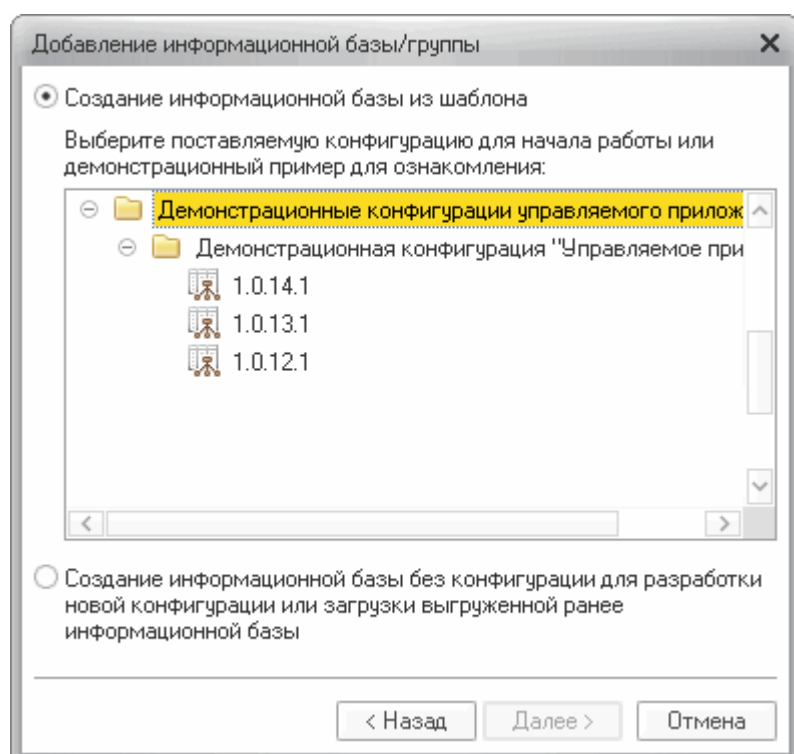
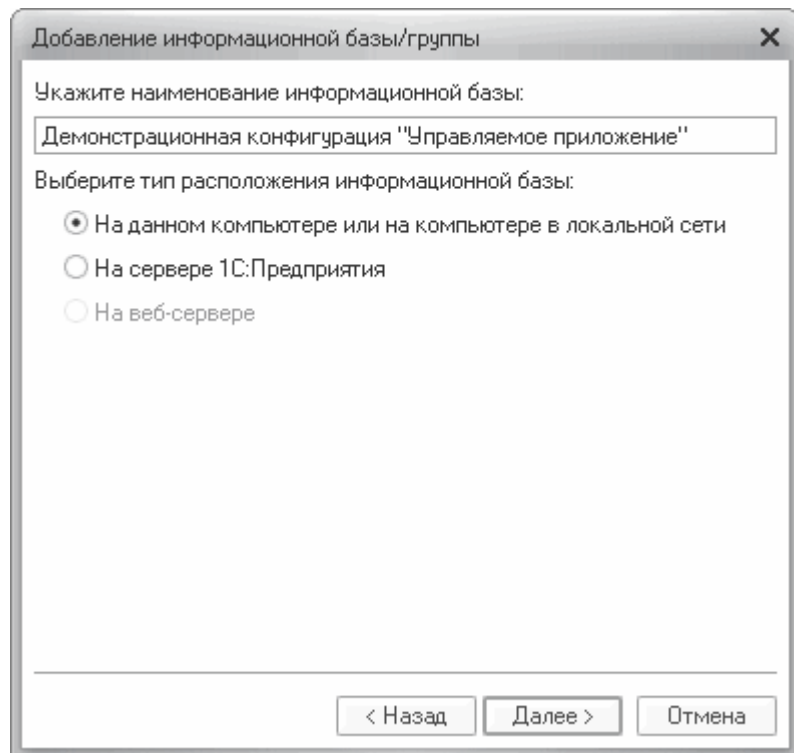


Рис. 12. Диалог выбора способа создания базы

В списке шаблонов выберите исходный шаблон и следует нажать [Далее >](#). На экране отобразится диалог ввода наименования и выбора типа расположения информационной базы.

**Рис. 13. Диалог ввода наименования информационной базы**

Наименование информационной базы представляет собой произвольную строку символов.

Информационная база может быть расположена на локальном компьютере, в сети, на сервере «1С:Предприятия» или на веб-сервере.

Для продолжения создания информационной базы следует нажать [Далее >](#).

В зависимости от выбранного типа расположения установите необходимые параметры информационной базы и следует нажать [Далее >](#).

На следующем этапе выберите параметры запуска информационной базы:

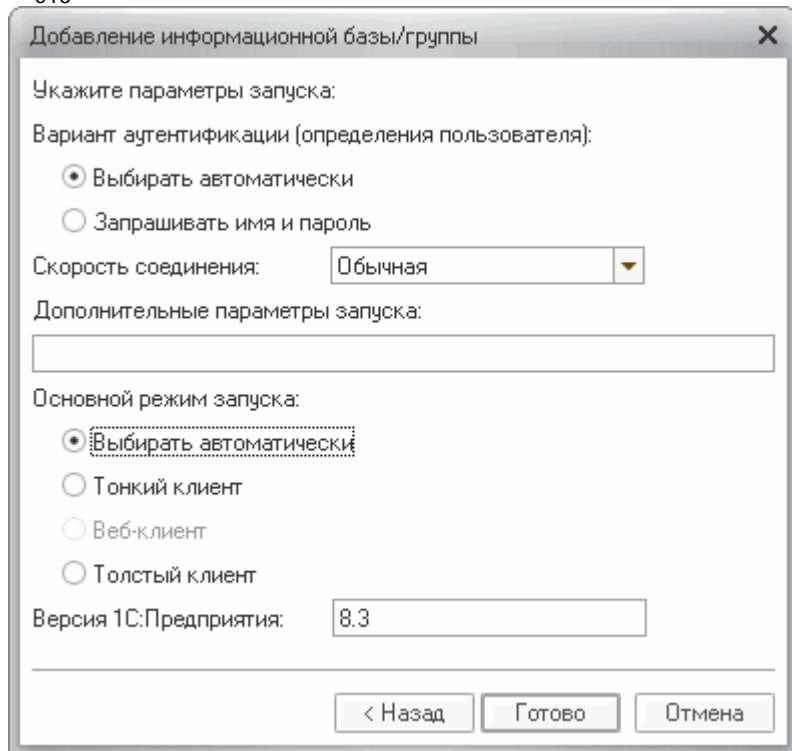


Рис. 14. Диалог редактирования параметров запуска

Параметр **Вариант аутентификации** может принимать следующие значения:

- **Выбирать автоматически** – в этом случае вначале будет предпринята попытка выполнить аутентификацию средствами ОС, а в случае неудачи – предложено ввести логин/пароль для доступа к информационной базе.
- **Запрашивать имя и пароль** – в этом случае аутентификация будет всегда выполняться в диалоге ввода имени пользователя и пароля.

Параметр **Скорость соединения** позволяет определить скорость соединения с информационной базой или сервером «1С:Предприятия». Параметр может принимать следующие значения:

- **Обычная** – обычная скорость. При работе системы не будет никаких особенностей.
- **Низкая** – низкая скорость соединения. В этом режиме работа «1С:Предприятия» будет сопровождаться некоторыми особенностями, которые описаны в главе «[Режим низкой скорости соединения](#)». Данный режим рекомендуется использовать при работе по медленным каналам связи, например, при подключении через GPRS-модем.
- **Выбирать при запуске** – в этом режиме выбирать скорость соединения можно будет при каждом запуске информационной базы с помощью флажка **Низкая скорость соединения** в нижней части окна **Запуск 1С:Предприятия**. Если в свойствах информационной базы указано конкретное значение (**Обычная** или **Низкая**), то флажок **Низкая скорость соединения** в окне **Запуск 1С:Предприятия** недоступен для изменения и отражает значение, выбранное в свойствах информационной базы.

Поле **Дополнительные параметры запуска** позволяет указать различные параметры командной строки, которые будут переданы исполняемому файлу. Подробнее про параметры командной строки можно прочитать во встроенной справке (раздел [Запуск "1С:Предприятия 8" и параметры запуска](#)).

Параметр **Основной режим запуска** определяет, какой клиент будет использоваться для доступа к информационной базе:

- **Выбирать автоматически** – в этом режиме вид клиентского приложения будет определяться автоматически;
- **Тонкий клиент** – для запуска будет использоваться тонкий клиент;
- **Веб-клиент** – для запуска будет использоваться веб-клиент;

- **Толстый клиент** – для запуска будет использоваться толстый клиент.

Поле **Версия 1С:Предприятия**: позволяет указать конкретный номер версии, который необходимо использовать для доступа к данной информационной базе.

Работа со списком информационных баз подробно описана в книге «1С:Предприятие 8.3. Руководство администратора» в главе, посвященной ведению списка информационных баз.

Глава 4. Интерфейс системы

В данной главе содержатся сведения об интерфейсе системы и описываются приемы навигации по программе.

В «1С:Предприятие» пользователь работает с системой окон. Существует два вида окон: **основное** и **вспомогательные**.

Основное окно приложения предназначено для навигации по приложению и вызова различных команд, а вспомогательное окно – для работы с конкретными объектами информационной базы (например, с документами или элементами списков), построения отчетов или выполнения обработки данных. Чтобы открыть ту или иную форму, пользователю следует найти в интерфейсе нужную команду (открытие списка, ввода документа, и т. д.) и выполнить ее (выбрать с помощью мыши или клавиши **Enter**).

Существует три вида (режима) интерфейса:

- [Формы в закладках](#),
- [Формы в отдельных окнах](#),
- [Такси](#).

Вид интерфейса может быть установлен разработчиком или же выбран пользователем в системных параметрах, если его использование предусмотрено разработчиком. Подробнее о переключении вариантов интерфейса см. раздел [«Системные параметры»](#), [здесь](#).

Описание интерфейса [Такси](#) и приемы работы с ним приведены в книге «1С:Предприятие 8.3. Руководство пользователя. Интерфейс «Такси».

4.1. Формы в закладках

В этом режиме интерфейса вся работа происходит преимущественно в одном окне, что может оказаться удобным, например, при одновременной работе пользователя с несколькими информационными базами или большим количеством используемых форм.

4.1.1. Основное окно

При запуске системы открывается основное окно программы. Окно предназначено для навигации по программе и вызова различных команд. В нем пользователю представляется вся структура прикладного решения. Основное разделение функциональности представляется в верхней части окна в виде панели разделов.

Окно устроено таким образом, что позволяет достаточно быстро находить необходимые области и вызывать требуемые команды. Для изменения размеров основного окна можно использовать небольшую область в правом нижнем углу (с тремя утопленными точками).

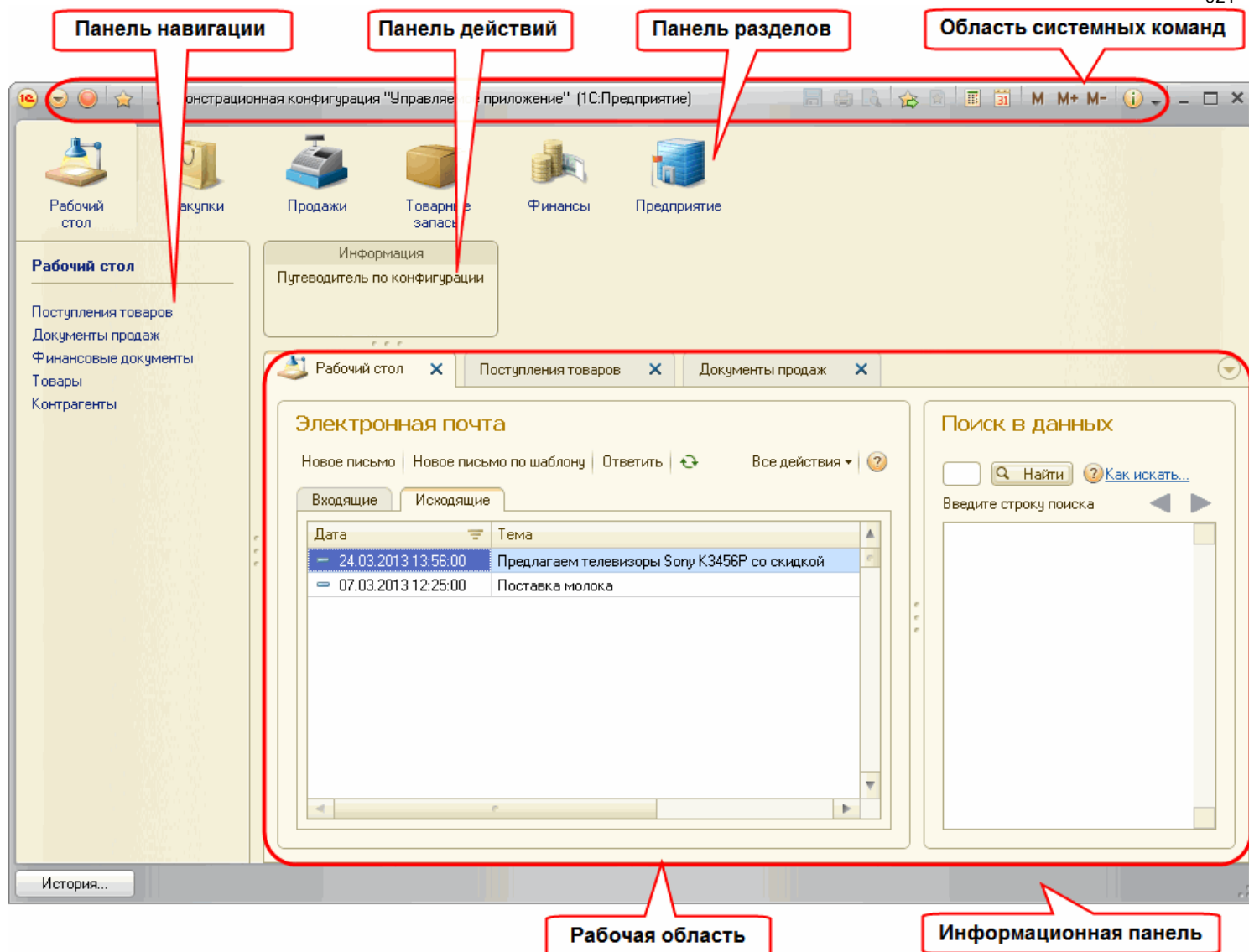


Рис. 15. Основное окно в интерфейсе с закладками

Формы отображаются на закладках рабочей области основного окна.

Закладки создаются для каждой открываемой формы. На панели может быть открыто неограниченное количество закладок.

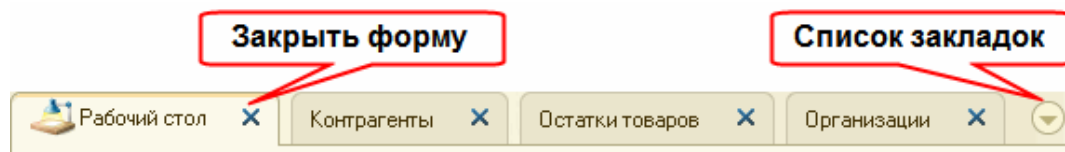


Рис. 16. Закладки на рабочей области

Переключение между закладками осуществляется с помощью мыши или сочетания клавиш **Ctrl + Tab** (**Ctrl + Shift + Tab**) и **Ctrl + F6** (**Ctrl + Shift + F6**).

Для закрытия формы используется клавиша **Esc**, сочетание клавиш **Ctrl + F4** или кнопка **Закреть форму** на закладке.

Переключение раздела не закрывает текущую открытую форму.

Закладка «Рабочий стол». Если на рабочем столе есть формы, то первой отображается закладка **Рабочий стол**. Закладка автоматически отображается при начале работы и не может быть закрыта нажатием клавиши **Esc**.

Список закладок. Чтобы просмотреть список всех открытых окон, следует нажать кнопку, расположенную у правого края панели окон. Первой в списке будет отображена форма рабочего стола, ниже в алфавитном порядке остальные открытые формы. При выборе формы из списка активизируется форма на соответствующей закладке.

Формы, блокирующие владельца, блокируют только окно, на котором расположена форма-владелец. При этом можно переключаться по закладкам других форм, производить выбор раздела, выполнять команды из панелей навигации и/или действий.

Открытие формы в отдельном окне. Для открытия расположенной на закладке формы в отдельном окне, используется команда [Открыть в отдельном окне](#) контекстного меню закладки. Данное действие доступно только для форм, имеющих навигационную ссылку. Так, например, форму создания элемента в новом окне на закладке открыть нельзя.

Окно сообщений. Формы, открытые в рабочей области основного окна, используют панель сообщений основного окна. При переключении закладок окно сообщений отображает сообщения, относящиеся к выбранной форме. Подробнее о работе с сообщениями см. раздел [«Сообщения»](#), [здесь](#).

Меню «Окна». В меню [Окна](#) (вызываемом при выборе [Главное меню – Окна](#)) наравне с отдельными окнами и основным окном отображаются названия документов, открытые в закладках. Текущее вспомогательное окно отмечается флажком, а для основного окна флажком помечается активная закладка, при отсутствии открытых закладок флажком помечается основное окно.

4.1.2. Рабочий стол

При запуске системы первым всегда отображается раздел, который называется **Рабочий стол**. Для перехода к рабочему столу из других разделов служит пункт [Рабочий стол](#) панели разделов. Для переключения между формами рабочего стола используется клавиша **F6**.

Например, рабочий стол может выглядеть следующим образом:

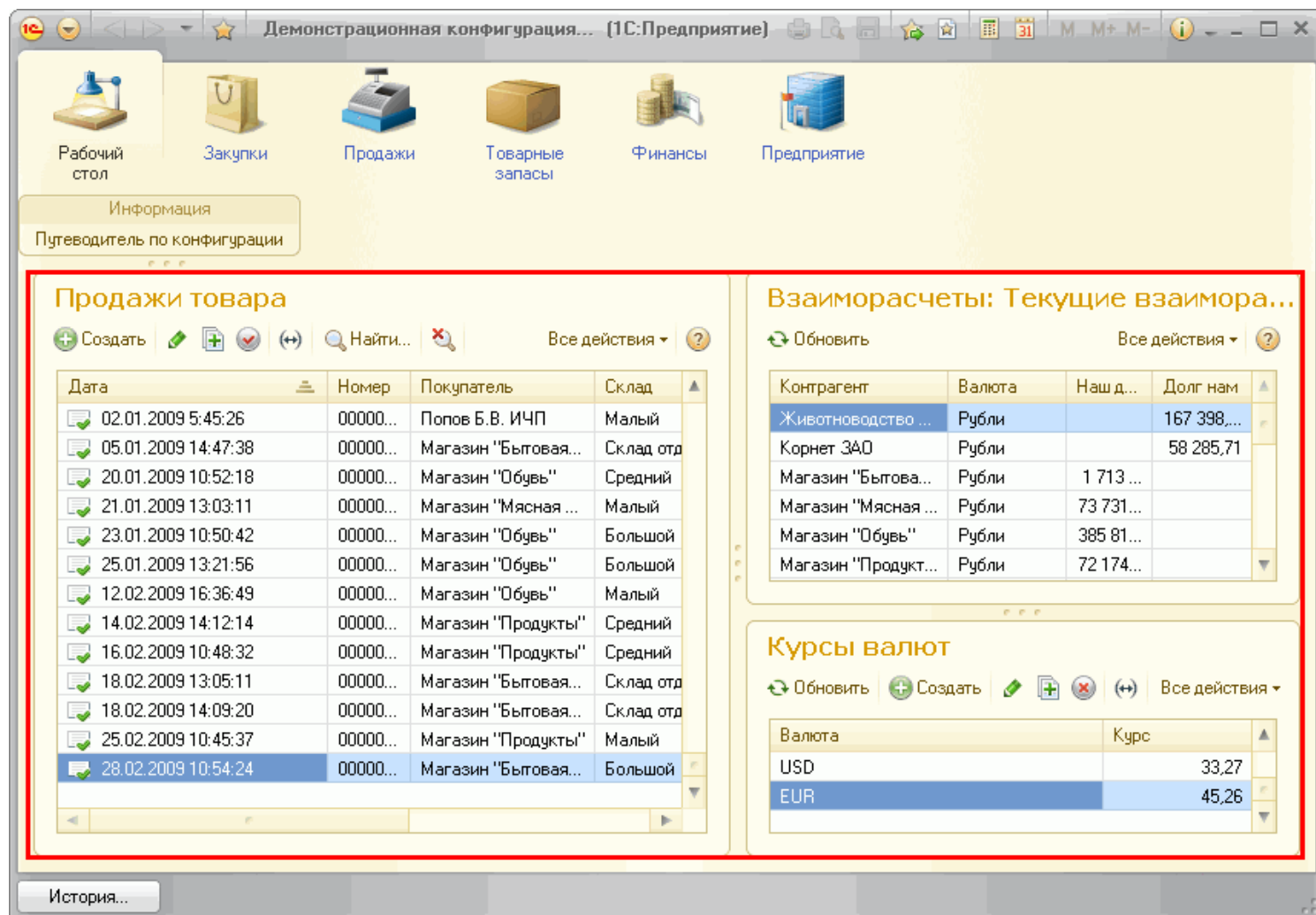


Рис. 17. Рабочий стол приложения

Существует возможность настраивать расположение форм на рабочем столе. Диалог настройки рабочего стола открывается соответствующим пунктом контекстного меню, вызываемого на любой из панелей. Подробнее о настройке рабочего стола см. раздел [«Рабочий стол»](#) [здесь](#).

4.1.3. Панель разделов

Панель разделов содержит список доступных пользователю разделов, из которых состоит система.

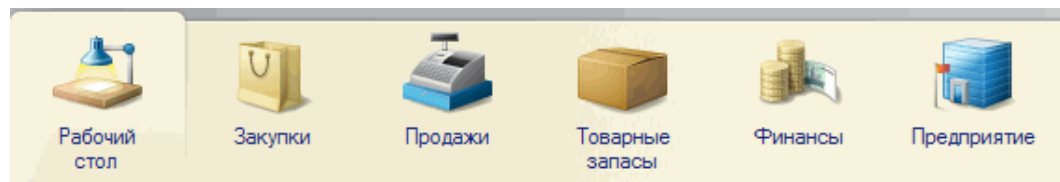


Рис. 18. Пример панели разделов

Чтобы перейти в требуемый раздел, следует нажать ссылку в названии раздела или его изображение. При переходе в раздел отображаемая в рабочей области информация не меняется, а меняется содержимое панели навигации и панели действий.

Если разделы не умещаются в списке по ширине окна, то по краям списка появляются кнопки прокрутки:

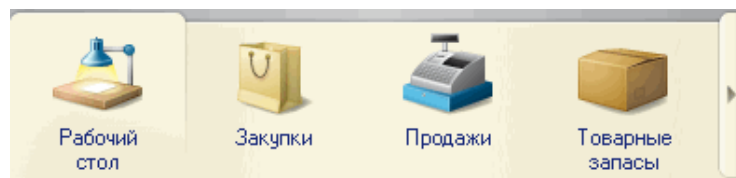


Рис. 19. Пример кнопки прокрутки на панели разделов

СОВЕТ. При работе с клавиатуры для перехода на панель разделов следует нажать **Alt + 1**.

Существует возможность настраивать панель разделов. Диалог настройки панели разделов открывается соответствующим пунктом контекстного меню, вызываемого на любой из панелей окна. Настройка подробно описана в разделе «[Панель разделов](#)» [здесь](#).

4.1.4. Панель навигации

Панель навигации отображает структуру текущего раздела. Панель представляет собой список гиперссылок. Если у раздела имеются подчиненные разделы, то они будут отображаться как сворачиваемые группы.

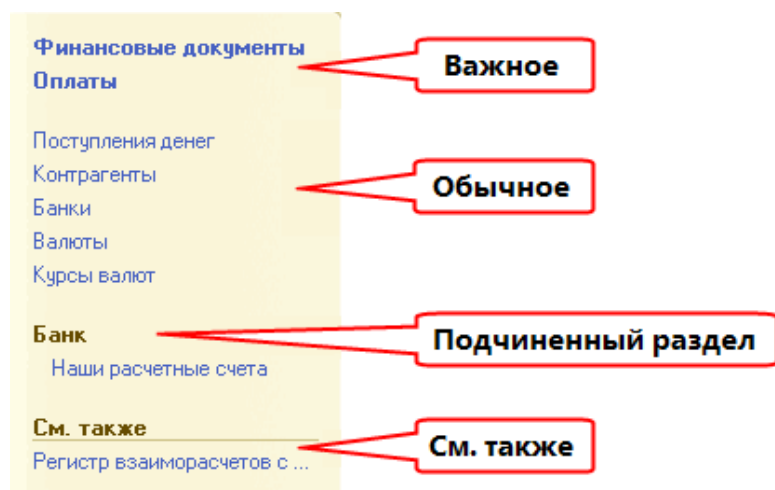


Рис. 20. Группы команд на панели навигации

Ссылки на панели навигации подразделяются на три группы:

- **Важное** – ссылки на наиболее важные области в контексте текущего раздела;
- **Обычное** – ссылки на данные текущего раздела;
- **См. также** – ссылки на дополнительную информацию, которая может не относиться к данным текущего раздела напрямую, но может быть востребована в некоторых случаях.

Состав списка ссылок и принадлежность конкретной ссылки к группе устанавливаются на этапе конфигурирования.

При нажатии ссылок, как правило, открываются новые формы. При этом формы открываются непосредственно в рабочей области основного окна в виде закладок

СОВЕТ. При работе с клавиатуры для перехода на панель навигации текущего раздела следует нажать **Alt + 2**.

Существует возможность пользовательской настройки панели навигации. Диалог настройки панели открывается соответствующим пунктом контекстного меню, вызываемого на любой из панелей. О настройке панели навигации см. раздел «[Панель навигации](#)» [здесь](#).

Панель навигации может не отображать ссылки, если у пользователя недостаточно прав на их использование или они были отключены пользователем.

4.1.5. Панель действий

На панели действий отображаются списки команд, которые доступны в текущем разделе. Эти команды объединены в группы: [Создать](#), [Отчеты](#), [Сервис](#). Также могут существовать иные группы, созданные в конфигурации.

В группе [Создать](#) размещаются команды создания новых объектов информационной базы, например, документов или элементов списков.

Группа [Отчеты](#) содержит команды открытия форм отчетов.

В группе [Сервис](#) отображаются команды вызова служебных инструментов. Если в группе нет ни одной команды, то она не отображается. Высота и ширина панели регулируются.

При наведении курсора на название команды отображается подсказка, которая может содержать дополнительную информацию о действии или ссылку на нее.

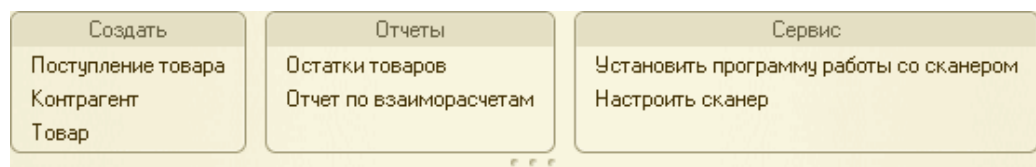


Рис. 21. Группы команд на панели действий

Чтобы изменить размеры панели действий, наведите курсор мыши на три утопленные точки, следует нажать левую кнопку мыши и потяните в нужную сторону. При этом если часть команд не уместилась, то эти команды отображаются в меню, открываемом при нажатии кнопки раскрытия. Например, как показано на рисунке ниже.

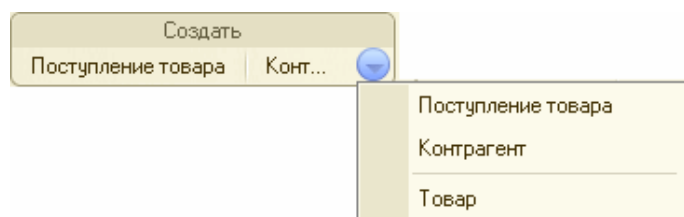


Рис. 22. Список команд в группе «Создать»

Высота панели действий сохраняется между сеансами работы для каждого раздела.

СОВЕТ. При работе с клавиатуры для перехода на панель действий текущего раздела следует нажать **Alt + 3**.

Состав и порядок команд в группах на панели действий можно изменять в форме настройки (подробнее см. раздел «[Панель действий](#)» [здесь](#)). Диалог настройки панели открывается соответствующим пунктом контекстного меню, вызываемого на любой из панелей окна.

Панель действий может не отображать команды, если у пользователя недостаточно прав на их использование или они были отключены пользователем.

4.1.6. Информационная панель

Панель предназначена для открытия окна истории, для обращения к последним данным, которые редактировал пользователь, и для отображения перечня последних оповещений (информации о действиях, выполненных в

программе).



Рис. 23. Информационная панель

При нажатии кнопки [История...](#) откроется список последних измененных объектов. Подробнее о работе с историей см. раздел «[История](#)» [здесь](#).

При нажатии гиперссылки оповещения на новой закладке откроется форма объекта, на который указывает ссылка. Например, по нажатии гиперссылки [Корнет ЗАО](#) откроется форма элемента списка [Контрагенты – Корнет ЗАО](#). Подробнее об оповещениях см. раздел «[Оповещения](#)» [здесь](#).

4.1.7. Область системных команд

Заголовок приложения является «составным» и содержит область системных команд.

В левой части **заголовка** расположено главное меню, кнопка вызова меню функции (см. раздел «[Меню функций](#)» [здесь](#)), меню работы с избранным ([Избранное](#)).

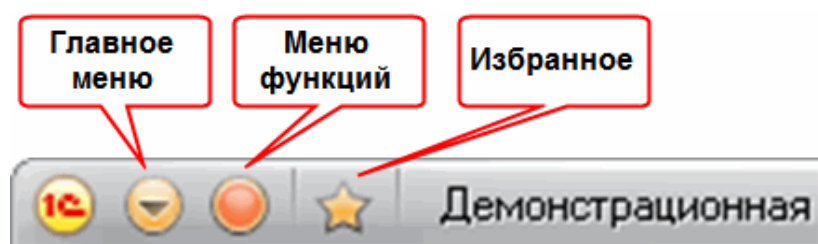


Рис. 24. Командная панель основного окна в режиме «Формы в закладках»

Главное меню содержит только общие команды – это команды для работы с файлами, команды управления окнами, сервисные команды и др.

СОВЕТ. При работе с клавиатуры для перехода в главное меню текущего окна следует нажать клавишу **F10**.

В центре располагается заголовок приложения.



Рис. 25. Правая часть области системных команд

В правой части области системных команд по умолчанию расположены команды работы с файлами, работы со ссылками и стандартные функции, такие как вызов календаря, калькулятора и т. д.

Закрытие основного окна и выход из программы осуществляются при вызове команды [Файл – Выход](#) главного меню или при нажатии кнопки [Закреть](#) основного окна. Поддерживается возможность навигации по программе при помощи клавиатуры. Таблицы сочетаний клавиш для навигации содержатся во встроенной справке.

4.1.8. Меню функций

Меню функций предоставляет удобный доступ к командам любого раздела и содержит все команды текущего раздела, сгруппированные аналогично составу панелей навигации и действий.

Вызов меню функций выполняется нажатием кнопки, расположенной в области системных команд, или с помощью сочетания клавиш **Alt + `**.

Если панель разделов отсутствует, то при вызове меню функций панель разделов отображается автоматически. Также выключенную панель разделов, панель навигации или панель действий можно вызвать с помощью горячих клавиш **Alt + 1**, **Alt + 2** или **Alt + 3** соответственно.

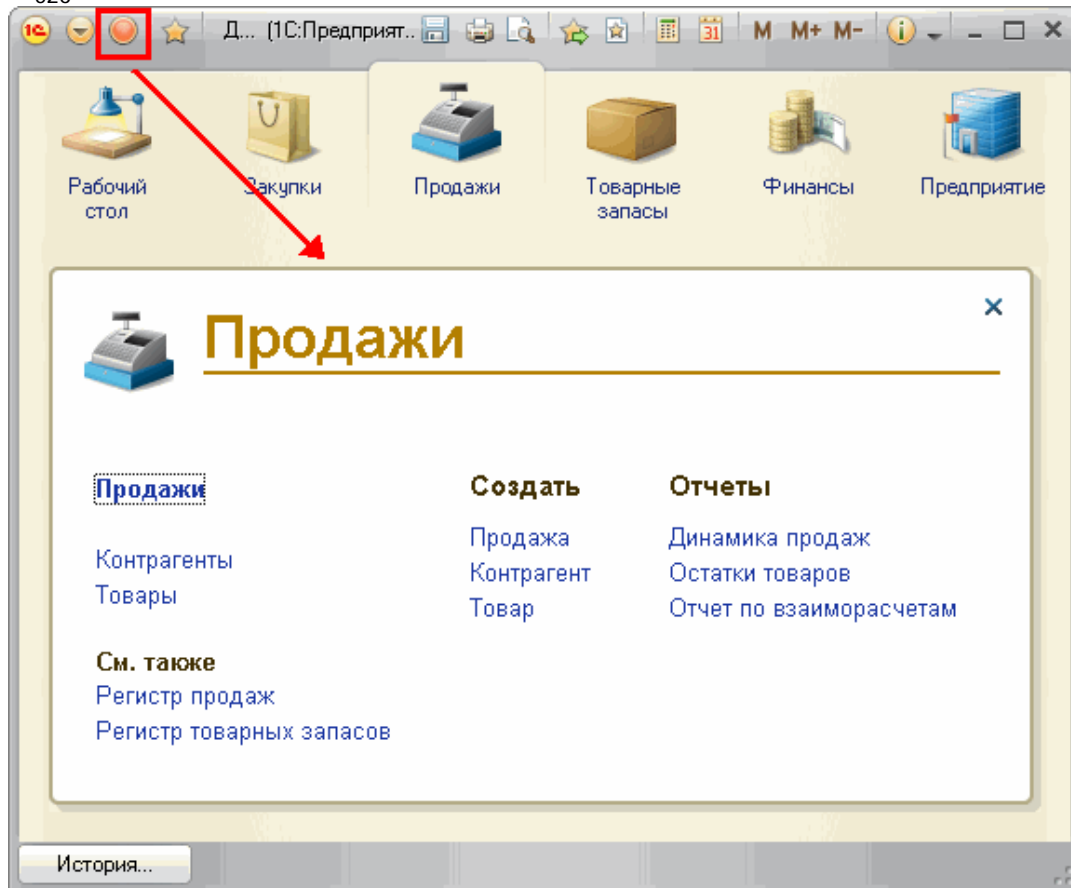


Рис. 26. Открытое меню функций

При работе с помощью клавиатуры навигация по командам осуществляется клавишами перемещения курсора, а также клавишами [Home](#), [End](#) и [Tab](#).

Команда выполняется по щелчку мыши, при нажатии клавиши [Enter](#), пробел или при помощи команды [Открыть в новой закладке](#) контекстного меню. При выборе команды меню закрывается автоматически.

Если меню функций открыто и при этом происходит переключение раздела, то состав команд обновляется в соответствии с выбранным разделом. При открытии меню функций предыдущая выполненная команда подсвечивается.

Чтобы закрыть меню функций, следует нажать клавишу [Esc](#) или крестик в правом верхнем углу [Меню функций](#).

4.1.9. Вспомогательное окно

Вспомогательные окна предназначены для работы с объектами информационной базы, построения отчетов или выполнения обработки данных. Они отображаются на экране независимо от основного окна. Чтобы открыть форму в отдельном окне, следует в контекстном меню формы выбрать команду [Открыть в отдельном окне](#).

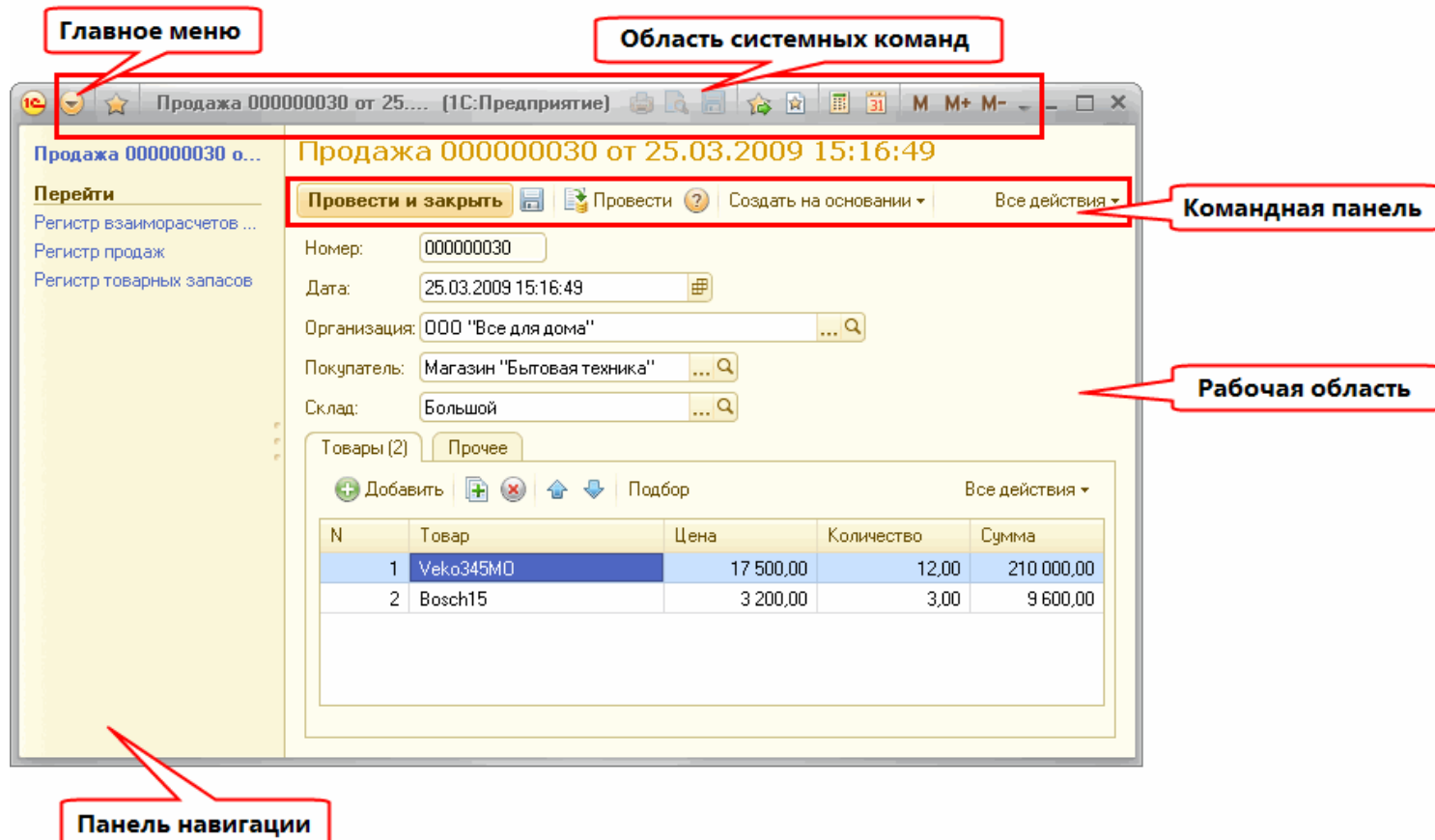


Рис. 27. Вспомогательное окно. Документ «Продажа»

Закрытие вспомогательного окна происходит по команде **Файл – Закрыть** и не приводит к закрытию всего приложения.

Система позволяет отображать неограниченное количество вспомогательных окон. Одновременно для каждого объекта (например, документа или элемента списка) может быть открыто только одно вспомогательное окно, если иное не предусмотрено в конфигурации.

СОВЕТ. На основное или вспомогательное окно приложения можно перетаскивать файлы из операционной системы – например, из окна **Проводник**. Операция приведет к открытию файлов аналогично действию команды **Файл – Открыть**.

4.1.9.1. Навигация во вспомогательном окне

Панель навигации вспомогательного окна позволяет переходить к просмотру различных сведений, логически связанных с данными, которые отображает основная форма окна.

В панели навигации могут располагаться команды перехода к самой форме, а также группы ссылок **Перейти** и **См. также**. Если какая-либо группа не была установлена на этапе конфигурирования, она не отображается. В зависимости от конфигурации на панели навигации могут отображаться и другие команды.

Если переход по ссылке осуществляется для еще не сохраненного объекта, система выводит предупреждение: **Данные еще не записаны. Переход к "XXXX" возможен только после записи данных. Данные будут записаны.**

При нажатии кнопки **OK** текущие данные будут сохранены и будет выполнен переход к выбранной области данных. При нажатии кнопки **Отмена** сохранение и переход не выполняются, и можно продолжить редактирование текущих данных в обычном режиме.

Работа с панелью навигации во вспомогательном окне не отличается от работы с панелью навигации в основном окне, за исключением того, что переход к самой форме в отдельном окне невозможен.

Для возврата к основной форме вспомогательного окна следует нажать гиперссылку в верхней части панели навигации. Заметим, что открытие нового окна при переходе по этой ссылке не допускается.

4.1.9.2. Командная панель формы

Командная панель формы содержит команды, непосредственно связанные с объектом, который отображается в основной форме. Команды отображаются на панели в виде кнопок.

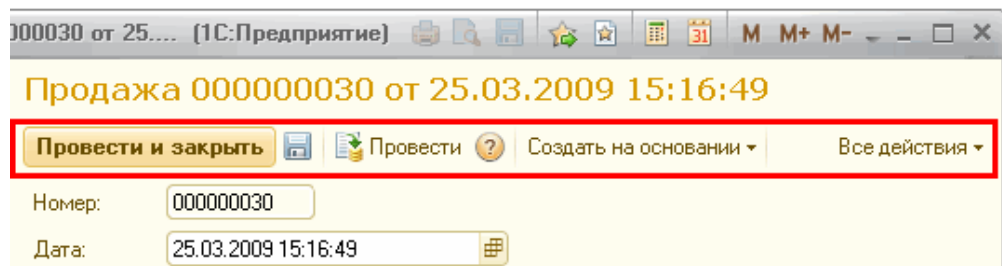


Рис. 28. Командная панель формы во вспомогательном окне

Кнопка, используемая по умолчанию (и при нажатии **Enter**), выделяется на панели полужирным начертанием текста и специальной формой. На [рис. 28](#) кнопкой по умолчанию является кнопка **Провести и закрыть**.

Меню **Все действия** содержит список всех доступных команд формы.

4.2. Формы в отдельных окнах

В этом режиме работы интерфейса каждое окно «1С:Предприятия» появляется на панели задач и в переключателе окон по нажатии клавиш **Alt + Tab**.

При работе в операционной системе Windows 7 все окна, кроме окна списка баз, группируются на панели задач в одну группу.

СОВЕТ. Между окнами одного сеанса можно переключаться с помощью клавиш **Ctrl + Tab**.

Описание работы в основном окне приводится ниже. Описание работы во вспомогательном окне см. [здесь](#).

4.2.1. Основное окно

При запуске системы открывается основное окно программы. Окно предназначено для навигации по программе и вызова различных команд. В нем пользователю видна вся структура прикладного решения. Основное разделение функциональности представляется в верхней части окна в виде панели разделов.

В общем случае основное окно приложения имеет следующий вид:

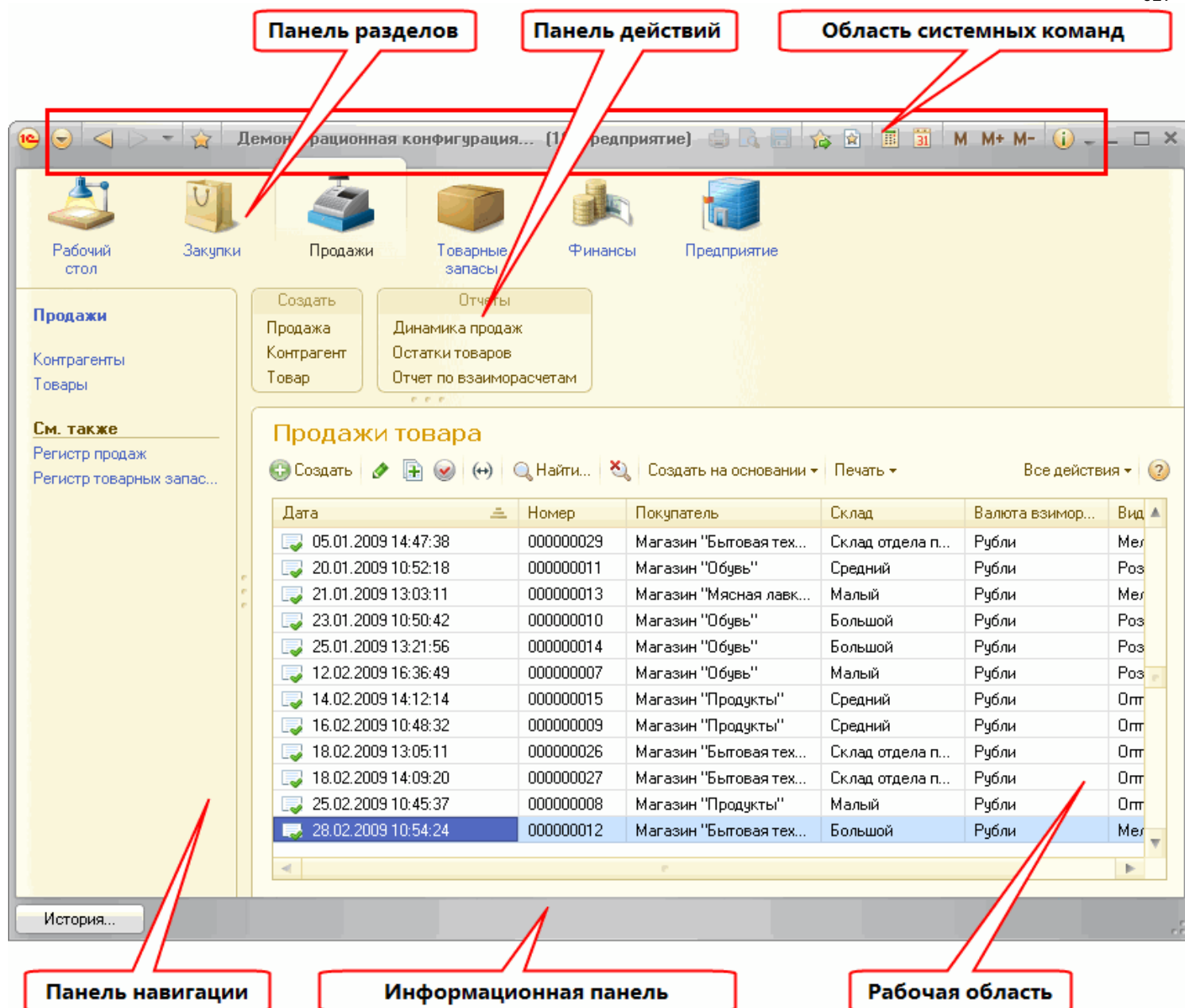


Рис. 29. Основное окно программы в режиме «Формы в отдельных окнах»

Окно устроено таким образом, что позволяет достаточно быстро находить необходимые области и вызывать требуемые команды. Для изменения размеров основного окна можно использовать небольшую область в правом нижнем углу (с тремя утопленными точками).

Приемы работы с панелями разделов, навигации, действий и информационной панелью аналогичны приемам работы в режиме в закладках.

В данном режиме при переходе в новый раздел обновляется как содержимое панели навигации и панели действий, так и содержимое рабочей области.

Команды панели навигации открывают формы в рабочей области основного окна, замещая друг друга.

Команды панели действий и информационной панели открывают формы в новом окне.

4.2.2. Область системных команд

Заголовок приложения является составным и содержит область системных команд.

В левой части **заголовка** расположено главное меню, кнопки перехода между областями программы ([Назад](#), [Вперед](#), [Список переходов](#)), меню работы с избранным ([Избранное](#)).

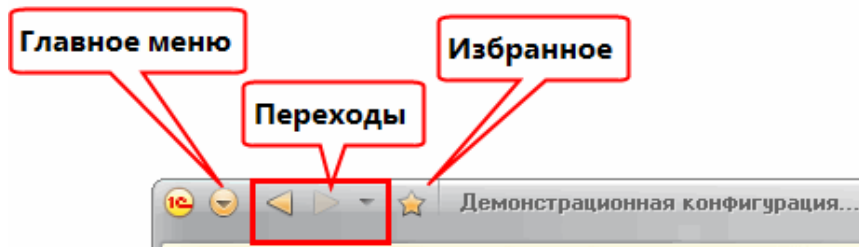


Рис. 30. Командная панель основного окна в режиме «Формы в отдельных окнах»

Чтобы вернуться к предыдущей области программы, следует нажать [Назад](#). Чтобы из предыдущей области программы перейти к следующей, нажать [Вперед](#). В списке [Переходы](#) содержатся все формы, которые были открыты во время сеанса.

Содержимое правой части области системных команд аналогично содержимому правой части области системных команд в режиме [Формы в закладках](#). Подробнее см. [здесь](#).

4.2.3. Вспомогательное окно

Описание работы во вспомогательном окне см. [здесь](#).

Глава 5. Работа в формах

В «1С:Предприятии» пользователь работает с формами. Форма может содержать различные элементы, с помощью которых информация отображается и может быть изменена пользователем: поля ввода, командные панели, кнопки, флажки, закладки, таблицы, различные диаграммы, поля текстового, табличного и HTML-документов, поля географической и графической схем. Данная глава содержит описание общих принципов работы с формами и элементами форм.

5.1. Общие принципы работы с формами

Существует несколько общих элементов, которые имеет каждая форма. Обычно в заголовке формы отображается название объекта или списка объектов, которые она отображает. Название формы задается в процессе настройки конфигурации и не может быть изменено пользователем.

Для **выбора объекта** следует нажать левую кнопку мыши, установив указатель мыши на объект, при работе с клавиатуры использовать клавиши управления курсором.

Если поле объекта имеет затененный фон, это означает, что объект не доступен для редактирования.

Для **перехода к редактированию объекта** следует дважды нажать левую кнопку мыши или клавишу [Enter](#).

Элементы формы могут иметь подсказку, в которой при настройке конфигурации был описан их смысл. Чтобы получить подсказку по конкретному реквизиту формы, следует поместить над ним указатель мыши. Надпись, поясняющая назначение реквизита, появится возле указателя через 1–2 секунды (если такая надпись была создана при настройке формы).

Перемещение по элементам формы. Для перемещения по элементам формы можно использовать клавиши [Tab](#) и [Shift + Tab](#) или просто щелкать мышью на соответствующем элементе диалога.

Некоторые элементы могут быть настроены таким образом, что для перехода по [Tab](#) и [Shift + Tab](#) они недоступны (например, командная панель). Для перехода к таким элементам следует использовать мышь или комбинацию клавиш [Alt + F10](#).

Также можно завершать ввод очередного реквизита нажатием клавиши [Enter](#), тогда курсор будет автоматически переходить на следующий элемент диалога. Последовательность перехода определяется при разработке конфигурации.

Если это обусловлено конфигурацией, форма может проверять корректность и полноту введенных данных. В случае неправильного ввода могут появляться соответствующие сообщения.

Кроме того, после завершения ввода реквизита системой могут выполняться некоторые автоматические действия, например, заполнение значений других реквизитов формы.

Сохранение параметров формы. Если форма может изменять размер, то при закрытии и ее повторном открытии форма будет иметь те же размеры и располагаться в том же месте экрана, что и при ее закрытии. Элементы управления при этом сохранят размеры, какими они были при закрытии формы.

Внешний вид формы определяется конфигурацией и может быть изменен пользователем, если доступна возможность настроить форму (пункт меню [Все действия – Изменить форму](#)). Порядок настройки форм излагается в разделе [«Настройка формы»](#) [здесь](#).

5.2. Поля

Поля предназначены для просмотра, ввода и редактирования данных различных типов. Чтобы ввести значение в поле, нужно привести курсор на область поля, нажать левую кнопку мыши и ввести значение – обычно с помощью клавиатуры или выбором из списка.

5.2.1. Поле ввода

Приемы работы с полем ввода зависят от типа данных, редактируемых или просматриваемых в элементе

формы, а также от настроек, заданных в конфигурации.

Рис. 31. Пример формы с полями ввода

Обычно для того чтобы ввести данные в поле, достаточно перейти к нужному полю ввода, нажав левую кнопку мыши или клавишу **Tab**. При этом доступны возможности редактирования символьных строк, предоставляемые операционной системой, например, работа с буфером обмена.

Красное подчеркивание означает, что данное поле необходимо заполнить.

Затененный фон у поля означает, что данное поле заполняется автоматически или не рекомендуется к заполнению (например, поле **Код** на [рис. 31](#) заполняется при записи автоматически). При попытке изменить значение в таком поле вручную система выдаст соответствующее предупреждение.

В случае ввода некорректных данных в выпадающем списке отобразится информация о том, что значение не найдено.

Нажатием клавиши **F4** можно выбрать элемент из списка элементов для выбора (см. [рис. 32](#)).

Рис. 32. Выбор группы контрагентов из списка

С помощью сочетания клавиш **Ctrl + Shift + F4** можно открыть форму выбранного элемента.

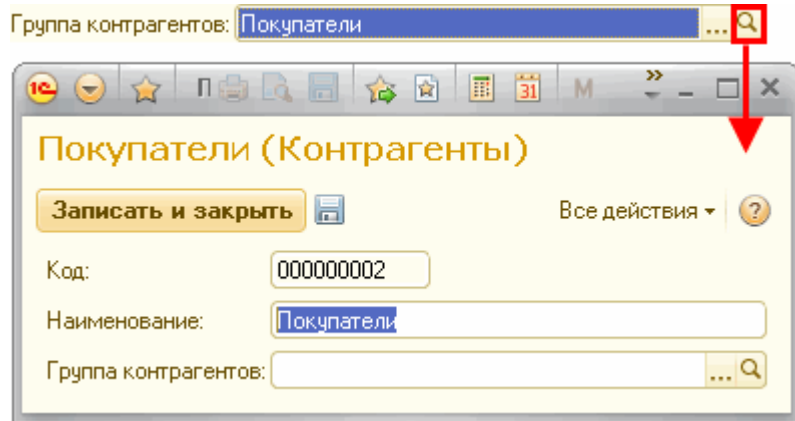


Рис. 33. Открытие формы выбранного элемента из поля ввода

Затемненный фон поля ввода (как у поля **Код** на [рис. 33](#)) означает, что редактировать значение в этом поле не следует.

Если для поля ввода установлен **многострочный режим**, то при нажатии клавиши **Enter** управление не переходит к следующему элементу формы, а в данном поле ввода начинается новая строка.

Для многострочного поля ввода может быть установлен расширенный режим ввода, при котором допускается использование символа табуляции, а также стандартный поиск текста. Следует учитывать, что для данного режима недоступен переход к следующему элементу формы по клавише **Tab**.

Если поле ввода имеет ограниченный размер данных, то при вставке в поле ввода данных из буфера обмена будет взята такая часть исходного текста, чтобы заполнить остаток места (или выделенный текст).

5.2.1.1. Строка

Если поле ввода связано со строковыми данными, то оно будет иметь вид:



Рис. 34. Пример поля ввода типа «Строка»

В незаполненном поле ввода, связанном со строковыми данными, в зависимости от конфигурации может отображаться подсказка ввода. Подсказка отображается в пустом поле ввода, скрывается при вводе текста и не влияет на вводимое значение.



Рис. 35. Подсказка в поле ввода

5.2.1.2. Число

Если поле ввода связано с числовыми данными (например, количество, сумма), то кнопка выбора будет иметь форму калькулятора, и при нажатии на нее число можно отредактировать с помощью отобразившегося калькулятора.

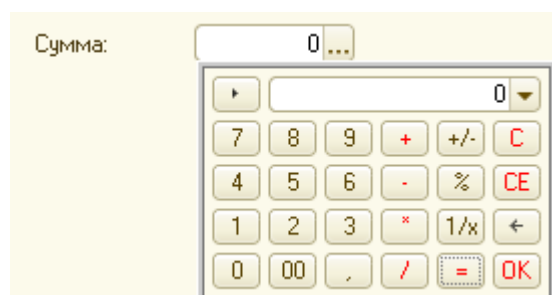


Рис. 36. Ввод числа в поле ввода при помощи калькулятора

Также для поля может быть настроено управление значением с помощью регулятора.



Рис. 37. Изменение значения в поле ввода с помощью регулятора

5.2.1.3. Дата

Если поле ввода предназначено для ввода даты, то в зависимости от настроек программы пустое поле даты будет отображать разделительные точки и двоеточия.

При вводе даты можно использовать встроенный календарь:

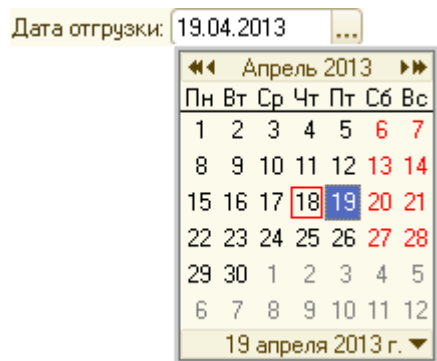


Рис. 38. Ввод даты в поле ввода с помощью календаря

Если поле ввода содержит дату и время, а отображается только дата или дата и часть времени, то при редактировании даты в поле ввода неотображаемая часть времени остается такой же, какой была до изменения.

5.2.1.4. Данные различных типов

Помимо рассмотренных выше типов значения реквизита, в форме можно вводить значения, которые могут относиться к одному из нескольких типов. В поле ввода такого реквизита присутствует кнопка **Выбрать**. При нажатии этой кнопки на экран будет выведен список типов, из которого требуется выбрать необходимый.

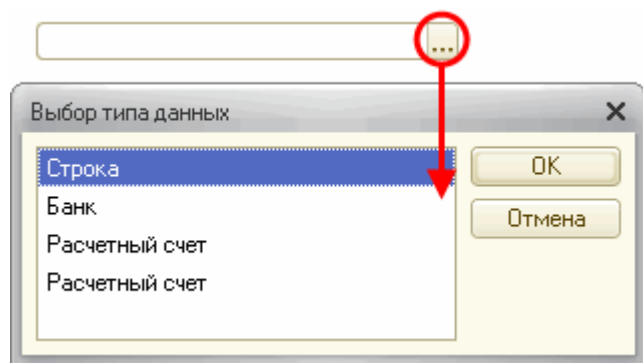


Рис. 39. Ввод данных выбранного типа

В этом окне необходимо выбрать наименование требуемого типа. Далее следует в открывшемся списке выбрать требуемое значение.

5.2.1.5. Выбор значения объекта

Значение объекта можно выбрать без открытия отдельного окна – в списке, который всплывает при нажатии кнопки выбора рядом с редактируемым реквизитом.

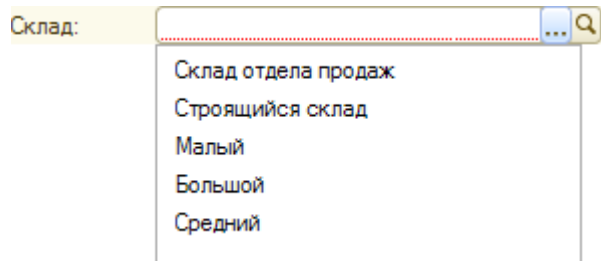


Рис. 40. Поле ввода со всплывающим списком

СОВЕТ. Также для выбора элемента можно использовать клавишу **F4**, а с помощью сочетания клавиш **Ctrl + Shift + F4** можно открыть форму выбранного элемента.

Необходимую строку списка можно выбрать, щелкнув на ней левой кнопкой мыши.

Всплывающий список поля ввода может хранить историю выбранных значений. Выбранные ранее значения отображаются в списке, который вызывается при нажатии клавиши **Ctrl + Стрелка вниз**.

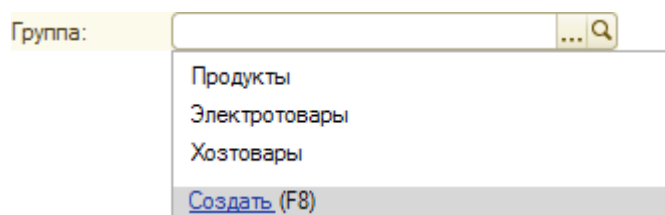


Рис. 41. История вводимых значений

5.2.1.6. Поиск значения в поле ввода

При позиционировании в поле ввода может быть предложено ввести строку для поиска, отобразить целиком список данных (команда **Показать все** или клавиша **F4**) и/или создать новый объект (команда **Создать** или клавиша **F8**).

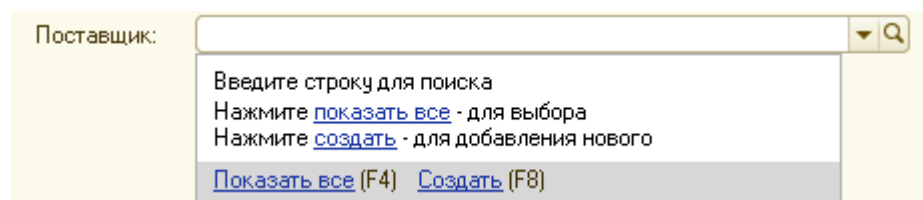


Рис. 42. Всплывающий список в поле ввода

Если введенного значения в списке нет, программа предложит отобразить все значения или создать новый объект.

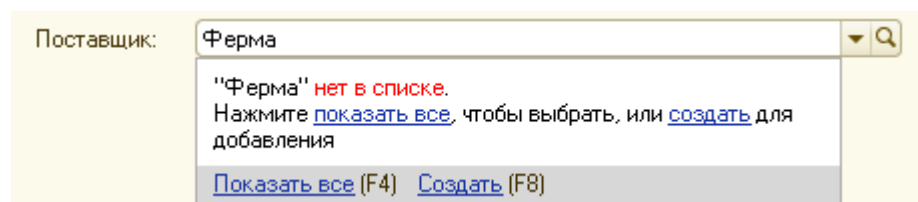


Рис. 43. Значение в списке не найдено

При начале ввода текста программа находит данные, отвечающие набранному тексту, и выводит их в виде выпадающего списка. В зависимости от настроек программы поиск может выполняться по началу строки или по любой части строки. Например:

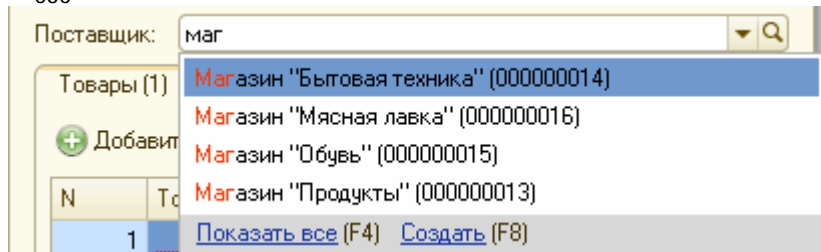


Рис. 44. Поиск по началу строки в поле ввода

Если для данных в поле установлена возможность поиска по любой части, достаточно ввести любой фрагмент текста, как показано на рисунке:

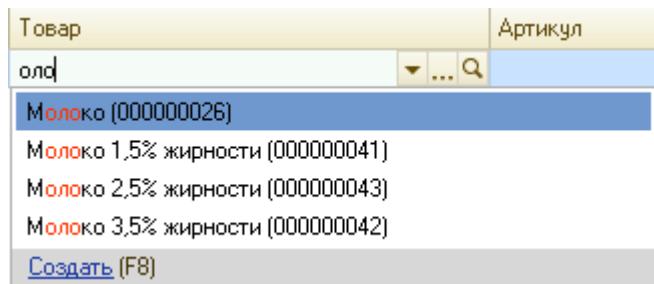


Рис. 45. Поиск по любой части строки в поле ввода

Если данных достаточно много и их обработка требует значительного количества времени, то во время поиска будет отображено соответствующее сообщение.

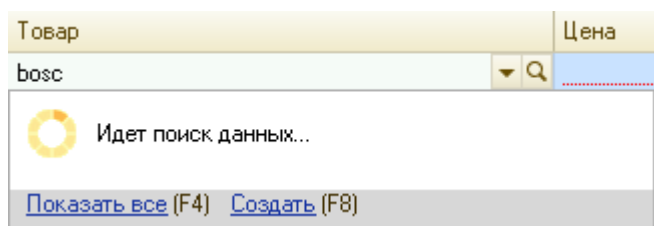


Рис. 46. Поиск данных в фоне

Если в поле выбраны данные, помеченные на удаление, система также выведет об этом предупреждение.

Если набранному тексту не соответствует ни один объект, то система предложит выбрать одно из действий: выбрать из всплывающего списка, показать все или создать новый объект (если такая возможность установлена в программе).

При выборе команды **Создать** (или при нажатии клавиши **F8**) откроется форма создания нового объекта.

Во время выбора в выпадающем списке поля ввода при щелчке мышью на другом элементе формы выпадающий список закроется, а значение, установленное в нем, не изменится.

ПРИМЕЧАНИЕ. Для реквизитов, тип которых не определен (например, выбран составной тип), программа не позволит ввести текст до тех пор, пока не будет указан конкретный тип данных.

Полнотекстовый поиск

При вводе данных в поле ввода может быть использован полнотекстовый поиск. Для поиска данных можно ввести одно или несколько слов. Поиск будет выполняться по любому из введенных слов.

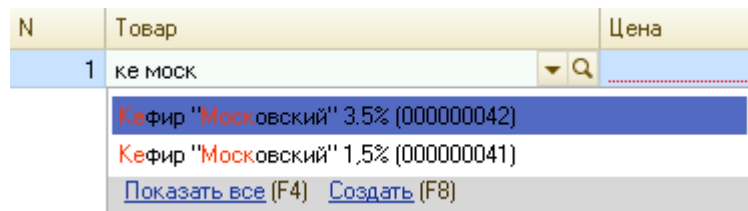


Рис. 47. Полнотекстовый поиск в поле ввода

Во время полнотекстового поиска будут найдены строки, содержащие искомые слова во всех сочетаниях. Будут найдены как целые слова, так и строки, в которых вводимые строки являются частью целых слов (аналогично действию оператора * полнотекстового поиска).

Отказ от выбранного значения

Для очистки значения, которое вводится путем выбора из списка, следует нажать кнопку очистки, расположенную справа от поля реквизита (если такая кнопка определена при конфигурировании), или клавиши **Shift + F4**.



Рис. 48. Поле ввода с кнопкой очистки

Для отмены ввода и возвращения к предыдущему значению в поле, следует нажать клавишу **Esc**.

В многострочном поле ввода при нажатии клавиши **Esc** на экране отобразится вопрос о сохранении изменений. Чтобы сохранить выполненные изменения, следует нажать кнопку **Да**. Для отказа от изменений следует нажать **Нет**. Чтобы вернуться к редактированию поля, следует нажать кнопку **Отмена**.

5.2.2. Флажок

Элемент формы **Флажок** предназначен для включения или выключения некоторого значения.

Используются флажки, имеющие два либо три состояния.

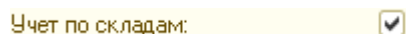


Рис. 49. Пример флажка на форме

Если флажок, расположенный в колонке таблицы или на форме, недоступен для редактирования, то он отображается особым способом.

Для переключения значения флажка следует щелкнуть по нему мышью или (при использовании клавиатуры) активизировать его клавишами **Tab** и **Shift + Tab** и нажать **Пробел**.

Для флажка с двумя состояниями также поддерживается переключение клавишами **+** (плюс) и **-** (минус).

5.2.3. Переключатель

Элемент формы **Переключатель** предназначен для выбора одного из нескольких возможных значений. Требуемое значение выбирается с помощью мыши. При использовании клавиатуры следует активизировать текущее выбранное значение переключателя клавишами **Tab** и **Shift + Tab**, клавишами управления курсором выбрать требуемое значение и нажать клавишу **Enter**. Например, в форме товара предлагается выбор вида товара:



Рис. 50. Переключатель вида товара

5.3. Тумблер

Элемент формы **Тумблер** предназначен для переключения значений (как переключатель) или для включения значения (аналогично флажку). В самом простом виде тумблер выглядит, как показано на рисунке:

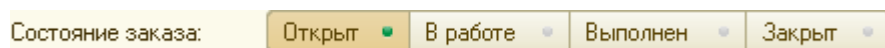


Рис. 51. Тумблер состояния заказа

Чтобы выбрать значение, следует нажать соответствующую кнопку тумблера. У нажатой кнопки активизируется зеленый индикатор, и она будет отображена более ярко.

5.4. Индикатор

Индикатор используется только для отображения текущего состояния реквизита формы, с которым он связан (определяется при проектировании конфигурации).



Рис. 52. Индикатор

5.5. Полоса регулирования

Элемент формы **Полоса регулирования** предназначен для быстрого и наглядного ввода числовых значений. По внешнему виду он напоминает обычный линейный регулятор (на рисунке под полем ввода, с которым он связан), расположенный вертикально или горизонтально.

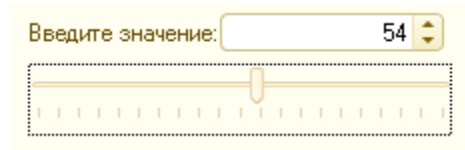


Рис. 53. Регулятор значения

Значение реквизита, связанного с этим элементом, определяется положением бегунка и значениями минимума и максимума, установленными при проектировании конфигурации.

Перемещать бегунок можно клавишами управления курсором, но удобнее – с помощью мыши.

5.6. Гиперссылка

В форме могут располагаться элементы управления, представляющие собой текст, при наведении указателя мыши на который указатель принимает вид руки (см. [рис. 54](#)) и появляется подчеркивание. Такой элемент называется **Гиперссылка**. Он предназначен для перехода к некоторому объекту (например, открытия других форм). Для выбора гиперссылки следует щелкнуть по ней мышью или сделать ее активной клавишами **Tab** и **Shift + Tab** и нажать клавишу **Enter**. Настройка отработки нажатия гиперссылки определяется при проектировании конфигурации. Гиперссылку могут иметь и картинки, размещенные в форме. При поднесении указателя мыши к картинке с гиперссылкой форма указателя также принимает вид, как на [рис. 54](#).

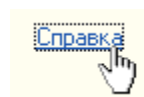


Рис. 54. Гиперссылка на форме

5.7. Кнопка

Кнопки в форме используются для выполнения некоторых действий, предусмотренных алгоритмом формы. Для выбора кнопки следует щелкнуть по ней мышью или сделать ее активной клавишами **Tab** и **Shift + Tab** и нажать клавишу **Enter**.

Одна из кнопок в форме может быть предопределенной, и ее нажатие может выполняться не только мышью, но и с помощью клавиш **Ctrl + Enter**.

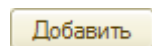


Рис. 55. Кнопка на форме

Для некоторых кнопок могут быть предусмотрены другие комбинации клавиш, которые позволяют нажать эту кнопку без использования мыши и без перехода к ней последовательным обходом элементов формы.

Для некоторых кнопок может быть задан режим отображения нажатия. Такие кнопки используются при установке или отмене какого-либо режима.

Если действие, связанное с кнопкой, недоступно, кнопка также будет отображаться как недоступная.

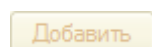


Рис. 56. Кнопка недоступна

5.8. Таблица

Форма может содержать таблицу, предназначенную для просмотра списка с несколькими колонками. Состав колонок определяется назначением конкретной формы, а строки являются собственно данными, которые просматриваются в списке и в некоторых случаях могут редактироваться. В качестве примера можно привести список номенклатуры в документах [Продажа](#).

Продажа 000000012 от 16.03.2009 11:29:08

Провести и закрыть | Провести | Создать на основании | Все действия

Номер: 000000012 Вид цен: Мелкооптовая

Дата: 16.03.2009 11:29:08

Организация: ООО "1000 мелочей"

Товары (2) | Прочее

N	Товар	Цена	Количество	Сумма
1	Veko345MO	17 500,00	4,00	70 000,00
2	Veko67NE	12 000,00	4,00	48 000,00

Количество (итог): **8,00** Сумма (итог): **118 000,00**

Рис. 57. Список товаров в документе «Продажа»

В некоторых формах колонки располагаются таким образом, что значения колонок в каждой строке таблицы выводятся в несколько уровней (два или более). Такой способ позволяет одновременно выводить в строке большое количество информации без прокручивания таблицы по горизонтали.

Если высота ячейки таблицы равна одной строке, а в ячейке при этом содержится несколько строк

текста, то отображается только первая строка текста, которая заканчивается многоточием.

Добавление, копирование, удаление, изменение данных, сортировка, отбор и другие действия с данными, размещаемыми в списке, обычно осуществляются с помощью кнопок командной панели списка или с помощью команд контекстного меню, вызываемых в области списка, если выполнение подобных действий предусмотрено при конфигурировании.

Приемы работы с таблицей зависят от типа данных, редактируемых или показываемых в элементе формы, а также от настроек, заданных в конфигурации или определенных средствами встроенного языка, и в основном совпадают с правилами работы с полем ввода. Чтобы начать редактирование ячейки, следует нажать клавишу **Enter** или выбрать ячейку и начать ее редактировать. Для завершения редактирования строки таблицы нужно нажать клавиши **Shift + F2**.

Просмотр таблицы. При просмотре таблицы можно использовать мышь и клавиатуру.

Для удобства просмотра содержимого таблицы можно менять ширину колонок. Для этого следует подвести указатель мыши к границе колонок, пока он не изменит вид, и нажать левую кнопку мыши. Перемещая мышь, установить нужную ширину колонок, затем отпустить кнопку мыши. Установленные размеры колонок запоминаются. Если при изменении ширины колонки нажать клавишу **Ctrl**, то будет изменяться ширина только данной колонки. Данный прием используется, если для каких-либо системных списков не предусмотрены заголовки.

Если ширина колонки не позволяет увидеть всего содержимого поля, то его можно посмотреть, воспользовавшись следующим способом. Поместить указатель мыши над нужной колонкой в нужной строке, и возле указателя появится содержимое поля полностью.

При просмотре таблицы порядок следования колонок можно изменить. Если необходимо установить порядок следования колонок, следует воспользоваться настройкой формы, описанной [здесь](#).

В некоторых таблицах допускается использование множественного выбора строк списка. Удерживая нажатой клавишу **Ctrl**, можно выбрать необходимые строки с помощью мыши или клавиш управления курсором.

Для листания таблицы постранично можно использовать клавиши **PgUp** и **PgDown**.

Чтобы перейти на первую строку таблицы, следует нажать клавиши **Ctrl + Home**. Чтобы перейти на последнюю – **Ctrl + End**.

Корректировка ячейки таблицы. Для корректировки реквизита таблицы следует выбрать ячейку, содержащую этот реквизит, и начать редактирование. Если для ячейки предусмотрен список значений, следует нажать **Ctrl + Стрелка вниз**, чтобы открыть его и выбрать значение из списка. Для окончания редактирования ячейки нужно нажать клавишу **Enter**. Если для редактирования данных в ячейке используется поле ввода, то доступны приемы работы для поля ввода. Подробнее см. [здесь](#).

Ввод строки. Для ввода в таблицу новой строки следует нажать клавишу **Ins**. Новая ячейка строки автоматически переключается в режим редактирования. Нажатие клавиши **Enter** после указания значения реквизита переводит курсор в следующую ячейку. Возможность перехода между ячейками строки во время редактирования с помощью повторного нажатия **Enter** или клавиши **Tab** устанавливается на этапе конфигурирования.

Также новая строка может быть создана автоматически при нажатии клавиши **Стрелка вниз** в последней строке таблицы, если такая возможность была установлена в конфигурации. Если при редактировании новой строки не была введена информация ни в одну ячейку, то новая строка будет удалена.

ПРИМЕЧАНИЕ. Если при вводе строки в таблицу курсор после нажатия клавиши **Enter** «перескакивает» через ячейки, значит, в данной конфигурации установлен пропуск этих колонок таблицы.

Удаление строки. Для удаления строки таблицы клавишами управления курсором или мышью выберите строку и следует нажать клавишу **Del**.

ВНИМАНИЕ! Удаление строки производится без дополнительного предупреждения.

Копирование строки. Новую строку в таблицу можно ввести путем копирования уже существующей строки. Для этого выберите клавишами управления курсором или мышью строку, которая будет служить образцом, и следует нажать клавишу **F9**. В таблицу будет добавлена новая строка, данные которой будут заполнены значениями, скопированными из данных строки-образца.

Номера строк. Крайняя левая колонка таблицы содержит номера строк. При вводе новой строки в таблицу ей автоматически присваивается очередной порядковый номер.

На форме может быть определен порядок следования строк таблицы. При этом существует возможность передвигать строки вверх и вниз. Для перемещения строк используются команды [Переместить вверх](#) и [Переместить вниз](#). При перемещении строк их номера автоматически изменяются.

Сортировка строк таблицы. Строки в таблице могут быть упорядочены произвольным образом, если это обусловлено конфигурацией системы. Можно установить сортировку строк по данным любой колонки, выбрав колонку и вызвав пункт меню [Все действия – Сортировать по убыванию](#) или [Сортировать по возрастанию](#). Также существует возможность перетаскивать строки таблицы с помощью мыши.

Если необходимо установить более сложное упорядочивание элементов таблицы, следует воспользоваться настройкой формы, описанной [здесь](#).

5.9. Группы

Система предоставляет возможность объединять различные элементы формы в группы. Это могут быть группы полей, колонок списка, страницы, группы команд. Также колонки списка могут быть объединены в вертикальные или горизонтальные группы. В качестве примера вертикальная группа колонок списка на форме будет выглядеть следующим образом:

N	Товар
	Артикул
1	Сапоги
	0Б-001

Рис. 58. Группа колонок списка

Элементы на форме могут визуально распределяться по группам. Группа может быть обособлена чертой или выделена рамкой. Например, как показано на рисунке.

Номер:	<input type="text" value="000000007"/>
Дата:	<input type="text" value="12.02.2009 16:36:49"/>
Организация:	<input type="text" value="ООО 'Товары'"/> ...
Покупатель:	<input type="text" value="Магазин 'Обувь'"/> ...
Склад:	<input type="text" value="Малый"/> ...

Рис. 59. Группа полей на форме

Элементы формы можно перемещать между группами. При этом системой определяется допустимость такого переноса. Если перенос требует изменения каких-либо свойств элемента, изменение осуществляется автоматически. Например, при переносе группы элементов в группу с видом [Страницы](#) ее свойство [Вид](#) изменится с [Обычная группа](#) на [Страница](#). Если в результате изменились требования к подчиненным элементам, они либо автоматически изменяются, либо удаляются.

На форме могут располагаться группы, которые можно свернуть. Такие группы отображаются особым образом: заголовок содержит ссылку или кнопку для сворачивания группы.

	Адрес	<input type="text"/>
Регион:	<input type="text" value="Москва"/>	...

Рис. 60. Сворачиваемая группа

При нажатии на заголовок такой группы или кнопку рядом с ним ее содержимое скроется.

Чтобы раскрыть содержимое группы, следует снова нажать на заголовок свернутой группы.

Подробнее о настройке форм см. раздел [«Настройка формы»](#) [здесь](#).

5.9.1. Командная панель

Форма может содержать группы элементов вида [Командная панель](#). Каждая командная панель представляет собой набор кнопок (команд), действие которых распространяется только на данную форму или на отдельную часть формы.

Для активизации командной панели можно использовать комбинацию клавиш **Alt + F10**; панель не участвует в обходе элементов формы по клавише **Tab** или **Shift + Tab**. Обходятся все кнопки, а при нажатии кнопки панели (клавишей **Enter**) выполняется действие.

5.9.2. Страницы

Форма может содержать страницы, если это обусловлено конфигурацией. Закладки страницы могут быть расположены вертикально (сверху, снизу) или горизонтально (слева, справа) от содержимого страницы. Для выбора определенной страницы необходимо щелкнуть на ней левой кнопкой мыши. Также между страницами можно переключаться, используя сочетания клавиш **Ctrl + PgUp** (вперед) или **Ctrl + PgDn**.

Каждая такая страница может содержать какие-либо «свои» элементы формы. Например, форма документа [Расходная накладная](#) может содержать закладку [Товары](#), на которой расположен список товаров, и закладку [Прочее](#) с полем для дополнительного описания отгрузки (как показано на [рис. 61](#)).

N	Товар	Цена	Количество	Сумма
1	Veko345MD	17 500,00	4,00	70 000,00
2	Veko67NE	12 000,00	4,00	48 000,00

Количество (итог): **8,00** Сумма (итог): **118 000,00**

Рис. 61. Страницы в документе «Расходная накладная»

5.10. Диаграмма

Диаграмма может быть встроена в форму и в табличный документ.

Диаграммой, встроенной в форму, можно управлять. Настройка производится с помощью реквизитов, размещенных в форме (состав определяется при конфигурировании). Если позволяют исходные данные, допускается выбор другого типа диаграммы. О работе с диаграммой в табличном документе см. раздел [«Работа с диаграммой»](#) в приложении «Редактор табличных документов», поставляемом в электронной версии документации.

5.11. Поле текстового документа

Поле текстового документа предназначено для работы с текстами. В нем размещается окно текстового редактора, предназначенное для ввода и сохранения строк текста. Подробнее о работе с редактором см. приложение «Редактор текстового документа», доступное в электронной версии документации.

5.12. Поле HTML-документа

Поле HTML-документа предназначено для просмотра HTML-документа на форме. При просмотре информации, размещенной в поле, доступен переход по ссылкам (если такие предусмотрены), поиск с помощью команд главного меню **Правка: Найти, Найти следующий, Найти предыдущий**, переход между страницами с помощью команд контекстного меню **Вперед, Назад**.

ПРИМЕЧАНИЕ. Если на поле отображается внешний ресурс, то поиск выполняется средствами браузера.

Для данного поля доступны команды меню **Файл: Печать, Сохранить как**.

5.13. Поле графической схемы

Элемент формы **Поле графической схемы** предназначен для отображения и редактирования графической схемы.

Редактирование графических схем описано в приложении 5 «Редактор графической схемы», доступном в электронной версии документации.

Порядок обхода элементов схемы определяется при конфигурировании.

С помощью пункта **Графическая схема – Масштаб** можно выбрать масштаб отображения графической схемы.

Аналогичные действия по изменению масштаба можно выполнять с помощью мыши, у которой имеется колесо прокрутки, при нажатой клавише **Ctrl**.

В этом поле могут отображаться общие карты маршрута бизнес-процессов или карты маршрута конкретных экземпляров бизнес-процессов с учетом пройденных и активных точек маршрута.

В качестве примера отображения карты маршрута в графической схеме можно привести карту маршрута бизнес-процесса продажи.

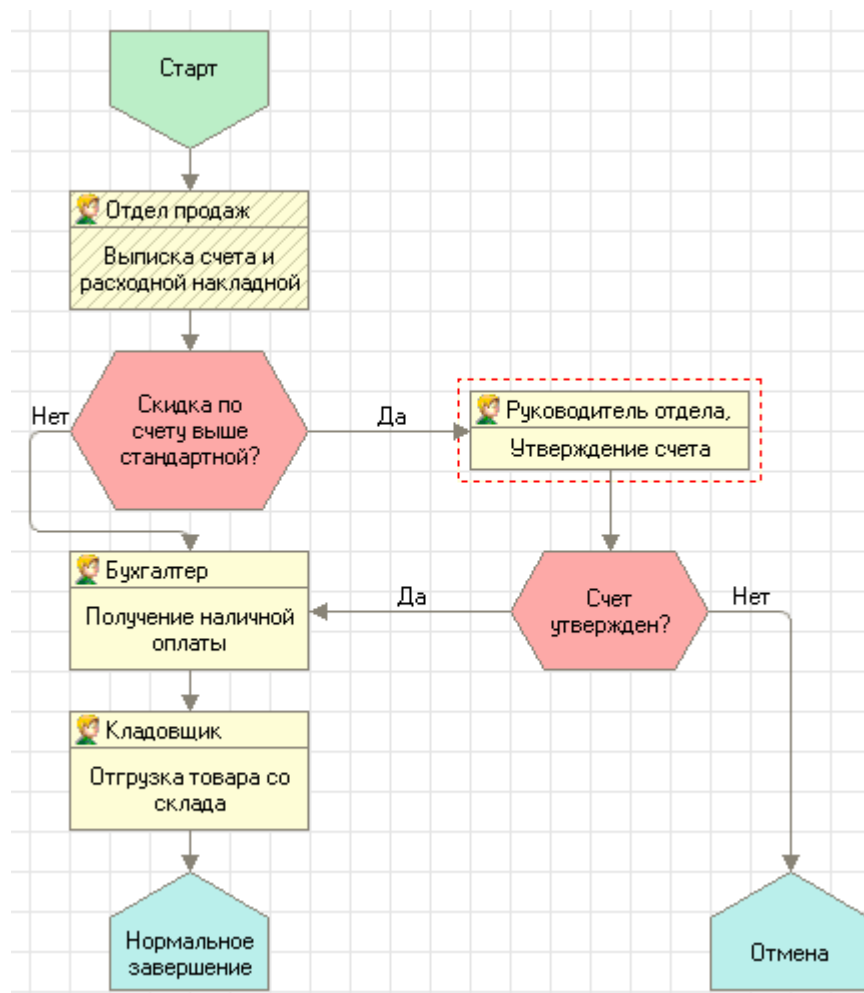


Рис. 62. Графическая схема карты маршрута

Пройденные точки маршрута отображаются заштрихованным фоном (на рисунке – [Выписка счета](#)). Точка маршрута считается пройденной, если все связанные с ней задачи выполнены.

Активные точки обводятся красной пунктирной линией. Точка считается активной, если с ней связана одна или более невыполненная задача (на рисунке – [Утверждение счета](#)).

5.14. Поле картинки

Элемент [Поле картинки](#) предназначен для размещения в форме различных картинок. Этот элемент также может содержать ссылку (см. описание в разделе «[Гиперссылка](#)»).

В некоторых случаях, когда изображение не помещается в поле элемента, при выборе элемента появляются линейки прокрутки, с помощью которых можно увидеть изображение полностью (если такая возможность была установлена в конфигурации). При наличии линейки прокрутки изображение можно двигать также с помощью клавиш управления курсором, [Home](#) (показать верхний левый угол), [End](#) (показать нижний правый угол), [PgUp](#) (перейти вверх на размер окна по вертикали, а при нажатой клавише [Alt](#) – по горизонтали), [PgDn](#) (перейти вниз на размер окна по вертикали, а при нажатой клавише [Alt](#) – по горизонтали).

Если мышь имеет колесо прокрутки, то вертикальное перемещение изображения выполняется прокруткой колеса от себя (вверх) и к себе (вниз). Если при прокрутке нажать клавишу [Shift](#), то выполняется горизонтальная прокрутка картинки.

Для элемента формы [Поле картинки](#) добавляется контекстное меню, состоящее из пунктов [Копировать](#), [Сохранить как](#), [Просмотреть картинку](#).

С помощью пункта меню [Копировать](#) можно скопировать в буфер обмена изображение, после чего его можно вставить в другое приложение.

Меню [Сохранить как](#) доступно для картинок, загруженных из файла. С его помощью можно сохранить картинку в файл на диске.

Если выбран пункт [Просмотреть картинку](#), то на экран выводится окно просмотра картинки, в котором есть кнопки увеличения и уменьшения масштаба, а также установки реального размера. Существует возможность изменять размеры окна. Для выбора нужного масштаба используются пункты меню [Увеличить](#), [Уменьшить](#) и [Оригинальный размер](#). Для изменения масштаба можно также использовать клавиши, расположенные на цифровой клавиатуре. Нажатие клавиши [+](#) (плюс) увеличивает картинку, клавиши [-](#) (минус) – уменьшает. Команда [Оригинальный размер](#) устанавливает реальный размер картинки.

Следует отметить, что для действий, выполняемых с помощью мыши, картинка должна быть «в фокусе», т. е. по ней следует предварительно щелкнуть мышью. Если щелкнуть одну из кнопок, то фокус будет на кнопке, и дальнейшие действия мышью (прокрутка и масштаб) не будут выполняться до тех пор, пока фокус не будет снова установлен на картинке.

Аналогичные действия по изменению масштаба можно выполнить с помощью мыши, у которой имеется колесо прокрутки, при нажатой клавише [Ctrl](#). Для увеличения масштаба необходимо крутить колесо от себя, для уменьшения – на себя. Изменение масштаба также можно выполнять с клавиатуры, используя клавиши [+](#) (плюс) и [-](#) (минус), расположенные справа от цифровой клавиатуры.

5.15. Поле табличного документа

Форма может содержать встроенный табличный документ (элемент формы [Поле табличного документа](#)). О работе с элементом формы [Поле табличного документа](#) см. в приложении 3, поставляемом в электронной версии документации.

Перемещение по таблице. Окно табличного документа в любой момент времени показывает только часть таблицы, расположенной в этом окне. Для выбора ячейки таблицы используется мышь. Выбранная ячейка становится активной. Для вывода на экран областей, не поместившихся в границах окна, используются линейки прокрутки.

СОВЕТ. Чтобы прокрутить документ по горизонтали, следует, удерживая клавишу **Shift**, прокрутить колесо мыши.

Для удобства прокрутки большого документа можно нажать колесо мыши и выбрать курсором направление прокрутки. Скорость прокрутки зависит от расстояния между текущим положением курсора и тем местом, где было нажато колесо (отображается специальным образом).

Управление видом таблицы. При помощи пунктов подменю **Таблица – Вид** можно управлять отображением различных областей и компонентов табличного документа: заголовков строк и колонок, сеткой таблицы и т. д.

Кратко поясним назначение пунктов меню **Вид**. Каждый пункт работает как переключатель: выбор (нажатие) вызывает или прекращает отображение соответствующей области.

Пункт меню	Действие
Зафиксировать таблицу	Фиксирует верхние строки и левые колонки табличного документа таким образом, чтобы при перелистывании табличного документа они постоянно присутствовали на экране
Отображать сетку	Включает и отключает показ сетки табличного документа – тонких линий, разделяющих ячейки таблицы
Отображать заголовки	Управляет показом заголовков строк и колонок табличного документа
Отображать группы	Включает и отключает показ групп – областей слева от заголовков строк и сверху от заголовков колонок
Редактирование	Включает и отключает запрет на любые изменения в табличном документе
Отображать примечания	Включает и отключает режим показа примечаний
Черно-белый просмотр	Отключает и включает показ цветов, заданных для табличного документа. При включении этого пункта (в меню напротив него появляется галочка) все цвета будут определяться установками операционной системы Microsoft Windows
Режим просмотра страниц	Отключает и включает показ минимального количества строк (определяется введенной информацией), области печати, разрывов страниц
Масштаб	Дискретно масштабирует изображение. При выборе этого пункта открывается подменю, в котором можно выбрать желаемый масштаб изображения табличного документа

СОВЕТ. Для масштабирования можно использовать мышь с колесом прокрутки. Следует нажать клавишу **Ctrl** и, не отпуская ее, повернуть колесо. Движением колеса от себя вы увеличиваете масштаб, к себе – уменьшаете.

Фиксация таблицы (шапки и боковой части). Редактор табличных документов системы «1С:Предприятие» позволяет зафиксировать верхние строки и левые колонки табличного документа таким образом, чтобы при прокрутке табличного документа они постоянно присутствовали на экране.

Чтобы зафиксировать верхние строки табличного документа, следует выделить строку табличного документа целиком и выбрать пункт **Таблица – Вид – Зафиксировать таблицу**. Зафиксированы будут все строки, расположенные выше выделенной строки.

Чтобы зафиксировать колонки табличного документа, следует выделить колонку табличного документа целиком. Зафиксированы будут колонки, расположенные слева от выделенной колонки.

Чтобы зафиксировать одновременно строки и колонки табличного документа, следует выделить одну ячейку табличного документа. Зафиксированы будут строки и колонки, расположенные соответственно выше и левее выделенной ячейки.

Для отмены фиксации следует повторно выбрать пункт **Зафиксировать таблицу**.

Защита от редактирования. Для запрета редактирования табличного документа используется пункт [Таблица – Вид – Редактирование](#), который должен быть отключен.

Если ячейка содержит текст, который полностью не помещается в видимой области, то при подведении указателя мыши текст показывается в виде подсказки. Если размер текста достаточно большой, чтобы отобразить его в виде подсказки, показывается только часть текста.

5.16. Поле форматированного документа

На форме может быть расположен элемент вида [Поле форматированного документа](#), предназначенный для хранения текстового документа, включающего элементы форматирования: выделение цветом, изменение шрифта текста, гиперссылки, картинки и т. д.

Такое поле может использоваться, например, при создании сообщения электронной почты или для редактирования простейшего html-документа.

Если поле форматированного документа находится в режиме [Только просмотр](#), то пользователь может переходить по ссылкам, которые содержит документ. Переход выполняется в новом окне браузера.

С помощью команд главного меню [Файл – Сохранить](#) и [Сохранить как](#) форматированный документ может быть сохранен как текстовый файл или html-документ.

Форматирование документа выполняется с помощью команд командной панели форматированного документа.

Кратко перечислим предоставляемые возможности форматирования.

Изменение шрифта. Шрифт документа можно изменить двумя способами:

- выбрать команду [Изменение Шрифта](#), в диалоге [Выбор шрифта](#) выбрать шрифт, размер шрифта, стиль начертания ([Жирный](#), [Подчеркнутый](#), [Наклонный](#), [Зачеркнутый](#)) и нажать кнопку [ОК](#);
- выделить текст, шрифт которого нужно изменить, и с помощью соответствующих команд панели инструментов увеличить или уменьшить размер шрифта пошагово, установить стиль шрифта (полужирный, курсив или подчеркнутый).

Выбор цвета фона и текста. Система «1С:Предприятие» обеспечивает возможность выделить текст цветом фона и цветом текста. Для этого следует выделить текст, на панели инструментов выбрать команду [Изменение цвета фона](#) (чтобы изменить цвет фона текста) или [Изменение цвета текста](#) (чтобы изменить цвет текста). В открывшемся диалоге [Выбор цвета](#) выбрать необходимый цвет и нажать кнопку [ОК](#).

Вставка гиперссылки. Текст форматированного документа может содержать ссылки.

Нужно выделить объект (текст или картинку), выбрать пункт [Вставить гиперссылку](#). На экран выводится диалог формирования. В поле [Ссылка](#) указать адрес гиперссылки и нажать кнопку [ОК](#).

Выделенный объект форматируется в соответствии со стилем гиперссылки. Обычно это синий цвет текста с подчеркиванием.

Вставка картинки. В форматированный документ можно вставлять картинки.

Для вставки картинки нужно выбрать пункт [Вставить картинку](#) и выбрать файл в стандартном диалоге выбора файла.

Вставка символов. Для добавления в текст символов из списка используется команда [Вставить символ](#).

Вставка списка. Чтобы начать маркированный список, используется команда [Маркированный список](#); чтобы начать нумерованный – [Нумерованный список](#). Затем нужно ввести текст.

Чтобы добавить следующий элемент списка, следует нажать клавишу [Enter](#).

Для завершения списка нужно нажать клавишу [Enter](#) дважды или нажать клавишу [Backspace](#), чтобы удалить последний маркер или цифру списка.

Настройка выравнивания. Текст документа может быть выровнен по левому краю, по правому краю или по ширине документа.

Нужно выделить текст, который необходимо выровнять, и нажать кнопку [Влево/Вправо/По центру/По ширине](#).

Текст с отступами. Можно увеличить или уменьшить отступ от края поля для одного или нескольких абзацев текста.

Чтобы увеличить отступ, нужно выделить текст и нажать кнопку [Увеличить отступ](#). Если необходимо его уменьшить, нажать [Уменьшить отступ](#).

Междустрочный интервал. Чтобы настроить междустрочный интервал, нужно выделить текст и выбрать команду [Междустрочный интервал](#). В открывшемся окне ввести размер интервала с клавиатуры или с помощью кнопок и нажать [ОК](#).

5.17. Поле планировщика

Элемент формы [Поле планировщика](#) предназначен для отображения и редактирования различных данных планирования. В зависимости от конфигурации, данный элемент можно использовать для планирования задач, расписаний, событий, встреч и т.д.

В поле планировщика всегда содержится одно измерение – шкала времени.

12 августа 2014 г.	
08:00	
09:00	✉ 09:17 - 09:37 Проверка почты
10:00	⚠ 10:00 - 11:00 Важное совещание
11:00	
12:00	
13:00	☕ 13:00 - 14:00 Обед
14:00	
15:00	
16:00	
17:00	

Рис. 63. Поле планировщика с периодом отображения "День"

В зависимости от настроек конфигурации, в поле могут использоваться несколько измерений.

Период отображения поля планировщика также может зависеть от конфигурации.

Существует возможность с помощью команд главного меню, расположенных в подменю [Планировщик](#) изменять период, отображаемый в поле планировщика:

- [К началу](#) - перейти к началу периода отображения.
- [Назад](#) - перейти к предыдущему периоду отображения. Например, если выбран период отображения [День](#) и отображается день 12 августа 2014 г., будет выполнен переход к 11 августа 2014 г.
- [Сегодня](#) - перейти к текущему дню.
- [Вперед](#) - перейти к следующему периоду отображения.
- [К концу](#) - перейти к концу периода отображения.

5.17.1. Добавление элемента

В зависимости от настроек конфигурации, существует несколько способов добавления элементов на поле.

Чтобы создать элемент, достаточно выбрать нажатием правой клавиши мыши поле напротив требуемого интервала на оси времени.

Если установлено в настройках конфигурации, при выборе времени появится окно быстрого редактирования элемента.

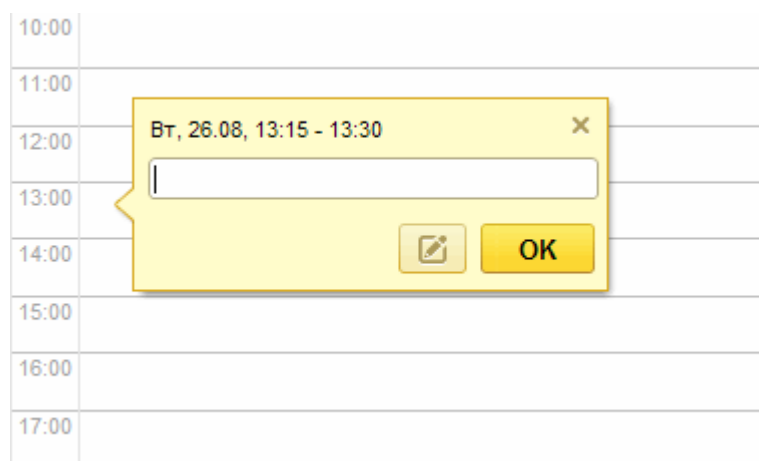


Рис. 64. Создание нового элемента планировщика

Чтобы сохранить информацию, нажмите кнопку **OK**.

5.17.2. Редактирование элемента

Чтобы просмотреть или изменить данные элемента, в контекстном меню или в окне быстрого редактирования следует нажать кнопку **Редактировать**. При этом откроется диалог редактирования элемента.

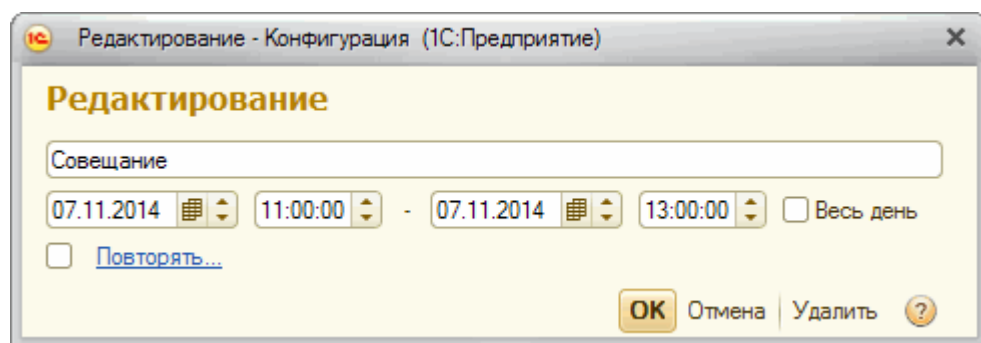


Рис. 65. Редактирование элемента планировщика

В диалоге устанавливается следующая информация об элементе планировщика:

- **Наименование** – наименование элемента планировщика;
- **Период проведения** – Если установлен флажок **Весь день**, время проведения события автоматически устанавливается **00:00 - 23:59** для выбранного дня;
- **Период повтора** -Чтобы установить период повтора события, нажмите ссылку **Повторять...** или установите флажок слева от нее. Расписание повторов события редактируется в отдельном окне.

Для завершения редактирования элемента планировщика нажмите **OK**.

Чтобы отменить внесенные изменения, нажмите **Отмена**.

Чтобы удалить событие, нажмите кнопку **Удалить**.

5.17.2.1. Перемещение элемента

Изменить интервал времени, занятый элементом, можно несколькими способами:

- Можно перетащить элемент на новые дату и время с помощью мыши.
- Изменить интервал времени в диалоге редактирования элемента.

5.17.2.2. Создание повторяющегося элемента

В планировщике существует возможность настроить повторение элемента с заданным интервалом, например,

Для этого установите флажок Повторять... или щелкните по ссылке в окне редактирования элемента планировщика.

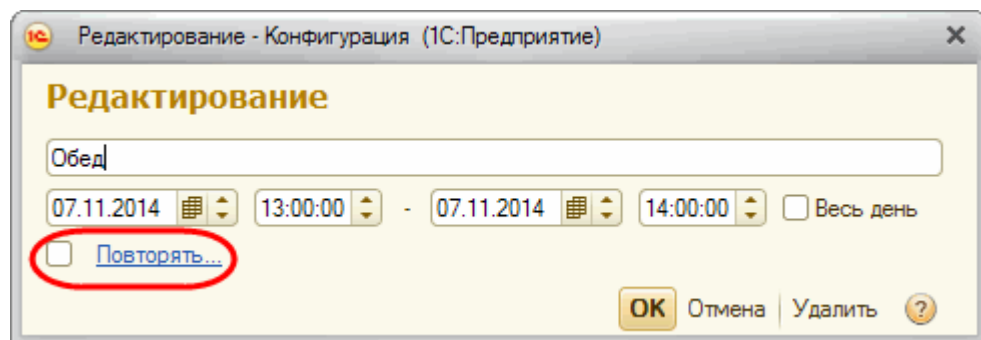


Рис. 66. Создание повторяющегося элемента

В открывшемся окне установите требуемые параметры.

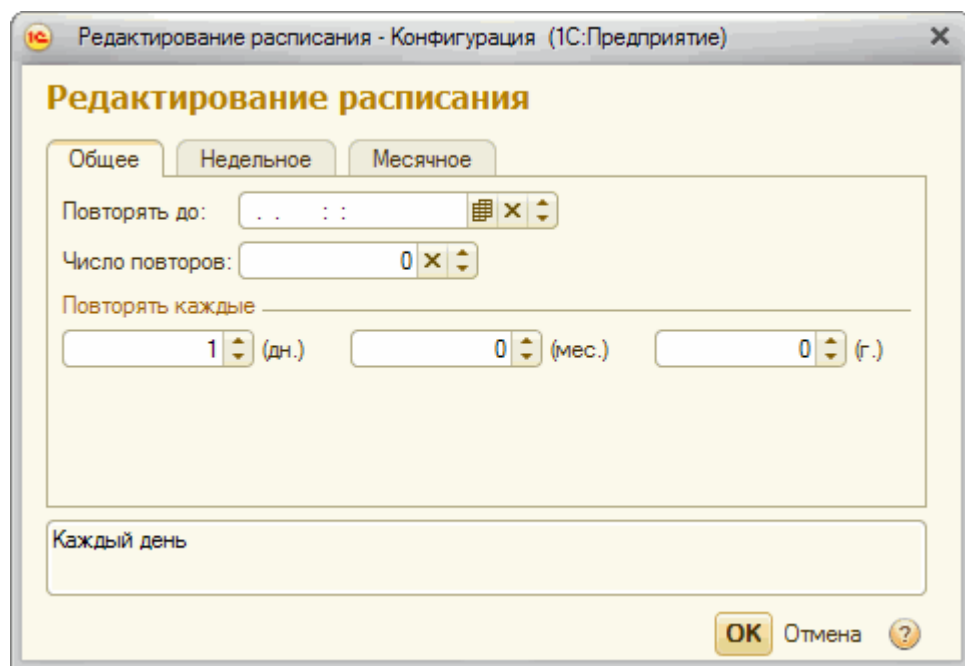


Рис. 67. Редактирование расписания элемента планировщика

На закладке **Общее** указывается дата завершения повторов (в поле **Повторять до:**), число повторов (поле **Число повторов**), интервал повтора в днях, месяцах, годах (**Повторять каждые**)

Проверить правильность настройки можно в нижней части окна редактирования расписания.

На закладке **Недельное** указывается недельное расписание элемента:

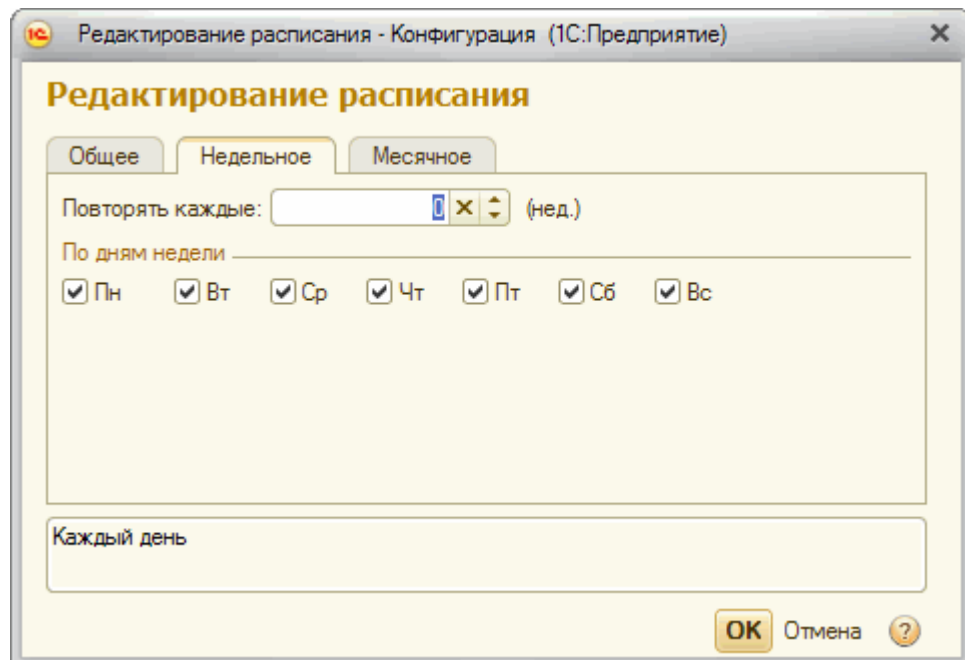


Рис. 68. Редактирование недельного расписания элемента планировщика

В поле **Повторять каждые** устанавливается интервал повтора элемента планировщика в неделях.

В группе **По дням недели** следует установить флажки по тем дням недели, в которых элемент будет повторяться.

На закладке **Месячное** указывается месячное расписание элемента планировщика.

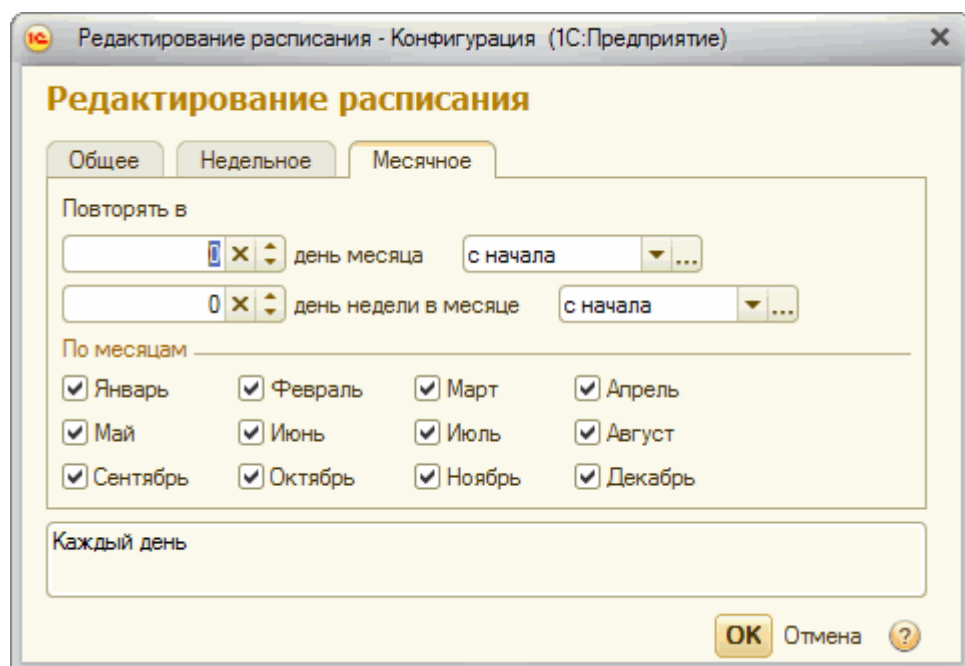


Рис. 69. Редактирование месячного расписания

В группе **Повторять в** следует выбрать номер дня месяца и/или дня недели в месяце и указать порядок отсчета (с начала или конца месяца/недели).

В группе **По месяцам** следует установить флажки по тем месяцам, в которых требуется повторение элемента.

В нижней части окна редактирования расписания представлены сводные данные установленного расписания повторов.

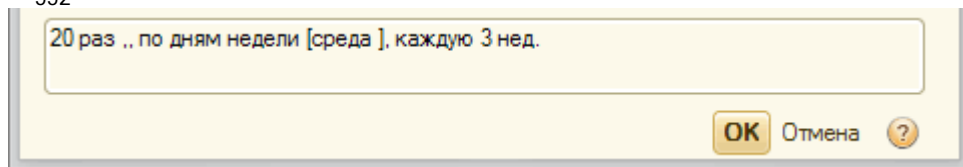


Рис. 70. Сводные данные расписания повторов

5.17.3. Удаление элемента

Данное действие доступно только для элементов планировщика, которые пользователь может редактировать. Чтобы удалить элемент, выберите его правкой кнопкой мыши и в контекстном меню выберите команду **Удалить**.

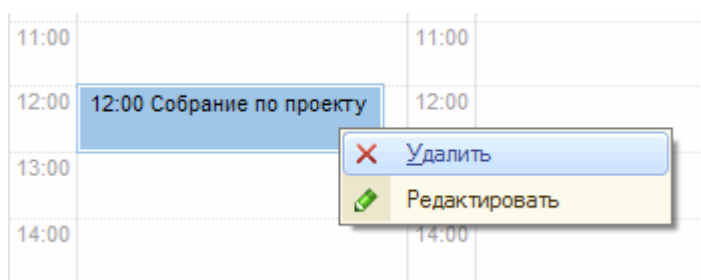


Рис. 71. Удаление элемента планировщика

5.18. Операции перетаскивания

В «1С:Предприятии» реализован механизм поддержки операций перетаскивания. С его помощью можно осуществлять перенос данных между разными элементами формы. Например, можно переносить элементы списка из одной группы в другую, переносить данные из таблицы в поле табличного документа или перенести список выделенных файлов из проводника Microsoft Windows в какой-либо элемент формы.

Операции перетаскивания поддерживают следующие элементы управления: таблица, поле табличного документа, поле календаря и поле картинки.

Возможность использования операций перетаскивания определяется при конфигурировании каждого элемента управления. Указание такой возможности следует искать в описании работы с конфигурацией. Далее приводится описание основных действий для элементов управления, для которых операции перетаскивания определены.

Для каждого объекта может быть определена возможность начала перетаскивания (из данного объекта можно производить перетаскивание) и возможность перетаскивания (разрешено перетаскивание в данный объект). Указание данных свойств производится при конфигурировании и может быть изменено средствами встроенного языка.

Процесс перетаскивания производится с помощью мыши и в основных приемах совпадает со стандартным, применяемым в Microsoft Windows.

Если при конфигурировании не производилось переопределение обработки событий перетаскивания, то выполняется простое перетаскивание. При этом результат перетаскивания зависит от вида элементов управления и типа исходных данных.

Возможность простого (без обработки) перетаскивания представлена в таблице.

Приемник Источник	Поле календаря	Поле табличного документа	Таблица	Поле картинки
Поле календаря	Нет	Да	Нет	Нет
Поле табличного документа	Нет	Да	Нет	Нет

Таблица	Нет	Да	Да	Нет
Поле картинки	Нет	Нет	Нет	Нет

Нет – указывает на невозможность простого перетаскивания.

Да – простое перетаскивание возможно.

При перетаскивании в поле табличного документа происходит преобразование перетаскиваемого значения в текстовое представление. При этом данные, для которых определено основное представление (например, для списка в виде кода или наименования), перетаскиваются как значение основного представления. Так, данные из поля календаря имеют тип значения *Дата* и преобразуются в строку в соответствии с текущими региональными установками. Если в таблице отображаются данные таблицы значений, то в поле табличного документа будет помещен индекс строки таблицы значений.

При перетаскивании данных внутри таблицы следует иметь в виду, что в динамических списках не может быть изменен порядок следования строк. Для иерархических данных может быть выполнен только перенос данных из одной группы в другую.

В таблице, отображающей статические данные (табличные части, таблицы значений, списки значений, дерево значений), допускается изменение порядка следования строк.

При этом действуют следующие правила перетаскивания:

- если перемещается строка снизу на строку, расположенную над перемещаемой, то она вставляется перед строкой, на которую перемещается;
- если перемещается строка вверх на строку, расположенную ниже, то она вставляется после строки, на которую перемещается;
- если перемещается несколько строк, то действия для них аналогичны действиям при перемещении одной строки, за исключением ситуации, когда строки выбраны не последовательно, а через промежуток. В этом случае выделенные строки группируются (располагаются рядом друг с другом). Позиция выделенной группы в любом случае выбирается исходя из пунктов 1 или 2;
- чтобы переместить строку перед последней, нужно переместить ее перед предпоследней, если строка перемещается сверху.

Если при конфигурировании разработчиком предусмотрена обработка событий по перетаскиванию, то указание такой возможности следует искать в описании работы с конфигурацией.

Помимо операций перетаскивания между элементами управления «1С:Предприятие» позволяет осуществлять перетаскивание данных из других приложений в элементы управления. Также можно перетащить мышью текст и список файлов.

5.19. Работа формы с сохранением настроек

Когда на форме расположено много реквизитов, то в конфигурации может быть предусмотрена возможность сохранять наборы установленных реквизитов и использовать их при необходимости. Для этого на форме располагаются команды [Сохранить параметры](#) и [Восстановить параметры](#).

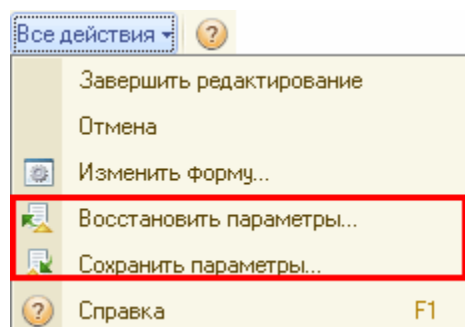
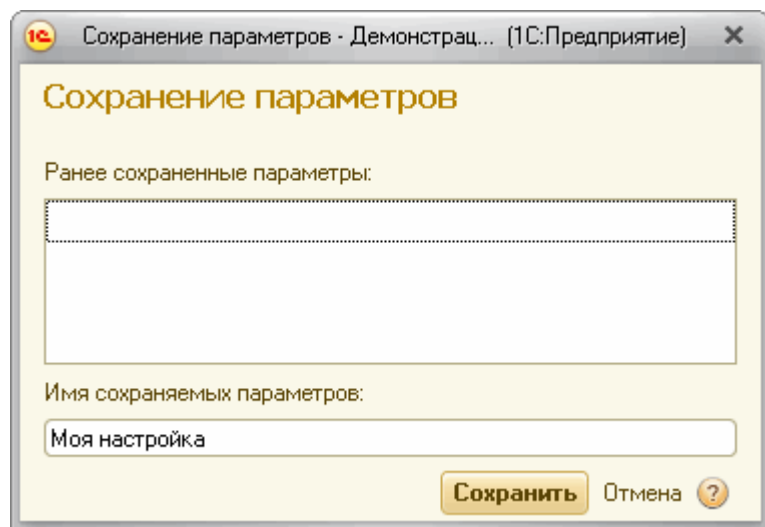
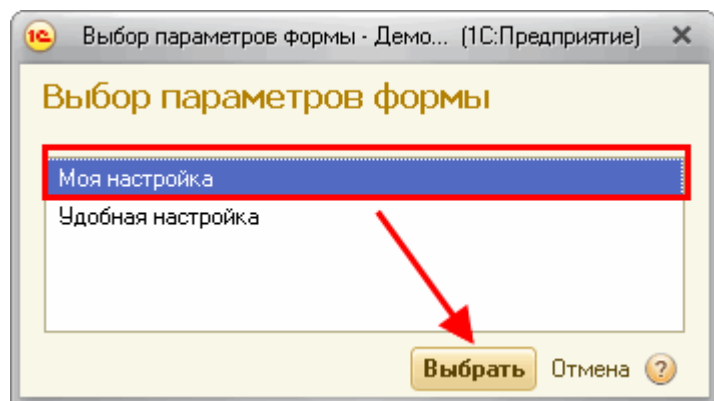


Рис. 72. Команды для работы с параметрами формы

Следует установить нужные значения в полях формы и выбрать пункт [Сохранить параметры...](#) В появившемся окне [Сохранение параметров <имя объекта>](#) ввести имя для набора сохраняемых параметров и нажать кнопку [Сохранить](#).

**Рис. 73. Сохранение параметров формы**

Для использования сохраненных ранее параметров формы служит пункт [Восстановить параметры](#). В появившемся окне нужно выбрать набор параметров и нажать кнопку [Выбрать](#).

**Рис. 74. Выбор сохраненных параметров формы**

Параметры из выбранного набора будут использоваться в качестве параметров формы.

5.20. Особенности некоторых форм

При работе в большинстве форм существует возможность обращаться к любым функциям программы (через главное меню, клавиши, команды панелей навигации и действия, команды командных панелей), а также переключаться в другие открытые окна. Однако некоторые формы не допускают вызова других функций и переключения на открытые окна. Например, основное окно программы недоступно до тех пор, пока эта форма открыта. При работе с такой формой следует завершить все необходимые действия в этой форме и закрыть ее, прежде чем обращаться к другим режимам. Форму можно также закрыть нажатием клавиши [Esc](#).

Кроме обычных форм конфигурация может оперировать с табличными и текстовыми документами.

Глава 6. Списки

В системе «1С:Предприятие» для отображения информации чаще всего используются различные **списки**.

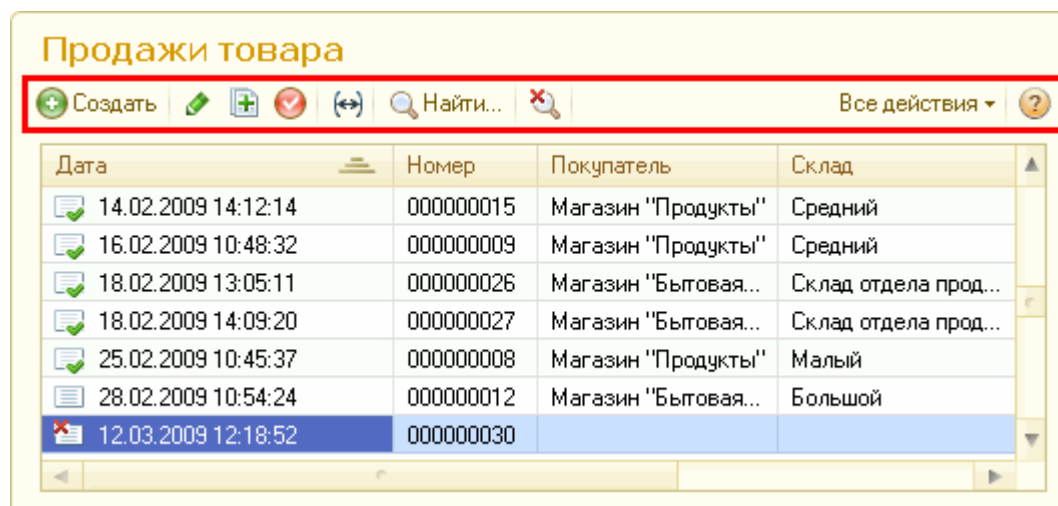


Рис. 75. Список документов «Продажи товара» и командная панель

Если это не запрещено в конкретной конфигурации (для конкретного пользователя), список может редактироваться в процессе работы – могут вводиться новые записи, редактироваться или удаляться существующие.

Работа со списками может осуществляться в двух формах: **форма списка** и **форма выбора элемента**.

Чтобы перейти к форме списка объектов, обычно следует выбрать соответствующую ссылку на панели навигации.

В форме выбора список открывается только при вводе значения реквизита формы нажатием кнопки выбора или клавиши **F4**. При этом в конфигурации могут быть определены различные формы для выбора элементов и групп.

Обычно в форме выбора можно установить курсор на требуемый элемент списка и по нажатию клавиши **Enter** выбрать текущий элемент как значение редактируемого реквизита, а также создать новый объект. В форме списка по нажатию клавиши **Enter** для текущего элемента открывается форма редактирования. Поведение программы для отработки нажатия клавиши **Enter** может быть изменено на этапе конфигурирования.

6.1. Просмотр списка

Для просмотра списка используются общие приемы работы с таблицами. Подробно эти приемы описаны в разделе «[Таблица](#)» [здесь](#).

Товары

Создать [иконки] Найти... Печать Все действия ?

Наименование	Код	Артикул	Поставщик
Вeko345MO	000000028	VEKO00001	ЭлектроБыт ЗАО
Хлеб	000000029	H987	Мосхлеб ОАО
Колбаса	000000030	Ko167	Животноводство ООО
Босоножки	000000031	Bos0009	Пантера АО
Торт	000000032	T78	Мосхлеб ОАО
ВekoNT02	000000033	NT02	ЭлектроБыт ЗАО
Вeko876N	000000034	876N	ЭлектроБыт ЗАО
Йогурт	000000035	S564	
Услуги	000000036		
Доставка	000000037		
Ремонт	000000038		

Рис. 76. Форма списка «Товары»

Форма списка в общем случае представляет собой таблицу, в которой выводится список элементов. Набор колонок таблицы определяется конкретной конфигурацией.

Состав и взаимное расположение колонок, отображаемых в таблице, могут различаться. То есть конфигурация может предлагать несколько визуальных представлений одного и того же списка.

ПРИМЕЧАНИЕ. В ячейке таблицы текст, содержащий более 20 000 символов, будет отображен не полностью.

При нажатии левой кнопки мыши на колонке списка система отсортирует информацию в списке по значению колонки.

Пользователь может настроить состав отображаемых колонок с помощью пункта меню [Все действия – Изменить форму...](#) Методика настройки форм приведена в разделе [«Настройка формы»](#) [здесь](#).

Если список содержит много элементов (строк) и колонок, то в форме отображается только определенная его часть, а внизу и справа появляются линейки прокрутки.

Как элементы, так и группы списка могут быть помечены к удалению, что может отображаться зачеркнутыми пиктограммами в крайней левой колонке списка.

Изменение ширины колонок в списках отслеживается. Размер сохраняется и восстанавливается при следующем открытии окна. Если список открывается в отдельном окне, то ширина колонок сохраняется вместе с размерами и положением окна.

6.1.1. Иерархические списки

Платформа «1С:Предприятие» предоставляет возможность работы с иерархическими списками с неограниченным числом уровней вложенности (ограничение может задаваться на этапе конфигурирования).

Иерархические списки состоят из элементов различных уровней, при этом элементы нижних уровней подчинены элементам верхних уровней. Существует два вида иерархии списков: иерархия групп и элементов и иерархия элементов. Вид устанавливается конфигурацией.

В списке с иерархией групп и элементов содержатся два вида элементов – группы и собственно элементы. Группа обозначает узел, в который входят другие (подчиненные) группы и элементы, а элемент является конкретным объектом.

Для списков с иерархией элементов любой из элементов может быть как узлом, так и отдельным объектом. Примером может служить список подразделений. Каждое подразделение может содержать в

своем составе другие подразделения, но набор свойств у всех подразделений будет одинаков.

Иерархический список имеет следующие режимы отображения: иерархический список, список, дерево. Режимы переключаются с помощью пункта меню [Все действия – Режим просмотра](#).

Выбранный режим просмотра отображается «нажатой пиктограммой» (такая пиктограмма имеет рамку напротив пункта меню).

Режим «Список». Если в качестве режима просмотра выбран [Список](#), то в форме показываются все элементы списка. При этом группы и элементы показываются неупорядоченно.

Наименование	Код	Вид цен
Магазин "Обувь"	000000015	Розничная
Магазин "Продукты"	000000013	Оптовая
Мосхлеб ОАО	000000012	Оптовая
Пантера АО	000000006	Закупочная
Покупатели	000000002	
Попов Б.В. ИЧП	000000007	Мелкооптовая
Поставщики	000000001	
Скороход АО	000000005	Закупочная
Шлюзовая ООО	000000004	Оптовая

Рис. 77. Режим просмотра «Список»

Режим «Иерархический список». Для списков с иерархией групп и элементов при включенном режиме [Иерархический список](#) показываются только элементы верхнего уровня, а после перехода в некоторую группу – только элементы, подчиненные этой группе.

Наименование	Код
Поставщики	000000001
Покупатели	000000002

Наименование	Код
Поставщики	000000001
Скороход АО	000000005
Пантера АО	000000006
Животноводство ООО	000000009
Корнет ЗАО	000000010
Элетробыт ЗАО	000000011
Мосхлеб ОАО	000000012

Рис. 78. Список контрагентов в режиме «Иерархический список»

- **Группа скрыта** – строка соответствует группе элементов. Нажатием левой кнопки мыши на этом значке можно раскрыть группу и перейти к работе с элементами, входящими в группу;
- **Группа открыта** – строка содержит название группы, открытой для просмотра. Такая строка всегда помещается в верхней части таблицы, под названиями колонок списка. Нажатием левой кнопки мыши на пиктограмме в строке можно закрыть группу;
- **Элемент списка** – строка является элементом списка.

Раскрыть группу в списке можно одним из следующих способов:

- дважды щелкнуть мышью на строке списка с наименованием нужной группы;
- нажать левую кнопку мыши на символе **[+]** в строке с наименованием нужной группы;

- установить курсор на строке с наименованием нужной группы и нажать клавиши **Ctrl + Стрелка вниз**;
- установить курсор на строку с наименованием группы и нажать кнопку **+** на цифровой клавиатуре;
- установить курсор на строку с наименованием нужной группы и выбрать пункт **Все действия – Перейти на уровень ниже**.

Для возврата на предыдущий уровень можно нажать клавиши **Ctrl + Стрелка вверх**, находясь в любой строке группы, или выбрать пункт **Все действия – Перейти на уровень выше**.

Режим «Дерево». Если в качестве режима просмотра выбран режим **Дерево**, то элементы списка отображаются в виде дерева.

Код	Наименование
[-] Товары	Товары
[+] 000000001	Обувь
[+] 000000011	Продукты
[-] 000000012	Электротовары
[+] 000000013	Чайники
[-] 000000014	Пылесосы
[-] 000000015	Вихрь
[-] 000000016	Тайфун-12
[-] 000000028	Veko345MO
[-] 000000034	Veko876N
[+] 000000039	Телевизоры
[+] 000000036	Услуги

Рис. 79. Иерархический список в режиме «Дерево»

Для удобства работы дерево имеет возможность раскрываться и закрываться. Знак **+** (плюс) в узле ветви указывает, что ветвь можно раскрыть. При нажатии левой кнопки мыши на этом знаке ветвь откроет для просмотра следующий уровень, а знак **+** (плюс) изменится на **-** (минус). Свернуть ветвь дерева можно нажатием мыши на знаке **-** (минус).

Если дерево раскрыто, то для перехода к нужной группе используются клавиши **Стрелка Вверх** и **Стрелка Вниз**.

Для одновременного раскрытия и перехода на группу нижнего уровня следует использовать сочетание клавиш **Ctrl + Стрелка Вниз**. Для перехода к родительской группе – **Ctrl + Стрелка Вверх**.

Чтобы свернуть узел дерева и все подчиненные, используются клавиши **Shift + Alt + Num-**. Чтобы свернуть все узлы дерева, используются клавиши **Ctrl + Shift + Alt + Num-**. Чтобы развернуть узел дерева и все подчиненные, используются клавиши **Shift + Alt + Num+**. Чтобы развернуть все узлы дерева, используются клавиши **Ctrl + Shift + Alt + Num+**.

Также при просмотре в любом из режимов «дерево» или «список» развернуть узел дерева можно нажатием клавиши **Num+**, а свернуть – нажатием клавиши **Num-**.

Для быстрого перехода в начало или конец списка можно использовать соответственно клавиши **Home** и **End**.

Для списков с иерархией элементов в режиме **Дерево** просмотр осуществляется так же, как и для списков с иерархией групп и элементов. При этом пиктограммы строк для списков с иерархией элементов не различаются.

Наименование	
⊖	Подразделения
⊕	Администрация
⊕	Бухгалтерия
⊕	Маркетинг и продажи
⊖	Хозяйственное обеспечение
⊕	Ремонтная группа
⊕	Сборочный сервис
⊕	Снабжение

Рис. 80. Список с иерархией элементов в режиме просмотра «Дерево»

6.2. Создание элемента списка

Для создания элемента следует нажать кнопку [Создать](#) формы списка или клавишу [Ins](#).

Если необходимо, чтобы элемент, создаваемый в иерархическом списке, сразу попал в определенную группу этого списка, то перед созданием следует перейти в эту группу (должен быть включен режим [Иерархический список](#) или [Дерево](#)).

ПРИМЕЧАНИЕ. Если для иерархического списка установлен режим просмотра [Список](#), то новый элемент будет создан в корневой группе.

Если в форме элемента предусмотрен реквизит [Группа](#), то независимо от режима и текущей группы для нового элемента можно указать принадлежность к любой другой группе.

Редактирование данных при вводе нового элемента списка осуществляется в выводимой на экран форме редактирования (окне) элемента.

Внешний вид окна, его свойства, поведение управляющих элементов определяются конфигурацией.

При редактировании реквизитов элемента списка используются общие приемы работы с формой. Подробно эти приемы описаны в разделе [«Общие принципы работы с формами»](#) [здесь](#).

При работе с формой создания элемента списка могут проверяться корректность и полнота введенных данных. В случае неправильного ввода могут появляться соответствующие сообщения. После заполнения реквизитов для записи элемента в список следует нажать кнопку [Записать и закрыть](#) или подобную ей кнопку, предназначенную для сохранения результатов редактирования, или закрыть форму кнопкой закрытия окна. После того как на экран будет выдан запрос о сохранении изменений, следует ответить [Да](#) для записи в список нового элемента или [Нет](#) – для отказа от записи.

Для отказа от ввода в список нового элемента следует нажать клавишу [Esc](#).

СОВЕТ. Для создания элемента списка используется команда группы [Создать](#) панели действий.

6.2.1. Создание группы списка

Для создания новой группы нужно выбрать пункт меню [Все действия – Создать группу](#) формы списка.

Если необходимо, чтобы новая группа, вводимая в список, сразу попала в определенную группу списка, то перед вводом следует перейти в эту группу (должен быть включен режим [Иерархический список](#) или [Дерево](#)).

Если в форме ввода группы предусмотрен реквизит [Группа](#), то для новой группы можно указать принадлежность к любой другой группе независимо от режима и текущей группы.

Также как и при создании элемента списка, при создании новой группы редактирование реквизитов группы осуществляется в выводимой на экран форме группы.

Система не проводит различий между кодами элементов списка и групп элементов, т. е. коды группы и элемента не могут совпадать.

В остальном приемы работы по созданию и редактированию группы списка совпадают с приемами работы по вводу элемента.

6.2.2. Копирование элемента списка

Новый элемент или группа могут быть добавлены в список копированием уже существующего элемента или группы. Для этого нужно поместить курсор на строку с элементом или группой, которые будут использоваться в качестве образца, и выбрать пункт [Все действия – Скопировать](#).

На экране отобразится форма создания нового элемента, в котором все реквизиты будут скопированы из реквизитов элемента-образца. Если для списка не задано автоматическое присвоение кодов, код нового элемента копируется из кода элемента-образца.

Корректировка реквизитов производится по общим правилам, изложенным далее.

6.3. Редактирование элемента списка

Для редактирования элемента необходимо установить курсор в любую ячейку строки, реквизиты которой требуется отредактировать, нажать клавишу [Enter](#), или дважды щелкнуть мышью в любом месте этой строки, или нажать клавишу [F2](#). На экран будет вызвана форма редактирования элемента списка, в котором следует отредактировать требуемые реквизиты.

Когда открыта форма элемента списка (или несколько форм), форма списка по-прежнему остается доступна для работы в окне приложения: можно вводить в нее новые элементы и группы, удалять их, переносить элементы из группы в группу, то есть использовать все возможности по работе со списками, предоставляемые программой.

Если в списке редактирование запрещено, то информация в открываемой форме будет недоступна для редактирования.

6.3.1. Реорганизация списка

При работе с иерархическими списками может возникнуть необходимость реорганизации элементов списка, то есть переноса существующих элементов или групп элементов из одной группы в другую.

Для переноса элемента или группы в другую группу списка нужно выполнить следующие действия:

1. В списке выбрать строку с тем элементом или группой, которую необходимо перенести в другое место списка.
2. Выбрать пункт [Все действия – Переместить в группу](#).
3. В открывшемся диалоге выбрать группу, куда предполагается перенести выбранный элемент.

СОВЕТ. Также элементы списка можно перемещать между группами с помощью мыши. При этом следует учитывать, что в иерархическом списке вручную перетащить элемент в родительскую группу нельзя.

Если в конфигурации задана поддержка уникальности кодов элементов списка, то при переносе код переносимого элемента не должен совпадать с кодами элементов, уже существующих в группе. Если коды совпадут, будет выдано сообщение: [Значение поля <Имя поля> не уникально](#). При этом необходимо будет исправить код у переносимого элемента.

6.3.2. Выбор значения из списка

Список может быть открыт для выбора элемента. В этом варианте список открывается нажатием кнопки выбора или клавиши [F4](#) при вводе реквизита из списка.

Когда список открыт для выбора значения, окно списка может иметь несколько другой вид, чем когда он просто открыт – иной состав и расположение колонок списка, отсутствие или иное положение дерева групп и так далее. В процессе настройки конфигурации для различных режимов работы со списком можно создать различные формы. Как правило, форма выбора содержит минимум необходимой информации, зачастую – только колонки наименования и кода.

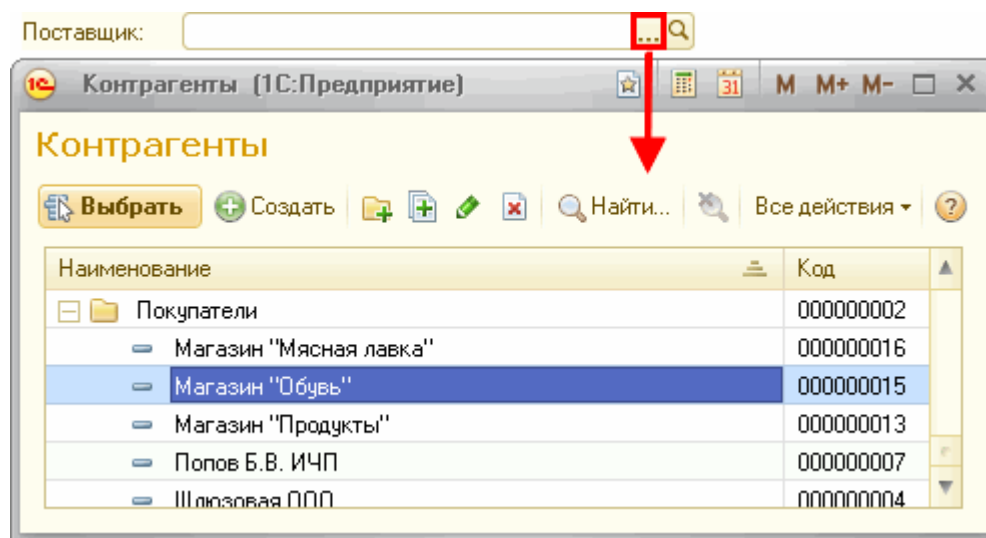


Рис. 81. Выбор значения реквизита из списка

В форме для выбора элемента также можно выполнять те же действия, что и в форме списка: добавлять в список новые элементы, редактировать дополнительные реквизиты существующих элементов списка, переносить элементы в другие группы (если в окне есть дерево групп) и тому подобное.

Например, если в списке организаций нужной организации нет, ее можно тут же ввести, а затем выбрать для использования в качестве реквизита документа. А для существующей организации можно исправить какие-либо реквизиты (если это доступно конкретному пользователю).

ПРИМЕЧАНИЕ. Следует учитывать принципиальное отличие поведения клавиши **Enter** в формах. В форме для выбора нажатие клавиши **Enter** приводит к выбору нужного элемента и закрытию (если иного не предусмотрено в конфигурации), а в формах списка – к открытию формы для просмотра и редактирования. Для редактирования элемента списка в форме для выбора используются команды меню **Все действия** или клавиша **F2**.

Информация в форме выбора синхронизирована с информацией формы списка, и все изменения, которые будут производиться в одном окне, будут отражаться в другом.

Для выбора элемента списка необходимо установить курсор в строку с нужным элементом и нажать клавишу **Enter** или кнопку **Выбрать** командной панели (если она предусмотрена в конфигурации).

Для иерархических списков в формах элементов можно указывать группу, к которой принадлежит данный элемент (если это предусмотрено в конфигурации).

Для некоторых реквизитов допускается выбор не только элементов, но и групп. Для быстрого выбора группы достаточно использовать стандартный прием выбора при нажатой клавише **Shift**.

СОВЕТ. На самом деле для выбора данных из списка необязательно пользоваться формой выбора. Достаточно ввести в поле ввода код или начало наименования необходимого элемента. Особенно удобно использовать код при вводе данных в тех случаях, когда коды постоянно используются и легко запоминаются (например, номер подразделения) или присутствуют в первичном документе (например, артикул или штрихкод товара).

6.3.2.1. Предопределенные элементы списка

Список может иметь предопределенные элементы (если это определено в конфигурации). Такие элементы введены на этапе конфигурирования и не могут быть удалены в режиме **1С:Предприятие**. Допускается только редактирование реквизитов таких элементов.

6.3.3. Поиск в списках

Если на форме присутствует поле строки поиска, то чтобы начать поиск, начните набирать текст. При этом необязательно переключаться в поле поиска. В списке отобразятся результаты поиска, совпадения будут выделены цветом. Перейти на строку поиска можно и с помощью сочетания клавиш **Ctrl+F**.

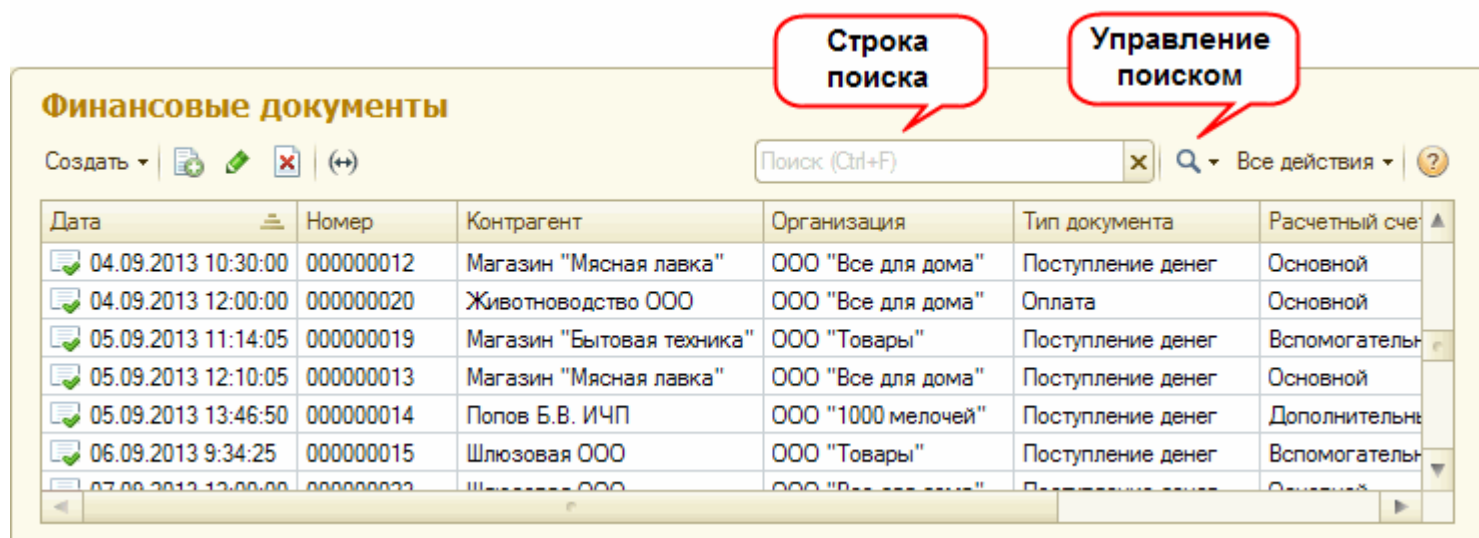


Рис. 82. Строка поиска на форме списка

Поиск будет выполнен по всем колонкам списка одновременно, по нескольким значениям. Например, как показано на рисунке ниже.

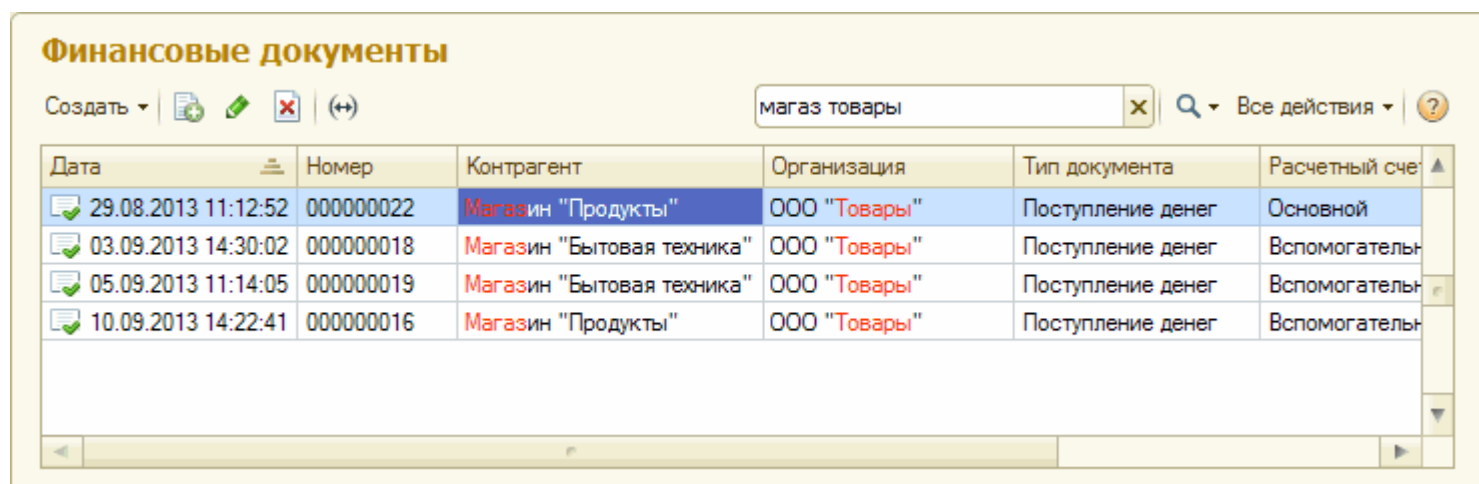


Рис. 83. Поиск в списке

При этом строка выражения поиска разбивается на части. В результат попадут строки списка, удовлетворяющие каждому условию:

- Строки ищутся по вхождению.

Если для списка включен полнотекстовый поиск, то поиск будет выполняться только по началу слов. Например, если нужно найти документы, относящиеся к магазину с названием **Мясная лавка**, достаточно будет ввести **маг мяс**.

Дата	Номер	Тип докуме...	Покупатель	Склад
02.08.2012 16:13:39	000000002	Заказ	Магазин "Мясная лавка"	Большой
02.08.2012 16:14:00	000000002	Продажа	Магазин "Мясная лавка"	Большой
04.09.2012 10:15:25	000000012	Заказ	Магазин "Мясная лавка"	Большой
04.09.2012 10:20:25	000000012	Продажа	Магазин "Мясная лавка"	Большой
05.09.2012 12:00:00	000000013	Заказ	Магазин "Мясная лавка"	Большой

Рис. 84. Поиск строки в списке

Если для списка не включен полнотекстовый поиск, то поиск будет выполняться в любом месте строки. И тогда, например, при поиске по строке **маг** будут найдены не только магазины, но и универмаги:

Дата	Номер	Тип докуме...	Покупатель	Склад
27.01.2012 16:37:08	000000018	Продажа	Универмаг "Столичный"	Большой
01.08.2012 12:12:00	000000001	Заказ	Магазин "Бытовая техника"	Большой
01.08.2012 12:15:00	000000001	Продажа	Магазин "Бытовая техника"	Большой
02.08.2012 16:13:39	000000002	Заказ	Магазин "Мясная лавка"	Большой
02.08.2012 16:14:00	000000002	Продажа	Магазин "Мясная лавка"	Большой

Рис. 85. Поиск в списке без полнотекстового поиска

При поиске объекта по номеру можно вводить номер как с ведущими нулями, так и без указания ведущих нулей. Например, если нужно найти **Заказ** с номером **000000016**, достаточно ввести **16**:

Дата	Номер	Покупатель	Склад
10.09.2012 14:21:40	000000016	Магазин "Продукты"	Малый

Рис. 86. Поиск документа Заказ № 000000016

При этом, если для списка используется полнотекстовый поиск и ведущие нули введены не полностью (например, **016**), ничего найдено не будет.

При поиске даты, ее следует вводить целиком (в формате ДД.ММ.ГГГГ). Поиск осуществляется за весь день. Например, найдем документы продаж на **02.08.2012**:

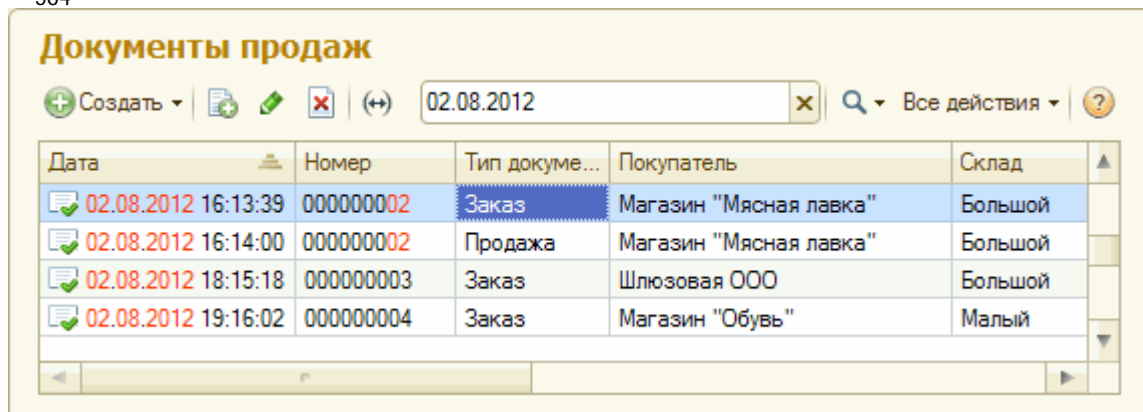


Рис. 87. Поиск даты в списке

Поиск чисел выполняется по точному равенству. Например, найдем документы на сумму **10500** рублей:

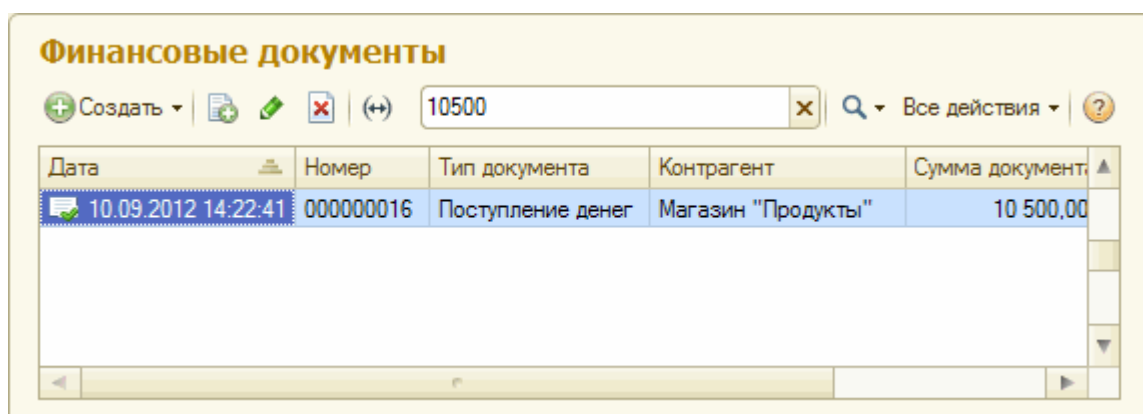


Рис. 88. Поиск числа в списке

Для значений полей флажка, при неиспользуемом полнотекстовом поиске, поиск выполняется по значению **Да** или **Нет**. Например, если нужно найти неиспользуемый склад, у которого флажок **Не использовать** установлен, в строке поиска введем **Да**. Если полнотекстовый поиск используется, то значения флажка в поиске не участвуют.

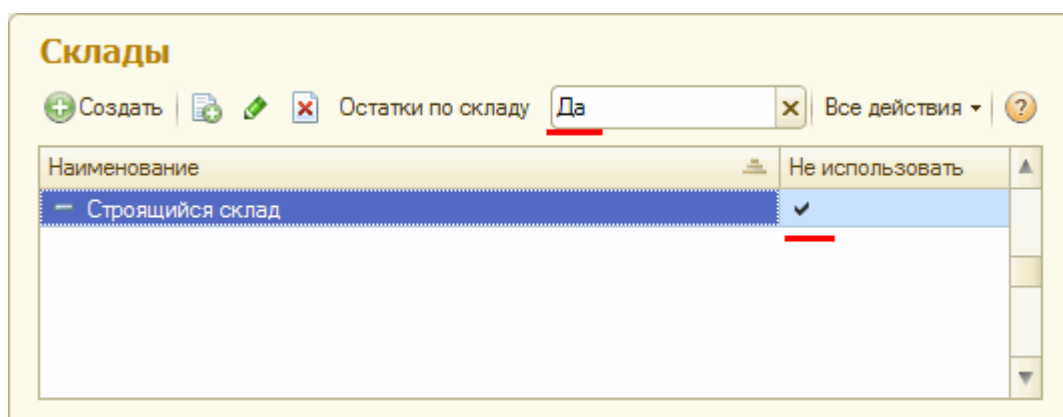


Рис. 89. Поиск по значению флажка

Для ссылок на объекты, поиск выполняется по тем полям, из которых состоит "значимая" часть ссылки (Например, в регистре перемещений товаров поиск будет выполняться по коду и/или наименованию товара. А если в списке движений документов нужно найти **Поступление товара 0000000009 от 25.06.2012** - поиск будет выполняться по номеру и дате документа - см. рисунок ниже).

Регистр товарных запасов

25.06.2012 9

Все действия

Период	Регистратор	Номер...	Товар
+ 25.06.2012 8:52:32	Поступление товара 000000009 от 25.06.2012 ...	1	Тапочки
+ 25.06.2012 8:52:32	Поступление товара 000000009 от 25.06.2012 ...	2	Сапоги
+ 25.06.2012 8:52:32	Поступление товара 000000009 от 25.06.2012 ...	3	Ботинки

Рис. 90. Поиск по ссылке

Найденные фрагменты строк в таблице выделяются.

Чтобы отменить поиск, следует нажать кнопку **Очистить** в поле строки поиска или нажать клавишу **Esc**.

Чтобы уточнить критерии поиска, используйте команду **Управление поиском - Расширенный поиск** или нажмите **Alt + F**.

Найти - Демонстрационная конфигурация "Управляемое прил... (1С:Предприятие)

Найти

Где искать:

Что искать:

Как искать: По началу строки По части строки По точному совпадению

Найти

Рис. 91. Форма поиска

В поле **Где искать** следует ввести или выбрать колонку из списка выбора, а в поле **Что искать** установить значение поиска или выбрать его из списка выбора.

Для строк можно использовать один из способов поиска: **По началу строки**, **По части строки**, **По точному совпадению**.

Для значения типа **Дата** – поиск по дате, включающее начало и конец выбранного дня, числовые и логические значения ищутся по точному соответствию (равенству).

Если в динамическом списке отображаются иерархические данные (справочника, плана счетов и т. д.), можно искать данные с учетом иерархии – во всем списке или в текущей группе.

Для поиска нажмите кнопку **Найти**. Поиск работает как фильтр – список показывает только те строки данных, которые соответствуют указанным условиям поиска (колонка + значение).

При поиске в иерархическом списке показываются элементы, которые соответствуют указанному поисковому выражению в виде простого (не иерархического) списка. Отмена поиска возвращает список обратно в иерархический режим просмотра.

Если список иерархический, то в диалоге отображается наименование текущей группы и флажки:

- **Искать только в текущей группе**,
- **Исключить подчиненные группы** - флажок доступен только если флажок **Искать только в текущей группе** установлен.

Если ни одной строки не найдено, то список будет пуст.

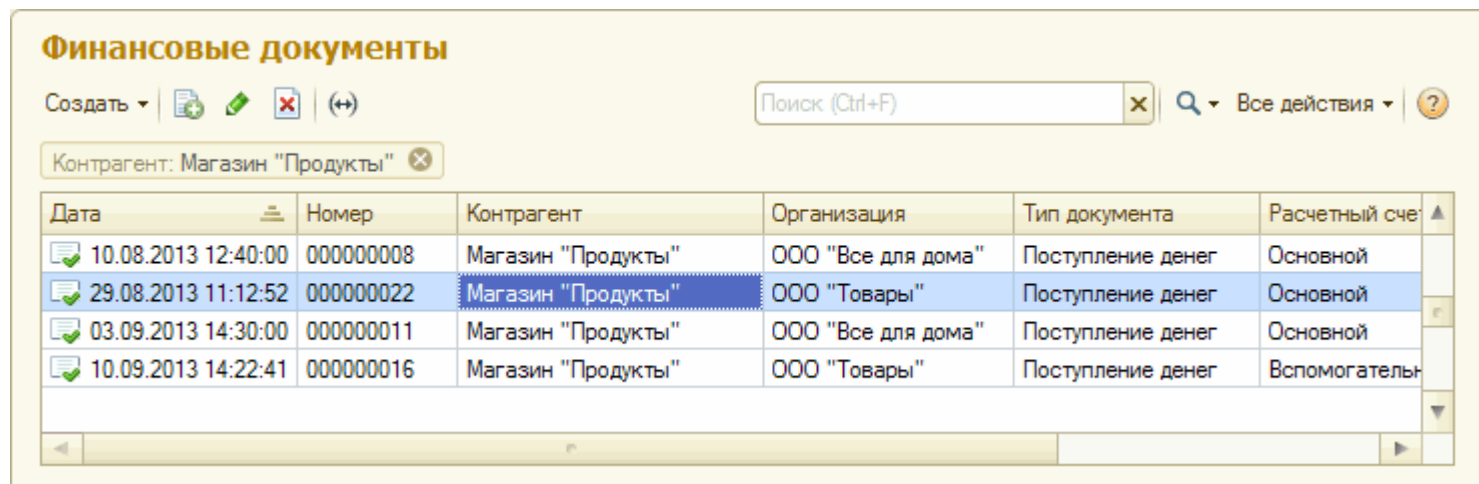


Рис. 92. Расширенный поиск

В списке найденных объектов можно уточнить параметры поиска.

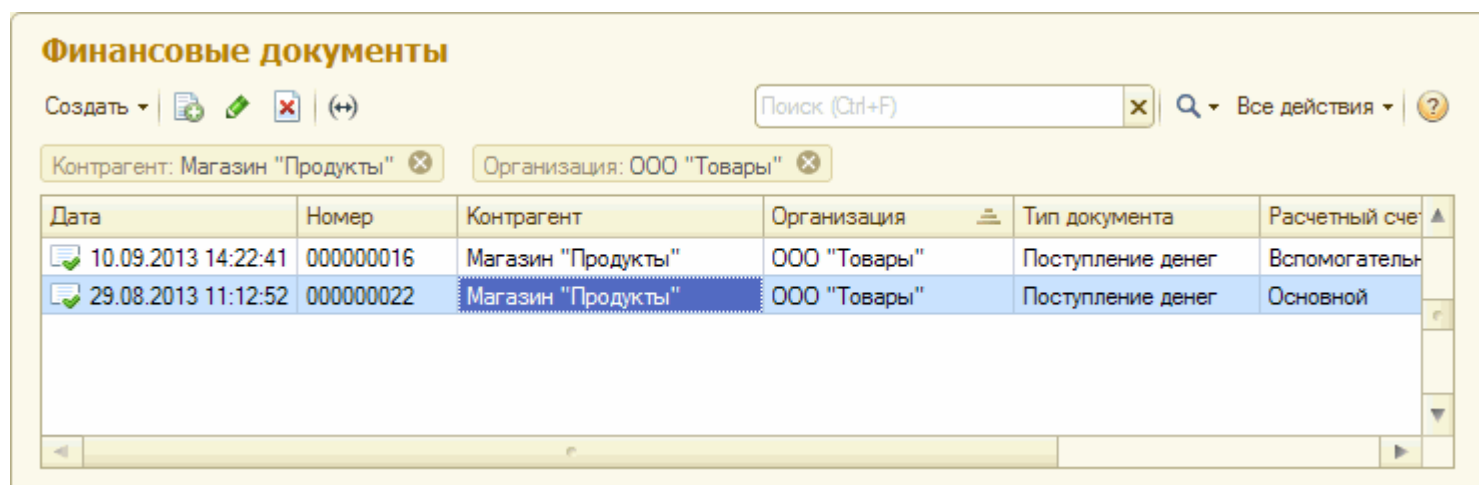


Рис. 93. Два поисковых выражения

В подменю [Управление поиском](#) сохраняются введенные ранее значения поиска. Значения поиска сохраняются между сеансами отдельно для каждой колонки. В списке ранее введенных поисковых выражений хранится не более пяти последних значений для каждой колонки.

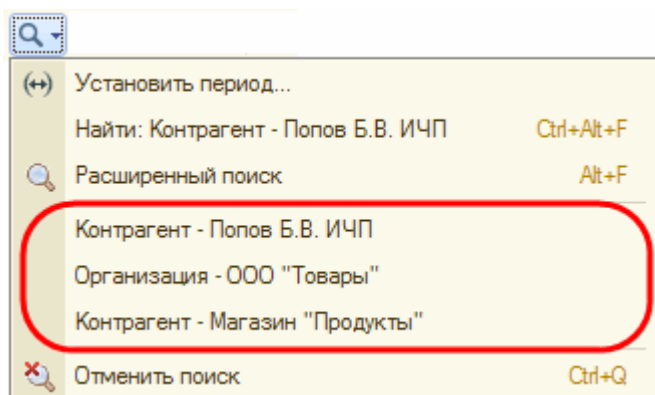


Рис. 94. История поиска

Чтобы изменить значение поиска, нужно нажать ссылку [<Где искать>](#): [<Что искать>](#) в элементе поиска.

Чтобы отменить поиск, следует нажать кнопку [X](#).

6.4. Настройка списка

Настройка порядка следования колонок в списке, их видимости, поведения списка при открытии и управления обновлением данными производится в диалоге [Настройка формы](#). Диалог открывается командой [Все действия – Изменить форму](#). Подробнее о настройке формы см. в разделе «[Настройка формы](#)» [здесь](#).

Настройка отображения данных, сортировки строк списка, а также группировки списка в таблице и оформления списка производится в диалоге [Настройка списка](#). Диалог открывается командой [Все действия – Настроить список](#).

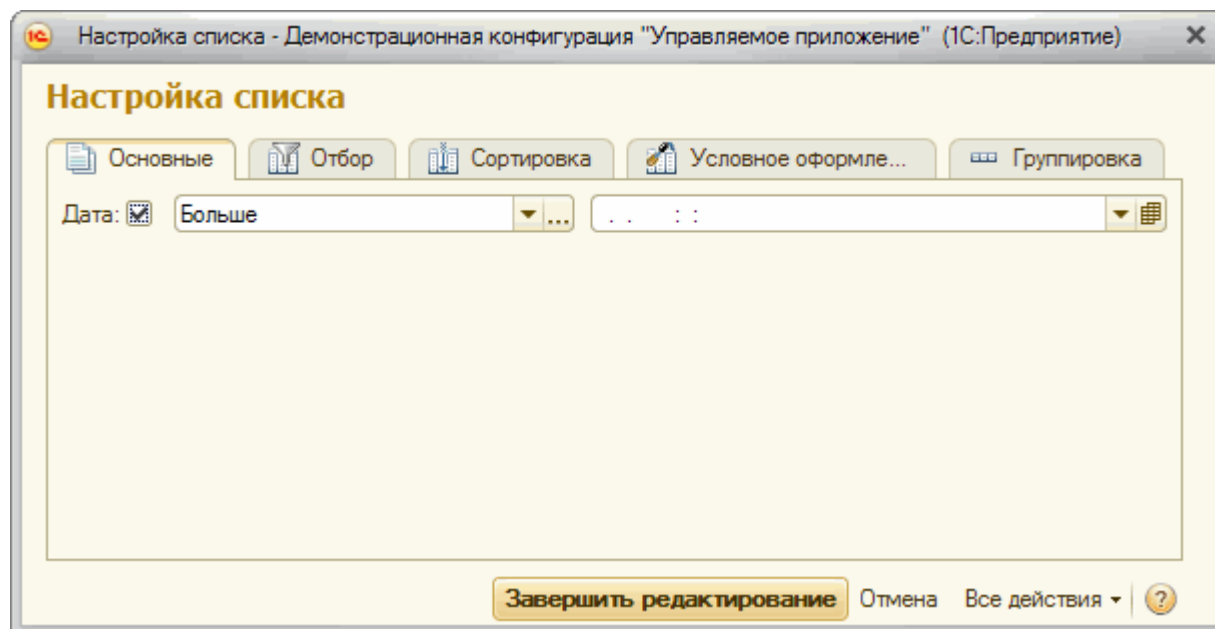


Рис. 95. Форма настройки списка

В диалоге предоставляется возможность настроить:

- отбор данных в списке;
- поля, по которым необходимо проводить упорядочивание;
- условное оформление списка;
- поля, по которым необходимо группировать данные;
- другие настройки, возможность изменения которых установлена разработчиком.

Все настройки распределены на закладках: [Основные](#), [Отбор](#), [Сортировка](#), [Условное оформление](#) и [Группировка](#). Чтобы установить значение настройки, следует установить значение настройки на соответствующей закладке. На закладку [Основные](#) попадают настройки

Элемент настройки применяется к списку, если напротив него установлен флажок использования. Для элементов можно настраивать порядок применения в границах одной настройки.

Настройка отображения данных в списке производится аналогично настройке варианта отчета в системе компоновки данных.

Вернуть настройки списка по умолчанию можно командой [Все действия – Установить стандартные настройки](#).

Для того чтобы определить состав настроек для изменения в основной форме списка, нужно выбрать [Все действия – Изменить состав настроек](#) и в открывшемся диалоге добавить или удалить необходимые настройки.

Чтобы принять настройки, следует нажать кнопку [Завершить редактирование](#).

Установленные значения настроек сохраняются между сеансами работы.

Чтобы сохранить установленный набор настроек в форме списка, нужно выбрать [Все действия – Сохранить настройки...](#) В открывшемся диалоге ввести название настройки и нажать кнопку [Сохранить](#).

Для использования созданных ранее настроек списков нужно выбрать [Все действия – Выбрать настройки](#). В открывшемся диалоге выбрать требуемую настройку из списка ранее сохраненных настроек и нажать [Выбрать](#).

6.4.1. Отбор

Чтобы установить отбор в списке, следует в диалоге настройки списка перейти на закладку [Отбор](#) и добавить элементы отбора.

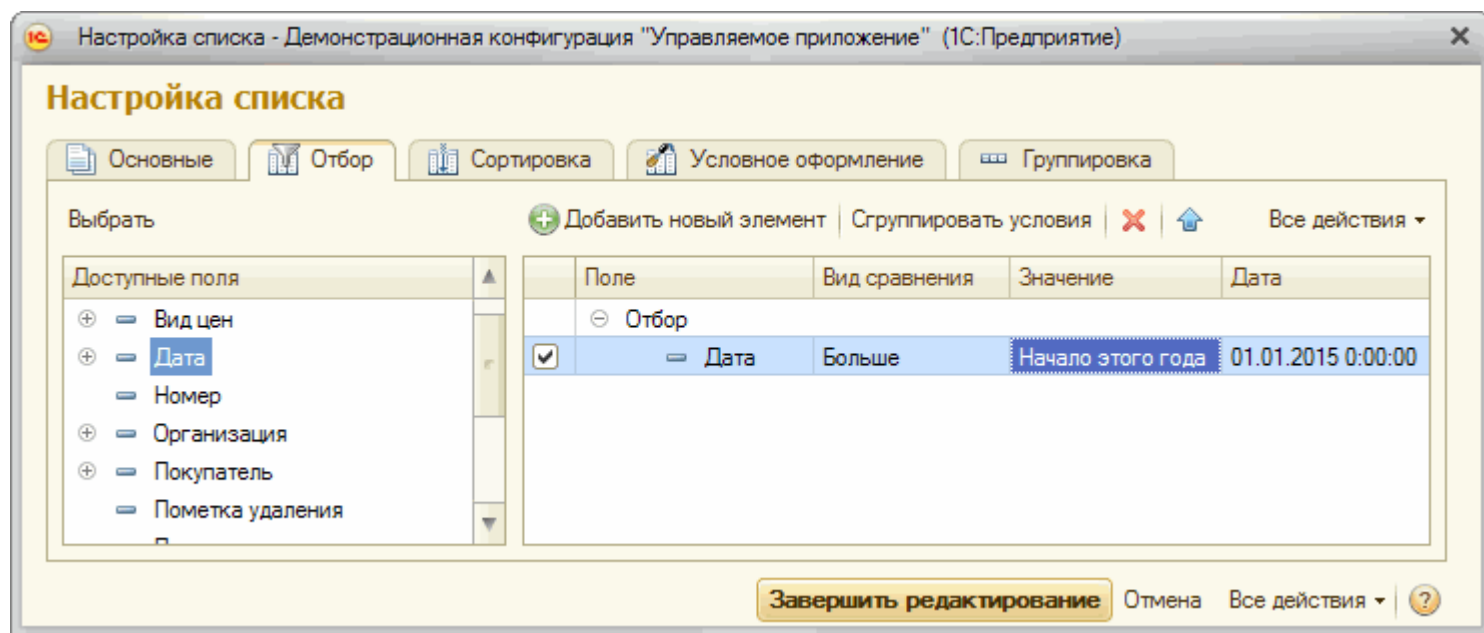


Рис. 96. Настройка отбора в списке

Для добавления элемента отбора следует нажать кнопку [Добавить новый элемент](#) или перетащить поле из списка [Доступные поля](#). Выбрать [Вид сравнения](#) в списке, всплывающем при нажатии кнопки [Выбрать](#) в колонке [Вид сравнения](#).

При необходимости можно изменить порядок, в котором условия будут применяться к списку, с помощью кнопок командной панели [Переместить вверх](#) и [Переместить вниз](#).

Чтобы применить настройку отбора в списке, следует нажать кнопку [ОК](#).

Подробнее о работе с отбором и о видах сравнения см. раздел «Отбор» [здесь](#).

6.4.2. Сортировка

Чтобы установить сортировку в списке, следует перейти на закладку [Сортировка](#) и выбрать значение сортировки или установить флажок использования напротив существующего элемента сортировки.

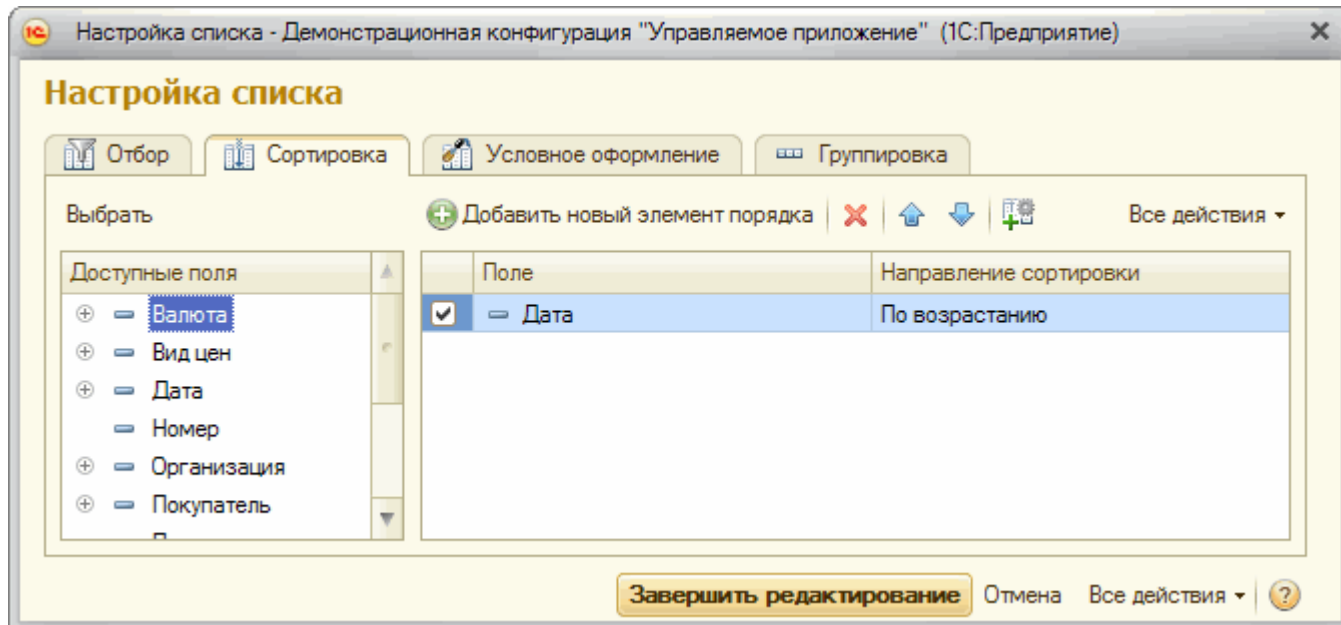


Рис. 97. Настройка сортировки списка

Для добавления элемента сортировки следует нажать [Добавить](#) или перетащить поле из списка [Доступные поля](#) и выбрать [Направление сортировки](#) ([По возрастанию](#)/[По убыванию](#)).

Если необходимо, установить порядок элементов сортировки кнопками [Переместить вверх](#) и [Переместить вниз](#).

Если полей сортировки несколько, то в списке данные будут отсортированы вначале по первому полю, потом по второму и т. д.

Чтобы применить настройку сортировки в списке, нажать кнопку [ОК](#).

Подробнее о работе с сортировкой см. раздел «Сортировка» [здесь](#).

ПРИМЕЧАНИЕ. Сортировка списка по полям, содержащим строки неограниченной длины, не разрешается.

6.4.3. Группировка

Чтобы сгруппировать записи в списке, следует в диалоге настройки перейти на закладку [Группировка](#) и на форме добавить поле, по которому необходимо провести группировку. Если было выбрано несколько полей, то записи в списке будут последовательно сгруппированы в порядке, установленном для полей группировки.

Чтобы применить настройку группировки, следует нажать кнопку [ОК](#).

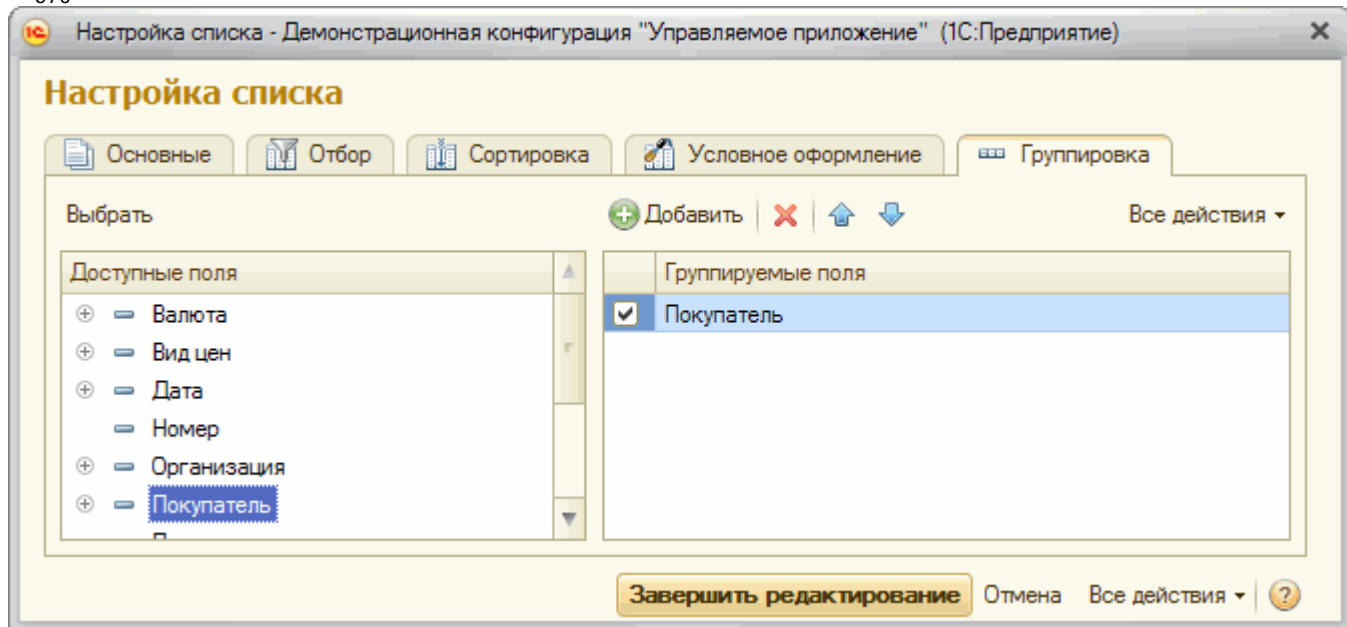


Рис. 98. Настройка группировки в списке

ПРИМЕЧАНИЕ. Группировка списка по полям, содержащим строки неограниченной длины, не разрешается.

6.4.4. Условное оформление

Для настройки условного оформления следует в диалоге настройки перейти на закладку [Условное оформление](#) и в табличном поле добавить необходимые элементы.

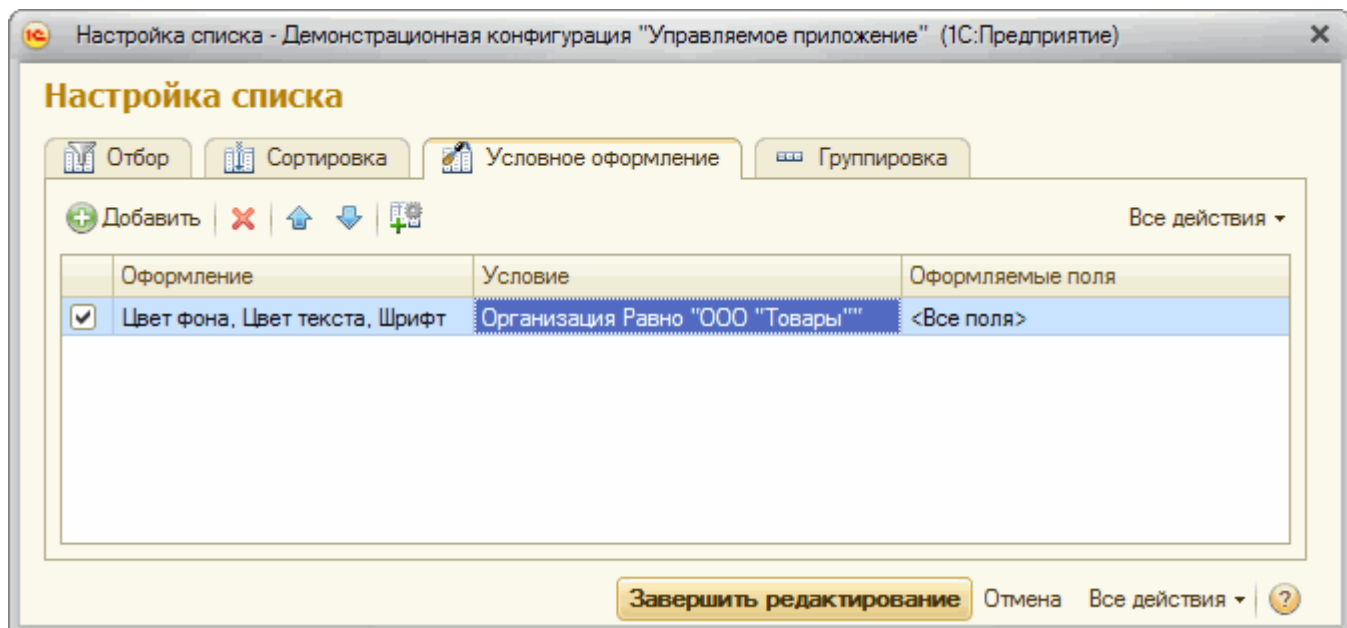


Рис. 99. Настройка условного оформления в списке

Чтобы создать элемент условного оформления, нужно нажать кнопку [Добавить](#). Редактирование элемента условного оформления выполняется в отдельном окне:

- на закладке [Оформление](#) выбрать требуемые параметры оформления: цвет фона, цвет текста и т. д.;
- на закладке [Условие](#) поставить условие, при выполнении которого оформление будет применяться к полям. Условие устанавливается аналогично отбору;
- на закладке [Оформляемые поля](#) выбрать поля из списка доступных полей, к которым будет применяться оформление, если условие выполнено.

Если необходимо, можно установить порядок используемых элементов оформления с помощью кнопок [Переместить вверх](#) и [Переместить вниз](#).

Порядок элементов оформления устанавливает последовательность, в которой оформление будет применяться к элементам отчета. Если по некоторым условиям к области отчета применяются два разных оформления, то в итоге применяется последнее из списка элементов условного оформления.

Подробнее о настройке условного оформления см. раздел «Условное оформление» [здесь](#).

Чтобы применить настройку условного оформления, следует нажать кнопку **ОК**.

6.4.4.1. Сохранение настроек между сеансами

Настройки динамического списка сохраняются автоматически при закрытии формы и автоматически загружаются при открытии формы в новом сеансе.

Режим просмотра списка ([Иерархический список](#) / [Дерево](#) / [Список](#)) сохраняется автоматически.

6.4.5. Настройка периода

Для некоторых списков существует возможность ограничить отображаемые данные по установленному периоду.

Интервал следует указать в диалоге [Настройка периода](#), вызываемом командой [Все действия – Установить период](#).

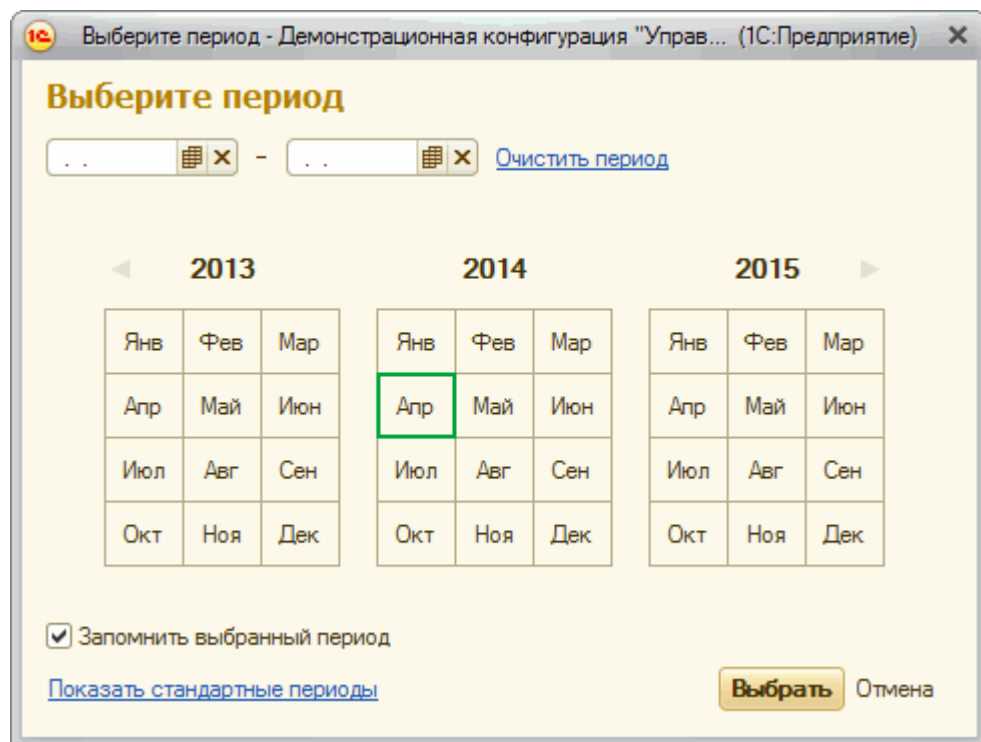


Рис. 100. Настройка периода в списке

В открывшейся форме доступны следующие действия:

- Установить период в поле выбора периода.
- Выбрать один из стандартных периодов.
- Также период можно указать произвольный период, выбрав даты начала и окончания периода в полях выбора дат.

Чтобы очистить выбор периода, следует нажать ссылку [Очистить период](#).

Для сохранения настроек периода при следующем открытии списка следует установить флажок [Запомнить выбранный период](#).

Чтобы завершить настройку периода, следует нажать кнопку [Выбрать](#). Также текущая позиция курсора устанавливается в качестве выбранного периода при нажатии клавиши [Пробел](#) или [Enter](#).

Для отмены настройки периода следует нажать кнопку [Отмена](#).

Выбранный период будет отображен в виде дополнительной кнопки на панели команд формы списка. Чтобы изменить период, нужно нажать на ссылку в новой кнопке.

Для того, чтобы сбросить фильтр по периоду, следует нажать на крестик этой кнопки.

6.4.5.1. Выбор произвольного периода

На форме период выбирается с помощью мыши одним из следующих способов:

- Нажать левую кнопку мыши на первом месяце периода. Удерживая кнопку мыши, переместить курсор до последнего месяца периода. Отпустить кнопку мыши. Период можно задавать и в обратном порядке (от последнего до первого месяца).
- Нажать левую кнопку мыши на первом месяце периода. После этого нажать левую кнопку мыши на последнем месяце периода. Период может быть задан и в обратном порядке.
- При двойном нажатии на месяце левой кнопки мыши этот месяц выбирается как период.

Добавление/удаление месяцев из периода

Чтобы добавить месяцы в диапазон, нажмите на требуемом месяце левую кнопку мыши, удерживая клавишу [Shift](#). При этом:

- Если выбран месяц после последнего месяца текущего выбранного периода, то в период добавляются месяцы от последнего до выбранного (включительно).
- Если выбран месяц ранее первого месяца текущего выбранного периода, то в период добавляются месяцы от выбранного до первого месяца текущего выбранного периода.
- Если выбран месяц внутри выбранного периода, изменений не происходит.
- Если период не был выбран, то период устанавливается так же, как и без удерживания клавиши [Shift](#).

Нажатие левой кнопки мыши вместе с клавишей [Ctrl](#) добавляет или удаляет месяцы в период:

- Если выбран месяц, следующий за последним месяцем выбранного периода или предшествующий первому месяцу выбранного периода, этот месяц добавляется в период.
- Если выбран первый или последний месяц в периоде, тот этот месяц из диапазона удаляется.
- Если период не был выбран, то период устанавливается так же, как и без удерживания клавиши [Ctrl](#).

6.4.5.2. Использование клавиш управления курсором

При нажатии на клавиатуре клавиш [Стрелка вверх](#), [Стрелка вниз](#), [Стрелка вправо](#), [Стрелка влево](#):

- Если текущий период - первая строка в году и нажата [Стрелка вверх](#), изменений не происходит.
- Если текущий период - последняя строка в году и нажата [Стрелка вниз](#), изменений не происходит.
- Иначе период устанавливается на месяц, который находится соответственно сверху, снизу, справа или слева от выбранной позиции.

При нажатии клавиш управления курсором одновременно с клавишей **Shift** текущий выбранный период не сбрасывается, а соответствующие месяцы добавляются в период.

6.4.5.3. Выбор стандартного периода

Чтобы перейти к выбору стандартного периода в форме настройки следует нажать ссылку [Показать стандартные периоды](#).

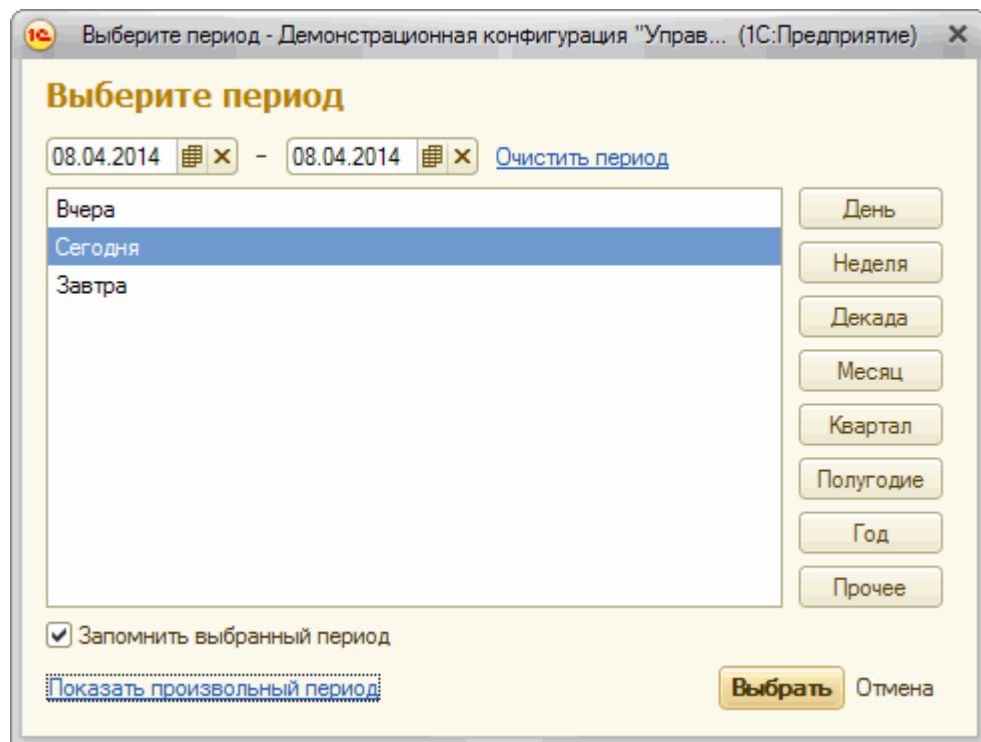


Рис. 101. Настройка стандартного периода в списке

В левой части формы отобразится список вариантов для каждого выбранного периода.

В правой части формы необходимо нажать кнопку с названием требуемого периода и в левой части выбрать вариант периода.

Чтобы вернуться к выбору произвольного периода, следует нажать ссылку [Показать произвольный период](#) или выбрать произвольную дату в полях выбора даты.

6.5. Печать списка

Для вывода списка в текстовый или табличный документ нужно выбрать пункт [Все действия – Вывести список](#).

В отобразившемся диалоге выбрать тип документа и требуемые колонки.

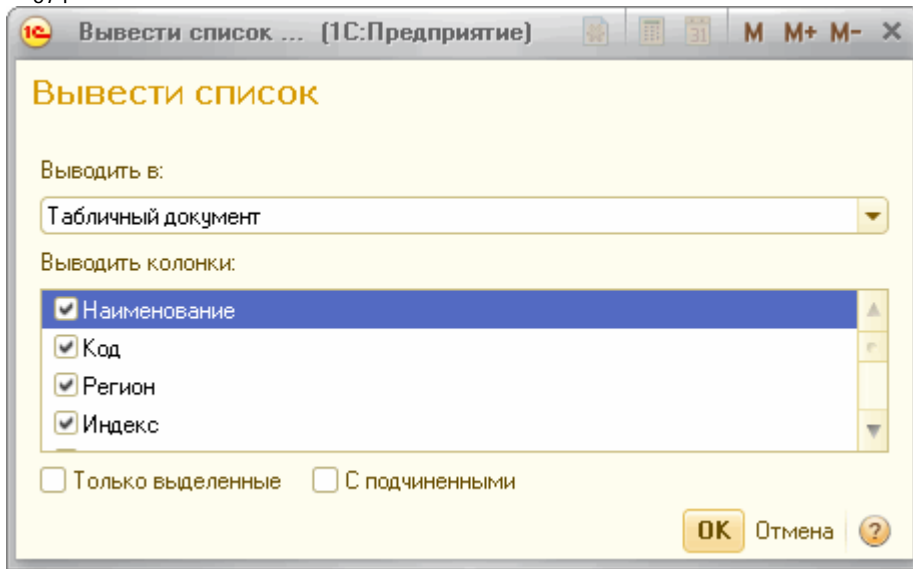


Рис. 102. Диалог настройки вывода списка

Если в списке предусмотрен режим множественного выделения строк, то в диалоге можно указать режим печати только выделенных строк (флажок **Только выделенные**).

Если в таблице выводится иерархический список, то в диалоге можно указать режим печати данных с включением подчиненных групп и элементов (флажок **С подчиненными**).

ПРИМЕЧАНИЕ: При выводе списка дополнительные настройки колонок (группировка и положение) не учитываются.

Глава 7. Работа с данными различных видов

Данная глава содержит описание функций для работы с такими данными, как документы, журнал документов, и информацию о специфических свойствах объектов информационной базы.

7.1. Нумерация элементов

Одним из обязательных реквизитов объекта конфигурации является его код (для документов – номер).

В зависимости от конфигурации при создании элемента его код (номер) может формироваться автоматически или вводиться вручную.

Если для конкретного объекта установлена автоматическая нумерация, то после сохранения элемента поле для ввода кода (номера) будет содержать код (номер), сформированный системой. Этот код (номер) можно исправить, однако программа проследит, чтобы указанный код (номер) не совпадал с кодами (номерами) объектов этого же вида, уже существующих в базе.

Если при сохранении элемента система обнаружит, что такой код (номер) уже использован, она выдаст соответствующее предупреждение.

Префикс. Для объектов конфигурации может быть задан префикс кода (номера). Таким образом, при создании элемента очередной код (номер) будет предложен не только исходя из правил автоматического присвоения кодов (номеров), но и с определенным префиксом. Тем не менее пользователь может полностью отредактировать предложенный код (номер), включая его префиксную часть.

7.2. Ввод элемента на основании

В системе «1С:Предприятие» существует механизм создания новых элементов на основании имеющихся данных.

Этот механизм позволяет, например, выставить счет на оплату, используя информацию из ранее созданной расходной накладной.

Порядок переноса конкретных реквизитов объекта-образца в объект-копию устанавливается при создании объектов в процессе разработки конфигурации.

Информацию об использовании возможности ввода объекта на основании другого объекта можно получить в описании конфигурации.

Для ввода объекта на основании другого объекта в форме списка следует выбрать объект-основание, на командной панели выбрать пункт [Создать на основании](#) и в выпадающем подменю выбрать вид создаваемого объекта.

После этого на экран будет выдана форма создания объекта выбранного вида, в котором реквизиты будут заполнены информацией из объекта-основания.

При создании документа ему присваивается очередной порядковый номер. В качестве даты документа будет установлена текущая дата.

7.3. Удаление (пометка удаления) элемента (группы)

Различают два режима удаления: непосредственное удаление и пометка удаления. Доступность режимов определяется правами конкретного пользователя. Если разрешено непосредственное удаление, то ответственность за нарушение ссылочной целостности информационной базы возлагается на администратора, разрешившего использование данного режима, и пользователя программы, выполнившего удаление элементов, на которые в информационной базе имеются ссылки.

Как правило, если это предполагается в конкретной конфигурации, система поддерживает ссылочную целостность объектов информационной базы: в этом режиме не проводится непосредственное удаление таких объектов системы, как, например, документы, – можно лишь установить пометку удаления объекта, но не удалить его. Собственно удаление производится при выполнении соответствующей сервисной процедуры – пункт [Поиск и удаление помеченных объектов](#) из списка [Все функции](#), если эта операция доступна конкретному конечному пользователю.

Чтобы установить пометку удаления элемента, нужно поместить курсор на строку с ним и выбрать пункт [Все действия – Пометить на удаление/Снять пометку удаления](#).

Для отмены пометки на удаление элемента списка поместить курсор на строку с ранее помеченным элементом и выбрать пункт [Все действия – Пометить на удаление/Снять пометку удаления](#). Это будет отражено в изменении условного значка в крайней левой колонке списка. Также команды установки и снятия пометки удаления могут быть доступны в форме объекта, в меню [Все действия](#).

ВНИМАНИЕ! Установка и снятие пометки удаления группы распространяются на все элементы, входящие в группу. Это же касается элементов подчиненного списка.

Подробнее о режимах удаления (непосредственное удаление, установка и снятие пометки удаления) см. в книге «1С:Предприятие 8.3. Руководство администратора».

7.4. Работа с данными из внешнего источника

Программа «1С:Предприятие» предоставляет возможность работать с данными из внешних баз. В зависимости от настроек программы данные из внешнего источника можно просматривать, использовать в отчетах, работать с ними через веб-клиента.

Общие параметры соединения с внешним источником настраивает администратор системы.

При любом обращении к данным внешнего источника, если к этому источнику подключение не было выполнено, будет выполнено подключение. Если подключение выполнено успешно, то выбранное действие продолжает выполняться. Если не были установлены параметры соединения, то при попытке подключения программа отобразит диалог для ввода этих параметров.

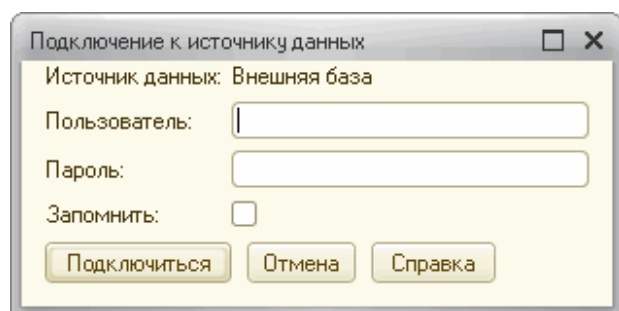


Рис. 103. Диалог подключения к внешнему источнику данных

При наличии соответствующих прав имя пользователя и пароль могут быть сохранены и использованы при последующих сеансах работы. Чтобы сохранить параметры, нужно установить флажок [Запомнить](#). После ввода данных программа выполнит подключение к внешнему источнику и отобразит сообщение [Соединение с внешним источником данных выполнено. Повторите действие](#). После этого можно продолжить работу с данными внешнего источника.

7.5. Печатная форма объекта

Если в процессе настройки конфигурации для объекта помимо экранной формы диалога была создана также печатная форма, то в диалоге редактирования объекта, как правило, присутствует кнопка, нажатие которой вызывает построение этой формы. Обычно такая кнопка имеет надпись [Печать](#), хотя может иметь и какую-либо другую подобную надпись.

После нажатия кнопки [Печать](#) (или выполнения каких-либо иных действий, если это указано в описании конфигурации) будет построена печатная форма.

Созданная печатная форма может открываться в отдельном окне стандартного табличного редактора системы «1С:Предприятие». Если в окне видна только часть формы, можно использовать клавиши управления курсором и линейки прокрутки для вывода на экран остальной части формы.

При необходимости автоматически построенная печатная форма может быть отредактирована. Чтобы иметь такую возможность, следует включить режим редактирования с помощью пункта главного меню [Таблица – Вид – Редактирование](#).

Отредактированная печатная форма может быть сохранена в файле на диске. Для этого следует выбрать

пункт **Файл – Сохранить** или **Файл – Сохранить как**.

Работа в табличном редакторе системы «1С:Предприятие» подробно описана в Приложении 3 настоящего Руководства, доступном в электронной версии документации.

Для непосредственного вывода печатной формы на принтер следует выбрать пункт **Файл – Печать** главного меню. На экран будет выдан стандартный диалог для настройки параметров печати. Подробнее см. раздел «Настройка печати» Приложения 3.

7.6. Работа нескольких пользователей

При работе нескольких пользователей в системе один и тот же объект могут пытаться отредактировать несколько человек. В этом случае только первому начавшему правку удастся завершить редактирование объекта. Всем остальным при попытке начать редактировать объект система выдаст предупреждение с номером сеанса и именем компьютера, с которого был заблокирован этот объект. Например, так:

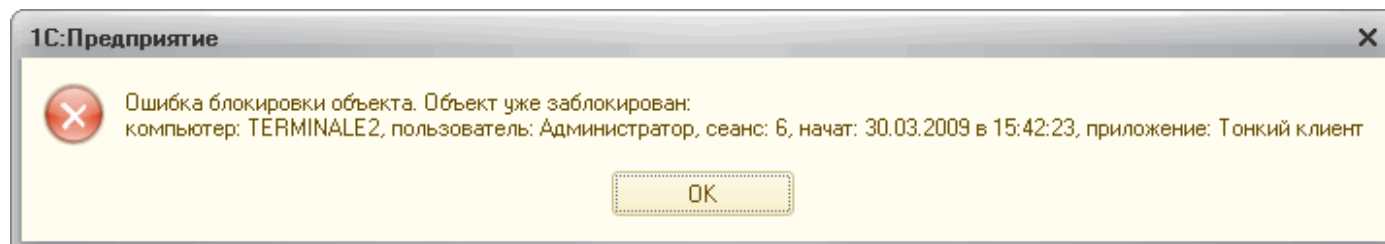


Рис. 104. Ошибка блокировки объекта

Чтобы внести свои изменения в освобожденный объект, его следует перечитать из базы данных.

7.7. Документы и журналы документов

В системе «1С:Предприятие» информация о хозяйственных операциях, совершаемых на предприятии, регистрируется при помощи документов. Документы могут отображаться в списках документов одного вида или журналах. Журнал представляет собой список документов нескольких типов.

В общем случае таблица журнала документов всегда включает в себя колонку **Тип документа**, которая содержит наименование документа.

Финансовые документы				
+ Создать + [Иконка] [Иконка] [Иконка] [Иконка] Найти... [Иконка] Все действия ▾ [Иконка]				
Отбирать: <input type="checkbox"/> по контрагенту: <input type="text"/> ... Q				
Дата	Номер	Тип документа	Контрагент	
05.11.2008 14:27:15	000000001	Поступление денег	Полов Б.В. ИЧП	
09.11.2008 12:20:14	000000007	Оплата	ЭлектроБьт ЗАО	
11.11.2008 9:31:27	000000002	Оплата	Скороход АО	
05.12.2008 12:19:50	000000006	Оплата	Пантера АО	
23.12.2008 12:24:06	000000006	Поступление денег	Магазин "Бытовая т...	
26.12.2008 14:12:40	000000009	Поступление денег	Магазин "Продукты"	
01.01.2009 13:15:08	000000010	Поступление денег	Магазин "Бытовая т...	
01.01.2009 14:17:34	000000012	Поступление денег	Магазин "Бытовая т...	
02.01.2009 12:23:20	000000004	Поступление денег	Магазин "Обувь"	
05.01.2009 13:22:36	000000008	Поступление денег	Магазин "Обувь"	
05.01.2009 14:38:19	000000013	Поступление денег	Магазин "Продукты"	
07.01.2009 12:19:03	000000004	Оплата	Корнет ЗАО	
09.01.2009 11:23:32	000000008	Оплата	ЭлектроБьт ЗАО	

Рис. 105. Журнал документов «Финансовые документы»

Как правило, левая графа журнала и списка документов – служебная. В ней различными пиктограммами

обозначается состояние документа:

Дата	Номер	Тип документа
02.03.2009 12:19:28	000000005	Оплата
03.03.2009 14:00:41	000000015	Оплата
31.03.2009 15:35:22	000000016	Оплата

Рис. 106. Состояния документа в журнале

- **Записан** – документ сохранен (записан), но не проведен;
- **Проведен** – документ проведен или для документа не предусмотрено проведение;
- **К удалению** – документ помечен на удаление.

7.7.1. Просмотр журнала документов

Для просмотра журналов документов используются общие приемы работы со списками. Подробно эти приемы описаны в разделе «[Просмотр списка](#)» [здесь](#).

Если в списке журнала документов выбран режим сортировки по дате, то при показе журнала записи выводятся в хронологическом порядке. При этом выполняются следующие правила:

- записи в журнале упорядочены по дате,
- внутри даты записи упорядочены по времени документа.

Если сортировка по дате отключена, то документы выводятся в порядке ввода.

7.7.2. Ввод нового документа из журнала документов

Если в журнале отображаются документы нескольких видов, то команда **Создать** представляет собой подменю, в котором можно выбрать вид создаваемого документа.

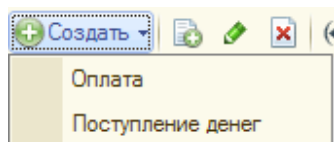


Рис. 107. Меню выбора вида документа

В этом меню будут выданы наименования только тех видов документов, которые отображаются в текущем журнале.

Если соответствующая команда предусмотрена в группе **Создать** панели действий, ввод документов можно проводить, не открывая журнала или списка документов.

В меню выбора вида документа необходимо указать наименование нужного вида документа. После этого на экран будет выдана форма нового документа для заполнения его реквизитов.

Если в журнале хранятся документы только одного вида, список видов документов выдаваться не будет, а сразу будет выдан диалог для заполнения реквизитов документа.

7.7.3. Дата и время документа

Практически любой документ имеет поля для ввода даты и номера документа.

Возможность указания времени документа является важной особенностью системы «1С:Предприятие». Документы выстраиваются в хронологическом порядке и зачастую обрабатываются именно в той хронологической последовательности, которую задают дата и время документа. Таким образом, время документа служит не столько для отражения астрономического времени его ввода в систему, сколько для четкого упорядочивания документов внутри даты.

Предложенную системой дату можно изменить. При записи документа ему, как правило, устанавливается текущее время.

7.7.4. Проведение документа

Проведением документа называется действие, которое отражает данные документа в тех или иных учетных механизмах на основании информации документа. Возможность проведения – свойство документа, определенное заранее. При проведении документа информация, содержащаяся в документе, учитывается в регистрах путем создания записи регистров. Записи регистров – это информация о том, как изменяется состояние регистра в результате проведения документа.

При нажатии кнопки формы документа, для которой определено проведение документа (обычно это кнопка [Провести и закрыть](#) или [Провести](#)), производится проведение документа и документ закрывается. В журнале документов проведенный документ будет помечен пиктограммой (см. [рис. 108](#)).

В некоторых случаях при проведении документа возникает ситуация, из-за которой проведение не может быть выполнено. Например, при проведении расходной накладной обнаружено, что на складе не числится нужного количества товара. При этом будет выдано сообщение о невозможности проведения документа. В этом случае документ не закроется автоматически. Можно внести в него исправления и заново попробовать провести.

Чтобы отменить проведение документа, следует выбрать пункт [Все действия – Отмена проведения](#). После подтверждения пиктограмма для документа изменится (см. [рис. 108](#)).

	Дата	Номер
Проведен	02.04.2009 10:46:33	000000031
	25.03.2009 15:26:46	000000030
Проведение отменено	28.02.2009 10:54:24	000000012
	18.02.2009 13:05:11	000000026

Рис. 108. Состояние проведения документа

При отмене проведения документа, если иного не задано в данной конфигурации, отменяются все действия, которые он выполнил в процессе проведения.

Непроведенный документ можно откорректировать, записать, провести (или не проводить), а пиктограмма в крайней левой колонке журнала будет правильно отражать текущее состояние документа.

7.7.5. Непроводимые документы

Документы, для которых при разработке конфигурации не предусмотрено проведение, не проводятся, но в списках документов показываются такой же пиктограммой, как у проведенного документа. Нельзя отменить проведение у непроводимых документов (см. [рис. 109](#)).

	Дата	Номер
Непроводимый документ	02.04.2009 10:38:22	000000002
Непроводимый документ помечен на удаление	02.04.2009 10:37:20	000000001

Рис. 109. Список непроводимых документов

7.7.6. Просмотр движений документов

Для анализа тех изменений, которые произвел в регистрах тот или иной проведенный документ, в конфигурации может быть предусмотрена возможность просмотра движения регистров. В описании конфигурации указано, каким образом выполняется просмотр движений регистров.

Форма просмотра движения регистров представляет собой список. В нем отображаются движения регистров. Состав колонок списка зависит от структуры выбранного регистра.

Для регистра товарных запасов форма просмотра движений может выглядеть следующим образом:

Регистр товарных запасов							
Найти...							Все действия ▾ ?
Период	Регистратор	Но...	Товар	Склад	Количество		
+ 02.03.2008 6:52:32	Поступление товара 0000...	1	Масло	Большой	1,00		
+ 28.03.2008 16:43:26	Корректировка остатка 0...	1	Bosch15	Большой	4,00		
+ 28.03.2008 16:43:26	Корректировка остатка 0...	2	Bosch1234	Большой	6,00		
- 03.11.2008 14:31:41	Продажа 000000001 от 03...	1	Туфли	Большой	2,00		
- 05.11.2008 13:47:09	Продажа 000000003 от 05...	1	Сапоги	Малый	2,00		
- 05.11.2008 14:25:53	Продажа 000000002 от 05...	1	Сапоги	Малый	2,00		
+ 11.11.2008 10:23:56	Поступление товара 0000...	1	Тапочки	Большой	2,00		
+ 15.11.2008 14:07:41	Поступление товара 0000...	1	Veko876N	Склад отдела продаж	100,00		
+ 15.11.2008 18:35:04	Поступление товара 0000...	1	Сапоги	Средний	1,00		
+ 15.11.2008 18:35:04	Поступление товара 0000...	2	Туфли	Средний	2,00		
+ 22.11.2008 12:44:06	Поступление товара 0000...	1	VekoNT02	Склад отдела продаж	200,00		
+ 23.11.2008 11:23:56	Поступление товара 0000...	1	Молоко	Малый	100,00		
+ 23.11.2008 14:03:57	Поступление товара 0000...	1	Торт	Средний	30,00		
- 23.11.2008 14:36:09	Продажа 000000028 от 23...	1	Йогурт	Малый	1 560,00		
+ 25.11.2008 13:16:59	Поступление товара 0000...	1	Босоножки	Большой	100,00		

Рис. 110. Регистр товарных запасов

Форма списка регистра всегда имеет две обязательные колонки.

Колонка **Номер строки** может содержать номер записи в наборе записей регистра, создаваемого при обработке документа.

В колонке **Период** отображается особая пиктограмма, которая служит для обозначения характера произведенных изменений в данном регистре. Знак + (плюс) обозначает прирост абсолютного значения измерений регистра, знак – (минус) – уменьшение. Именно знак + (плюс) позволил определить в приведенном выше примере, что количество товара увеличилось.

7.8. Бизнес-процессы

Бизнес-процессы в «1С:Предприятии» предназначены для объединения отдельных операций в цепочки взаимосвязанных действий, приводящих к достижению конкретной цели. Например, цепочку по выписке счета, приему наличной оплаты и отпуску товара со склада можно представить как бизнес-процесс **Продажа товара**.

Цепочки взаимосвязанных действий бизнес-процесса представляются с помощью **карты маршрута бизнес-процесса**. Карта маршрута описывает логику бизнес-процесса и весь его жизненный цикл от точки старта до точки завершения в виде схематического изображения последовательности прохождения взаимосвязанных точек маршрута.

Точка маршрута отражает этап жизненного цикла бизнес-процесса, связанный с выполнением, как правило, одной автоматической или ручной операции.

Задачи в «1С:Предприятии» позволяют вести учет заданий по исполнителям и служат отражением продвижения бизнес-процессов по точкам маршрута. При этом задачи могут создаваться не только бизнес-процессами, но и другими объектами информационной базы и непосредственно пользователями.

При переходе бизнес-процесса на какую-либо точку маршрута, предусматривающую действия конкретного исполнителя, формируется задача (или несколько задач в случае группового действия). После того как исполнитель отметит задачу как выполненную, бизнес-процесс автоматически переходит к следующей точке маршрута в соответствии с картой. Таким образом, задачи являются движущей силой бизнес-процессов.

Описание конкретных бизнес-процессов и их взаимосвязи с задачами приводится в описании конфигурации.

Общие принципы работы с бизнес-процессами и задачами аналогичны работе с другими объектами (например, документами и списками), поэтому в данном руководстве будут описаны только особенности.

7.8.1. Список бизнес-процессов

По умолчанию в списке бизнес-процессов отображается следующая информация:

- **Дата** – дата и время создания бизнес-процесса;
- **Номер** – номер (является уникальным для данного бизнес-процесса);
- **Стартован** – пометка старта бизнес-процесса;
- **Завершен** – пометка завершения бизнес-процесса, т. е. все задачи, порожденные бизнес-процессом, были выполнены;
- **Ведущая задача** – задача, на основе которой был создан данный бизнес-процесс.

Список бизнес-процессов **Продажа товара** может выглядеть следующим образом:

Дата	Номер	Стартован	Ведущая задача	Завершен
06.04.2009 0:00:00	3	✓		
06.04.2009 14:17:39	4	✓		✓
06.04.2009 14:20:15	5	✓		✓
06.04.2009 14:20:32	6	✓		✓
07.04.2009 0:00:00	8			
07.04.2009 14:20:43	7	✓		

Рис. 111. Список бизнес-процессов «Продажа товара»

С помощью пиктограмм обозначается состояние бизнес-процесса:

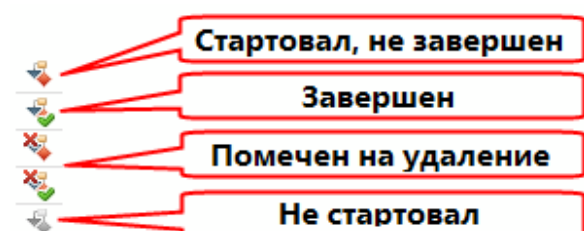


Рис. 112. Состояния бизнес-процесса

Если бизнес-процесс еще не стартовал, т. е. еще нет ни одной соответствующей ему задачи, не установлена пометка **Стартован**, то эта пиктограмма отображается черно-белой, а не цветной.

Завершен – бизнес-процесс считается завершенным, когда достигает точки завершения в соответствии с картой маршрута и все задачи по нему выполнены.

7.8.2. Список задач

По умолчанию в списке задач отображается следующая информация:

- **Номер** – номер (является уникальным для данной задачи);

- **Наименование** – наименование задачи;
- **Дата** – дата и время создания задачи;
- **Выполнена** – отметка о выполнении задачи;
- **Бизнес-процесс** – бизнес-процесс, породивший данную задачу;
- **Точка** – точка маршрута бизнес-процесса, в которой была создана данная задача.

Кроме этих данных могут быть определены дополнительные колонки для вывода значений любых других реквизитов бизнес-процесса или задачи.

Состояние задачи обозначается с помощью пиктограмм:

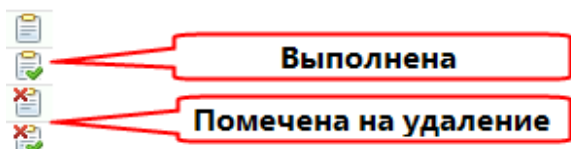


Рис. 113. Состояния задачи

7.8.3. Выполнение задачи

Выполнение задачи – важный этап в жизненном цикле бизнес–процесса. При выполнении задачи бизнес-процесс осуществляет переход на следующую точку маршрута в соответствии с картой маршрута, что приводит к формированию новых заданий в рамках данного бизнес-процесса.

Для того чтобы выполнить задачу, следует выбрать пункт **Все действия – Выполнено** для нужной задачи в форме задачи.

В системе может быть предусмотрено открытие дополнительных окон при выполнении задачи, например, для выдачи предупреждающих сообщений или ввода дополнительной информации. Также может быть предусмотрена проверка, которая запрещает выполнение задачи, если не соблюдены определенные условия (например, не согласован документ или не установлена скидка по счету).

Выполненные задачи помечаются специальной пиктограммой, и у них устанавливается флажок **Выполнена**.

Глава 8. Отчеты

Для получения данных из информационной базы в «1С:Предприятии» используются отчеты.

В режиме [1С:Предприятие](#) пользователь может сформировать отчет с предложенными системой настройками или установить свои настройки. Опытный пользователь может самостоятельно разработать свой собственный вариант отчета.

Внешний вид формы отчета по умолчанию представлен на рисунке ниже.

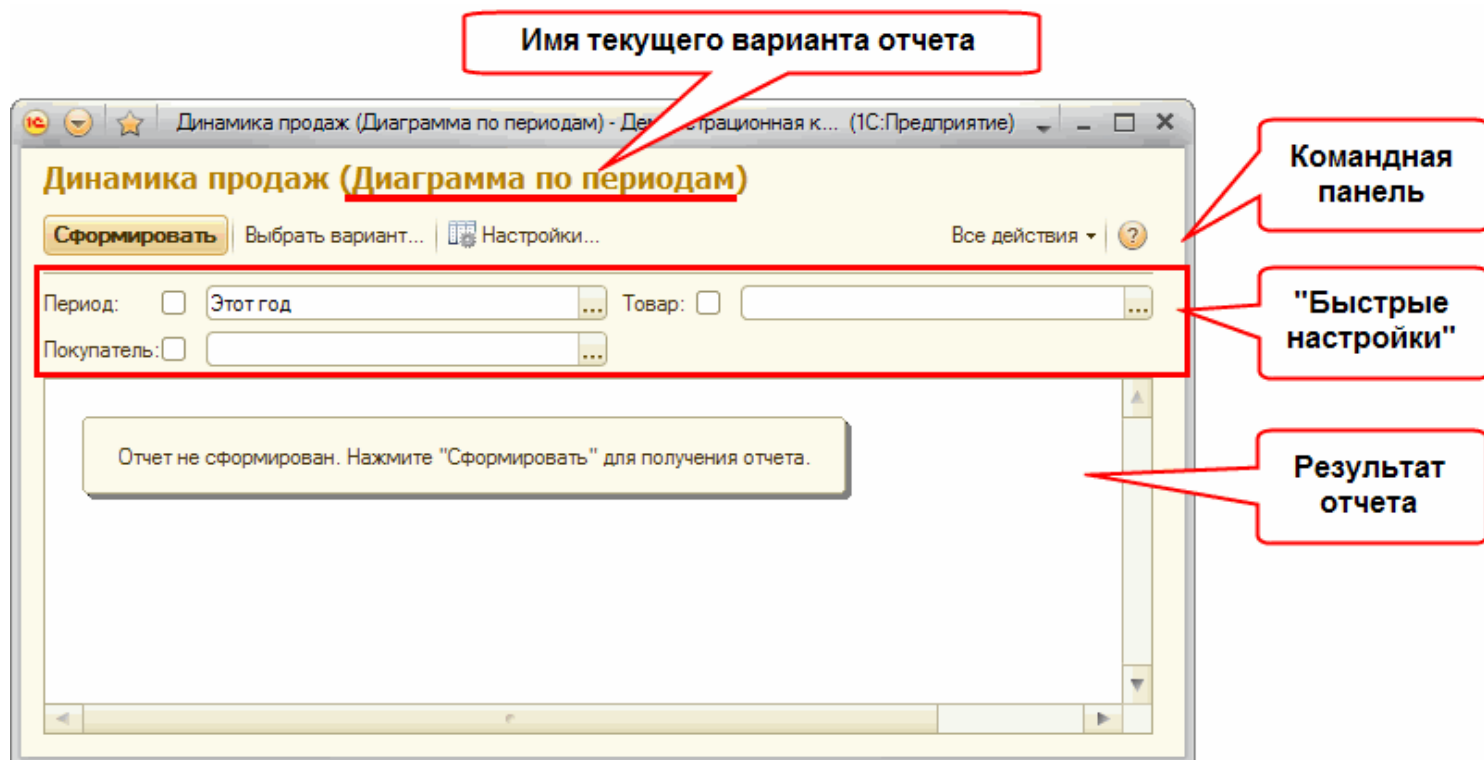


Рис. 114. Форма отчета

Чтобы сформировать отчет с предустановленными настройками, достаточно нажать кнопку [Сформировать](#) на командной панели отчета.

Если для отчета предусмотрено несколько вариантов, то перед выполнением следует выбрать требуемый вариант. Для этого на форме отчета используется кнопка [Выбрать вариант](#). Подробнее о работе с вариантами см. раздел [«Варианты отчета»](#) [здесь](#).

Для того чтобы установить настройки отчета, можно воспользоваться полем быстрых настроек или же специальным диалогом, открывающимся нажатием кнопки [Настройки...](#) Подробнее о настройках см. разделы [«Использование быстрых настроек»](#) и [«Настройки»](#) далее.

После того как результат отчета сформирован системой и выведен в табличный документ, можно использовать механизм расшифровки. Подробнее см. раздел [«Работа с расшифровкой отчета»](#) [здесь](#).

В зависимости от настройки отчета на форме отчета в подменю [Все действия](#) могут быть доступны следующие команды:

- [Настройки](#) – открывает форму настроек;
- [Выбрать настройки](#) – открывает форму для выбора ранее сохраненной настройки;
- [Сохранить настройки](#) – сохраняет измененные настройки отчета;
- [Установить стандартные настройки](#) – загружает настройки из варианта отчета;
- [Изменить вариант](#) – открывает окно настройки варианта отчета;
- [Выбрать вариант](#) – открывает форму для выбора варианта отчета;

- [Сохранить вариант](#) – открывает форму для сохранения варианта отчета;
- [Открыть новое окно](#) – открывает новую форму отчета с текущими настройками;
- [Изменить форму](#) – открывает форму настройки формы отчета.

Подробнее о работе с настройками см. раздел «Использование быстрых настроек» и главу «Изменение варианта отчета» далее.

Работа с расшифровкой отчета описана в разделе «[Работа с расшифровкой отчета](#)» [здесь](#).

ПРИМЕЧАНИЕ. На [рис. 114](#) и далее в примерах приведены формы, в которых системные команды скрыты.

8.1. Настройка отчета

Некоторые настройки отчета можно редактировать в отдельной форме. При выполнении отчета такие настройки используются вместе с основными настройками отчета.

Чтобы открыть форму редактирования настроек, следует нажать кнопку [Настройки...](#) на форме отчета.

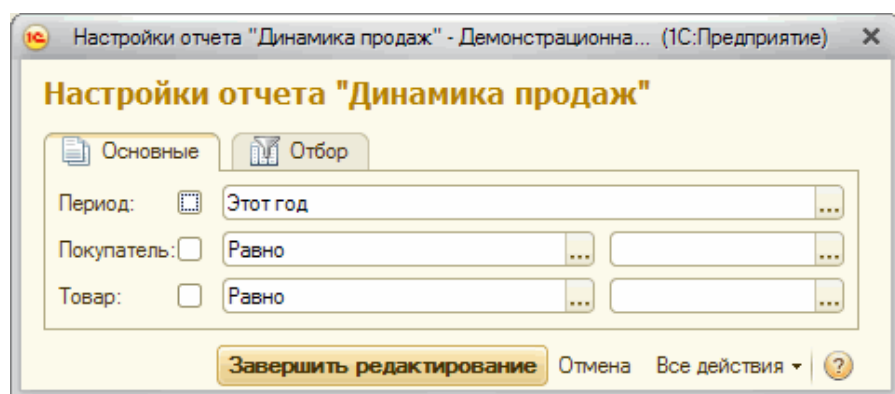


Рис. 115. Форма настроек отчета «Динамика продаж»

Для сохранения выбранных настроек следует нажать кнопку [Завершить редактирование](#).

Чтобы отредактировать состав настроек, отображаемых на форме отчета, нужно выбрать команду [Все действия – Изменить состав настроек](#). Форма редактирования списка настроек откроется в новом окне.

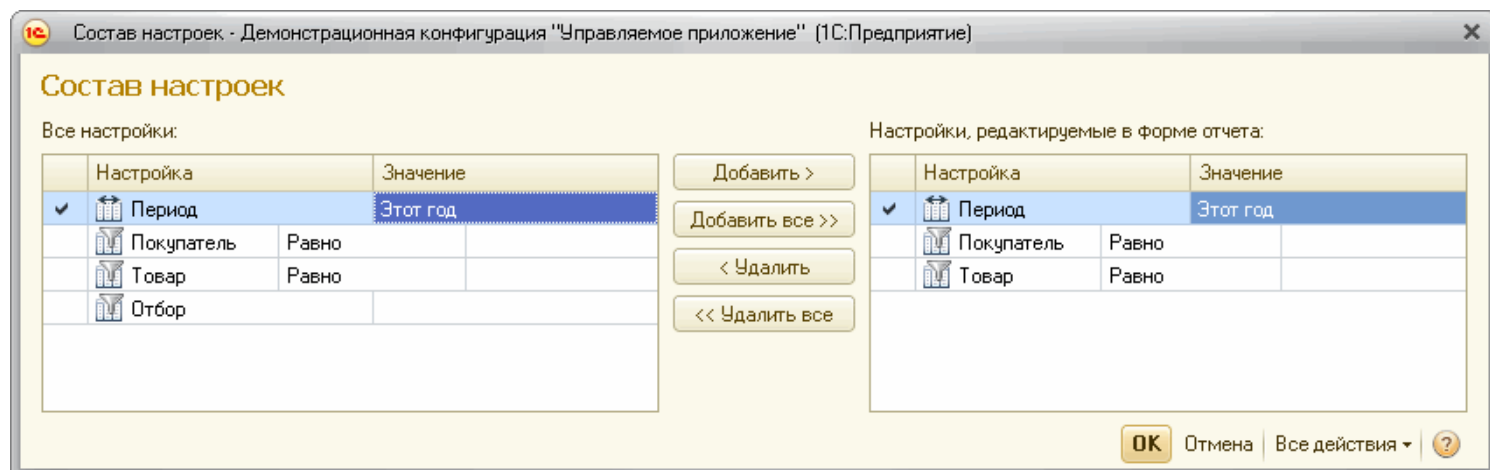


Рис. 116. Изменение состава настроек отчета

Слева отображаются все доступные настройки отчета, справа – настройки, редактируемые в форме отчета.

Для управления составом пользовательских настроек используются кнопки:

- [Добавить >](#) – добавляет настройку из левого списка в правый;
- [Добавить все >>](#) – добавляет все настройки из левого списка в правый;

- < **Удалить** – удаляет настройку из правого списка;
- << **Удалить все** – удаляет все настройки из правого списка;

Стандартные пользовательские настройки восстанавливаются командой **Все действия – Установить стандартные настройки**.

Если при помощи настроек не удастся получить требуемый отчет, то можно создать собственный вариант отчета. Создание варианта отчета требует наличия некоторого опыта работы с отчетами. Подробнее о работе с вариантами отчета см. главу «[Изменение варианта отчета](#)» [здесь](#).

При закрытии формы отчета или смене текущего варианта отчета пользовательские настройки автоматически сохраняются. Пользовательские настройки сохраняются отдельно для каждого варианта отчета.

Среди настроек могут быть выделены такие настройки, которые предполагается редактировать наиболее часто (например, фильтр по товару в отчете **Динамика продаж** или фильтр по организации в бухгалтерском отчете). Такие настройки будут редактироваться непосредственно в форме отчета.

Если в форме настроек отчета установить отбор, то элемент отбора будет добавлен в список настроек и в настройки, отображаемые в форме отчета.

Чтобы использовать в отчете выбранную настройку, установите значение настройки в соответствующем поле.

В зависимости от типа настройки для изменения значения следует выполнить одно из действий:

- выбрать значение из всплывающего списка по нажатию кнопки **Выбрать** в поле настройки,
- установить значение в новом окне, отобразившемся по нажатию кнопки **Выбрать** в поле настройки.

Пример выбора настройки отчета в новом окне представлен на рисунке:

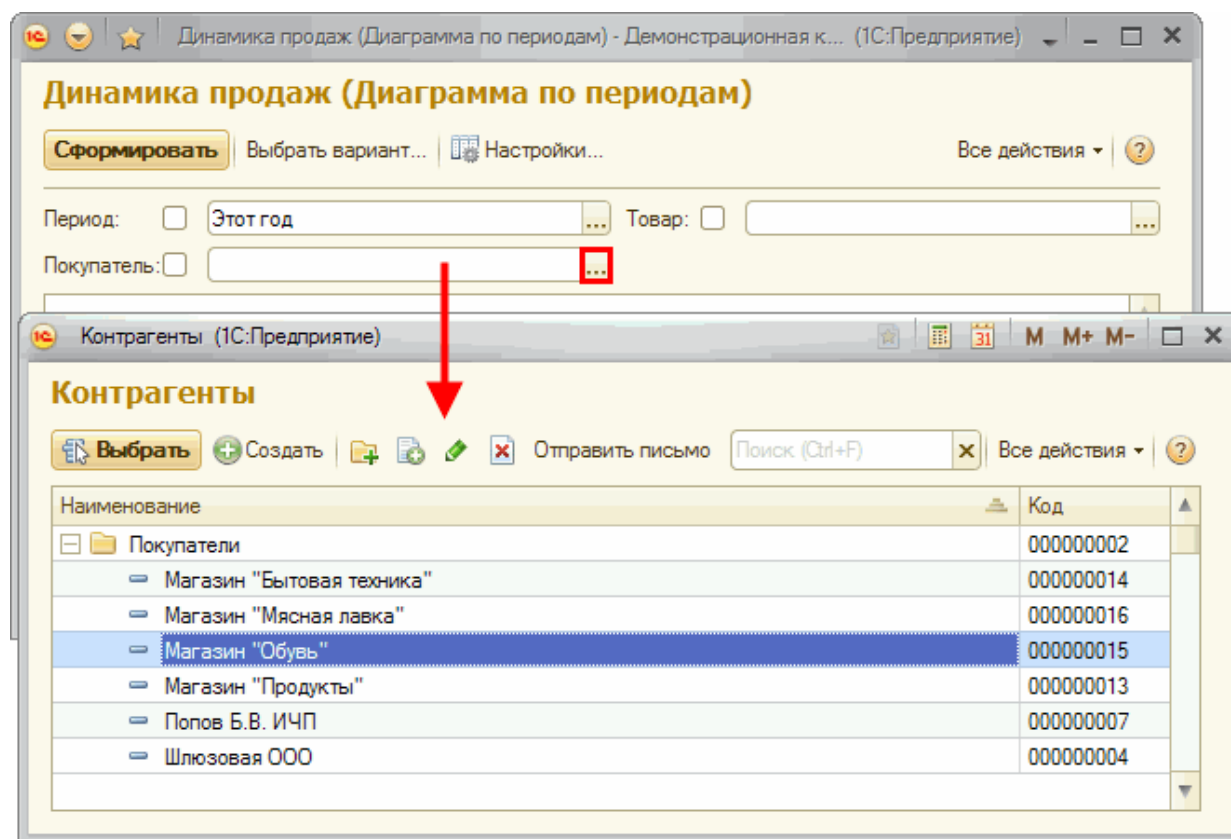


Рис. 117. Выбор значения быстрой настройки отчета

При изменении настроек не происходит формирования отчета.

Чтобы сформировать отчет с установленными настройками, следует нажать кнопку **Сформировать**.

Чтобы отменить использование выбранной настройки в отчете, нужно снять флажок в соответствующей строке.

8.2. Варианты отчета

Для отчета может быть создано несколько вариантов. Вариант определяет, в каком виде будут выводиться данные в результат отчета. Например, для отчета [Динамика продаж](#) может быть создан вариант, в котором данные о продаже товаров по периодам отображаются в виде диаграммы, и вариант, в котором данные по продажам для покупателей отображаются в виде таблицы. При этом каждый вариант отчета обладает своим набором настроек.

Система предоставляет возможность создания нового варианта отчета непосредственно в режиме [1С:Предприятие](#). Перед выполнением этого действия внимательно ознакомьтесь с разделом [«Изменение варианта отчета»](#). В случае вызова этого действия новый вариант отчета сохраняется в информационной базе.

Имеется возможность определять несколько вариантов настроек для каждого отчета.

Подробнее работа с вариантами настроек описана в главе [«Изменение варианта отчета»](#) [здесь](#).

8.3. Работа с отчетами

Рассмотрим подробнее работу пользователя с отчетом на примере отчета [Динамика продаж](#) демонстрационной конфигурации. Запустим отчет с помощью соответствующей команды на панели действий. Форма отчета откроется в новом окне.

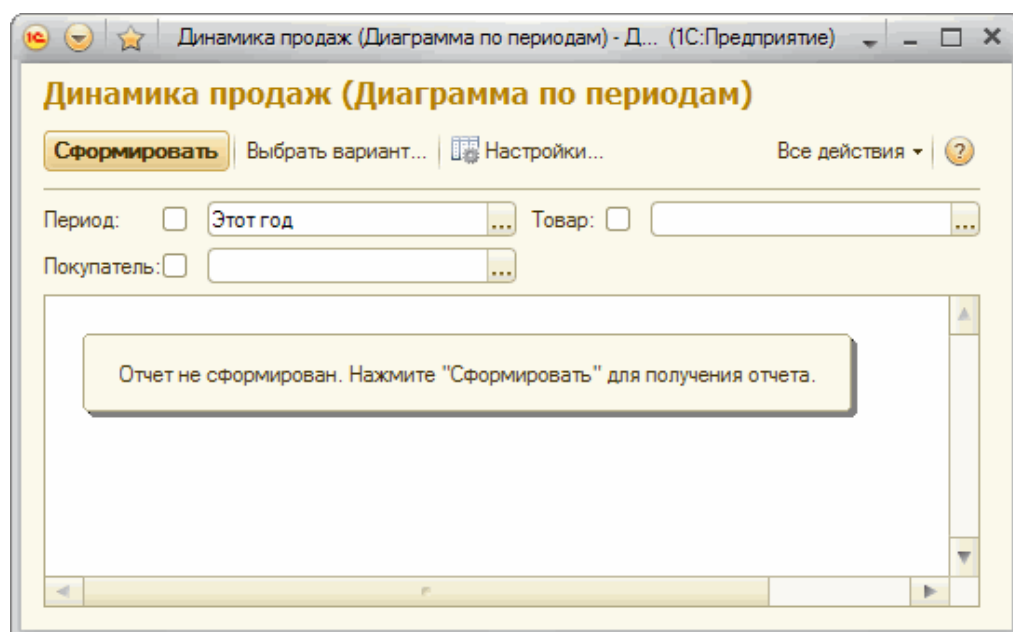


Рис. 118. Форма отчета «Динамика продаж»

При запуске для отчета установлен вариант [Диаграмма по периодам](#). Предположим, нам нужен другой вариант.

Тогда нужно нажать кнопку [Выбрать вариант...](#) и в новом окне выбрать вариант отчета [Продажи товаров по покупателям](#).

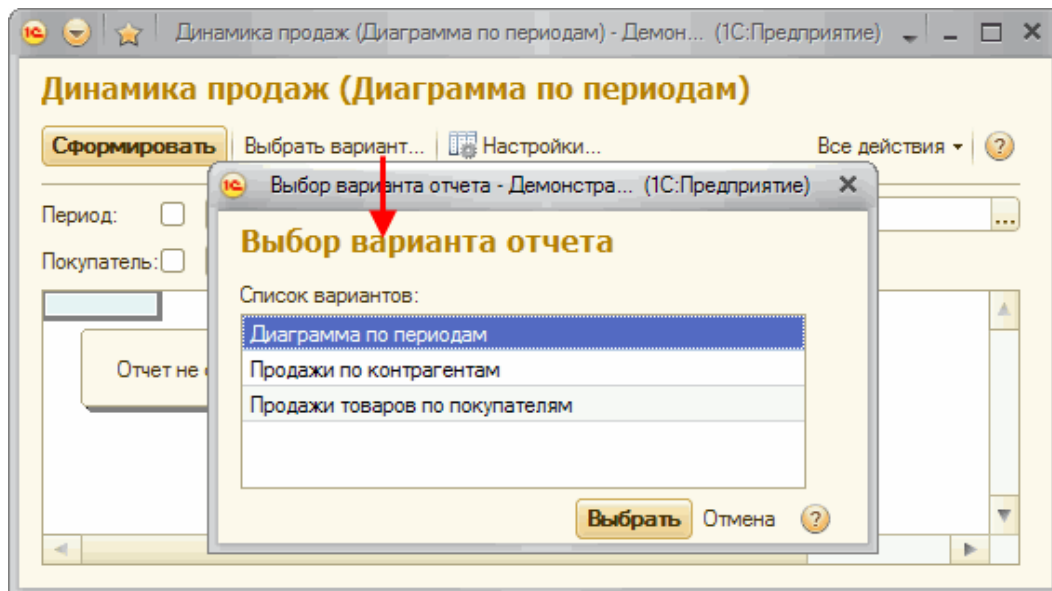


Рис. 119. Выбор варианта отчета «Динамика продаж»

Заметим, что этому варианту отчета были установлены настройки **Период – Этот год** и группировка по полям **Товар** и **Покупатель**.

Предположим, что это настройки, необходимые в данный момент, и сформируем отчет нажатием кнопки **Сформировать**.

Товар	Сумма
Покупатель	Оборот
Bosch1234	5 000,00
Попов Б.В. ИЧП	5 000,00
Bosch15	8 100,00
Попов Б.В. ИЧП	8 100,00
Sony K3456P	13 500,00
Шлюзовая ООО	13 500,00
Veko345MO	10 000,00
Магазин "Бытовая техника"	10 000,00
Veko67NE	15 000,00
Шлюзовая ООО	15 000,00
Босоножки	12 500,00
Попов Б.В. ИЧП	12 500,00
Ботинки	6 000,00
Магазин "Обувь"	6 000,00
Валенки	4 500,00
Магазин "Обувь"	4 500,00

Рис. 120. Отчет «Динамика продаж» в варианте «Продажи товаров по покупателям»

ПРИМЕЧАНИЕ. Для удобства работы в отчетах, содержащих одну таблицу или группировку, заголовок таблицы автоматически фиксируется. При этом если область фиксации табличного документа превышает область экрана, фиксация не используется.

8.3.1. Использование быстрых настроек

Допустим, необходимо получить отчет о продажах за предыдущий год. Для этого нам следует изменить параметр **Период** и нажать кнопку **Сформировать**:

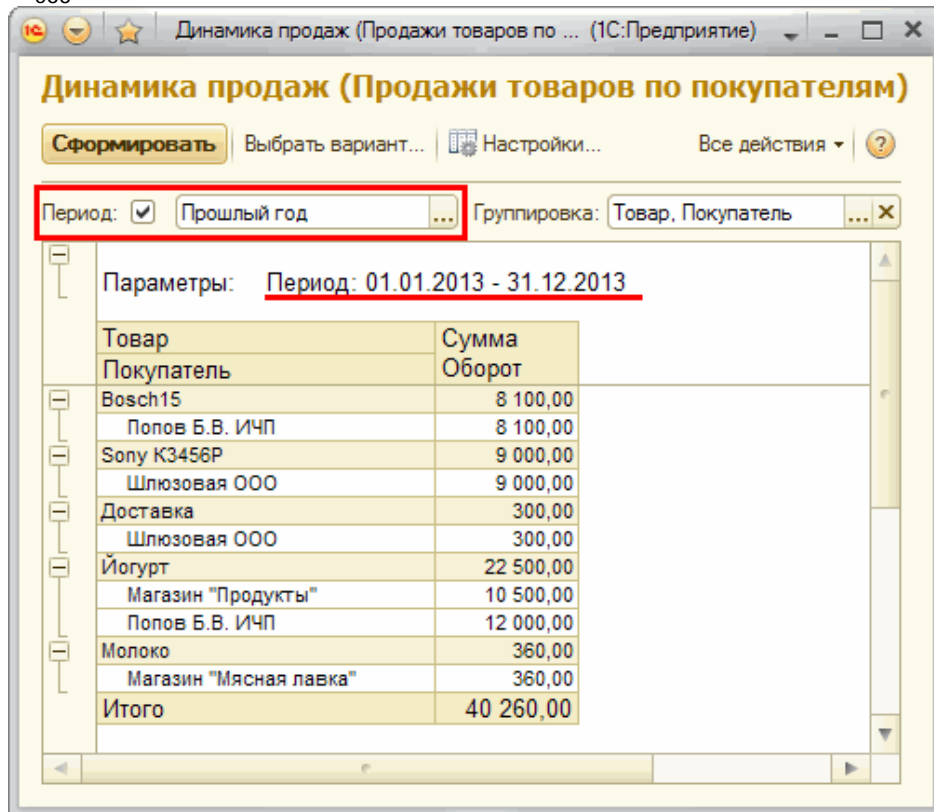


Рис. 121. Отчет «Динамика продаж» за прошлый год

8.3.2. Настройки

Предположим, нам недостаточно изменить видимые параметры ([Период](#) и [Группировка](#)). Чтобы увидеть параметры, которые можно настроить для варианта отчета, следует нажать кнопку [Настройка](#). Список настроек откроется в отдельном окне:

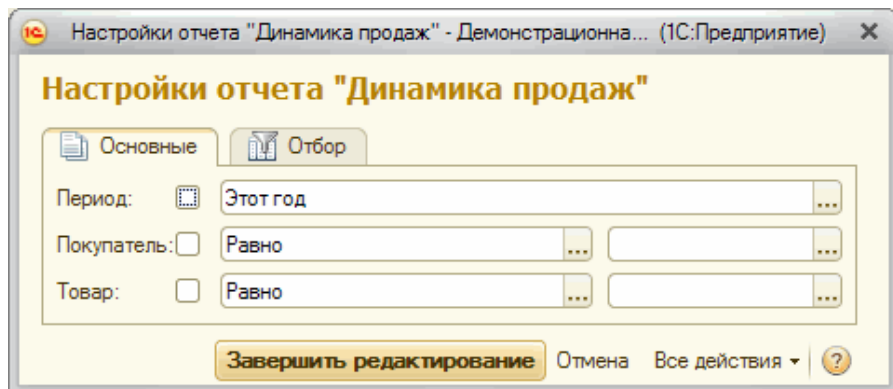


Рис. 122. Форма настроек отчета «Динамика продаж»

8.3.2.1. Отбор

Чтобы установить отбор, следует на форме настроек установить значение отбора. Для этого следует перейти на закладку [Отбор](#).

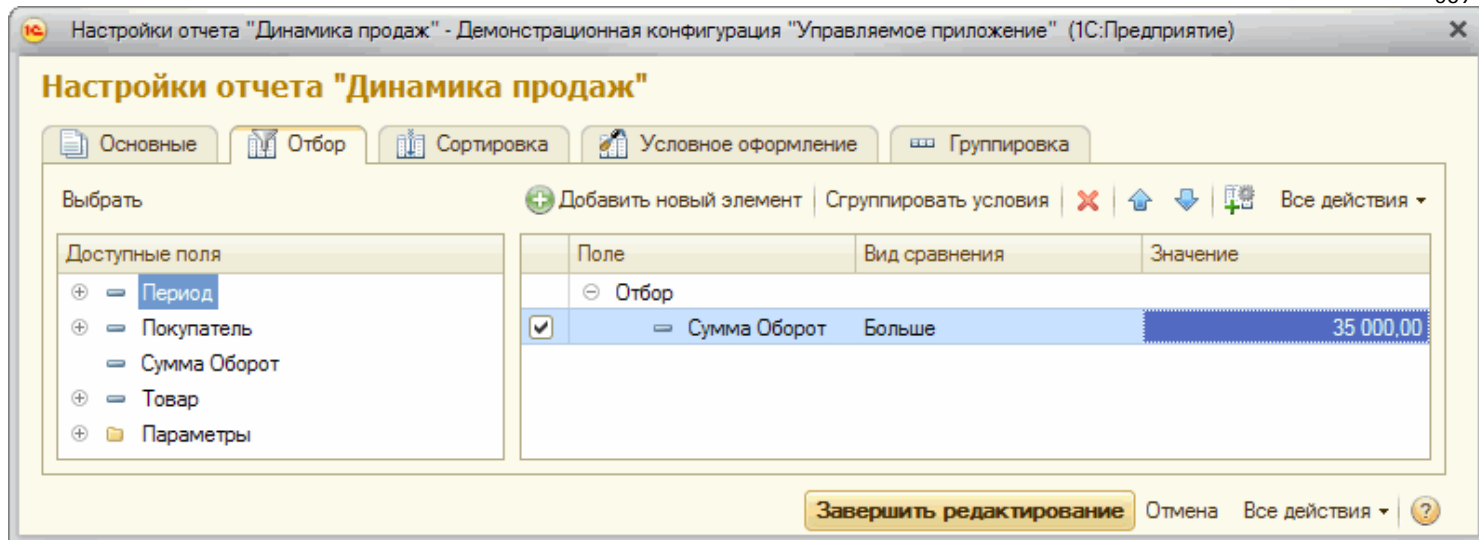


Рис. 123. Форма настройки отбора для отчета «Динамика продаж»

1. На закладке **Отбор** следует нажать кнопку **Добавить новый элемент** или перетащить поле из списка **Доступные поля**.
2. Выбрать **Вид сравнения** в списке, всплывающем при нажатии кнопки **Выбрать**.
3. Установить значение отбора, выбрав его в списке, открывающемся при нажатии кнопки **Выбрать**, или набрать его в колонке.
4. Если необходимо, установить порядок элементов отбора с помощью кнопок **Переместить вверх** и **Переместить вниз**.
5. Нажать кнопку **Завершить редактирование**.

Подробнее о работе с отбором и о видах сравнения см. раздел «[Отбор](#)» [здесь](#).

8.3.2.2. Сортировка

Чтобы установить порядок записей результата отчета, в форме настроек перейти на закладку **Сортировка**.

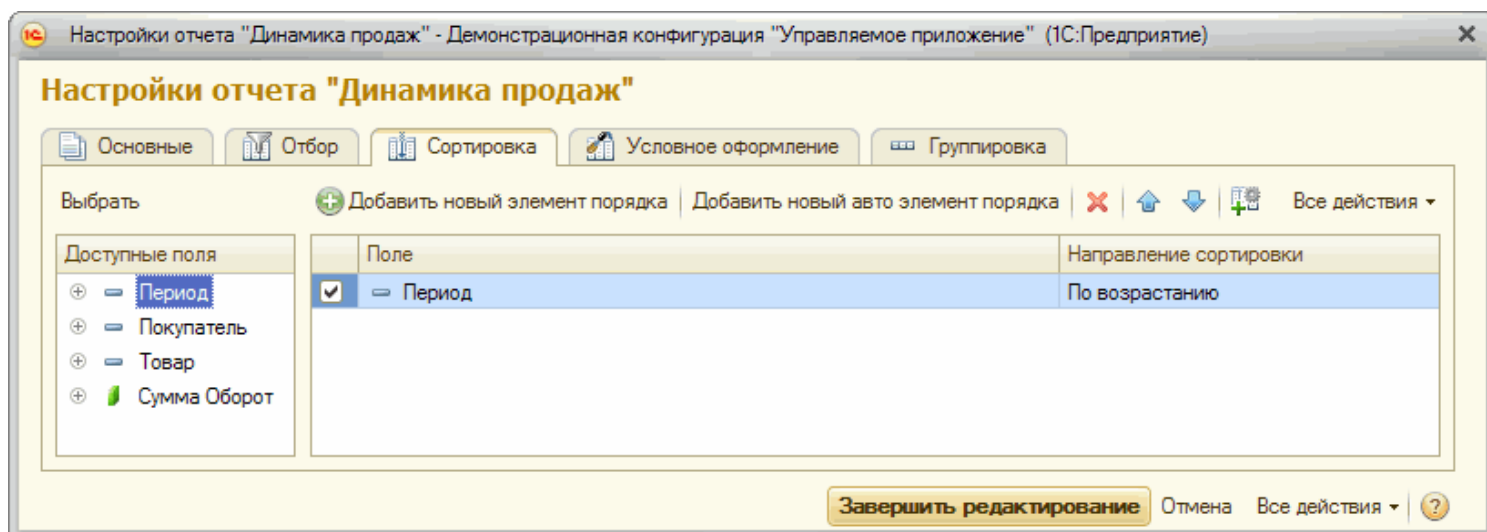


Рис. 124. Настройка сортировки результата отчета «Динамика продаж»

1. В открывшемся окне нажать **Добавить новый элемент** и выбрать поле или перетащить поле из списка **Доступные поля**.
2. Выбрать **Направление сортировки** (**По возрастанию/По убыванию**).
3. Если необходимо, установить порядок элементов сортировки с помощью кнопок **Переместить вверх** и **Переместить вниз**.

4. Нажать кнопку [Завершить редактирование](#).

Если полей сортировки несколько, то в отчете данные будут отсортированы вначале по первому полю, потом по второму и т. д.

Подробнее о работе с сортировкой см. раздел «[Сортировка](#)» [здесь](#).

Пример настройки сортировки

В качестве примера отсортируем отчет [Отчет по оплатам](#) по поставщику и в разрезе поставщика – по дате документа. Настройки отбора будут выглядеть следующим образом:

	Поле	Направление сортировки
<input checked="" type="checkbox"/>	Поставщик	По возрастанию
<input checked="" type="checkbox"/>	Дата	По возрастанию

Рис. 125. Настройка сортировки в отчете

В форме редактирования нажмем [Завершить редактирование](#). Результирующий отчет примет вид, как на рисунке:

Дата	Номер	Поставщик	Расчетный счет поставщика	Сумма
18.01.2009 14:29:35	000000014	Животноводство ООО	Основной	50 000,00
18.01.2009 22:11:11	000000009	Животноводство ООО	Основной	62 500,00
07.02.2009 12:18:23	000000003	Животноводство ООО	Основной	9 000,00
03.03.2009 14:00:41	000000015	Животноводство ООО	Основной	50 000,00
07.01.2009 12:19:03	000000004	Корнет ЗАО	Основной	60 000,00
01.03.2009 14:06:12	000000011	Мосхлеб ОАО	Основной	4 500,00
02.03.2009 12:19:28	000000005	Мосхлеб ОАО	Основной	4 500,00
05.12.2008 12:19:50	000000006	Пантера АО	Основной	50 000,00
27.01.2009 13:20:03	000000010	Пантера АО	Основной	150 000,00
11.11.2008 9:31:27	000000002	Скорход АО	Основной	2 000,00
09.11.2008 12:20:14	000000007	ЭлетроБыт ЗАО	Основной	50 000,00
09.01.2009 11:23:32	000000008	ЭлетроБыт ЗАО	Основной	70 000,00
15.02.2009 14:16:36	000000013	ЭлетроБыт ЗАО	Основной	500 000,00
23.02.2009 13:14:40	000000012	ЭлетроБыт ЗАО	Основной	1 100 000,00

Рис. 126. Результат настройки сортировки отчета

8.3.2.3. Условное оформление

Чтобы установить условное оформление записей результата отчета, следует на форме настроек в параметре [Условное оформление](#) нажать кнопку выбора или клавишу **F4**.

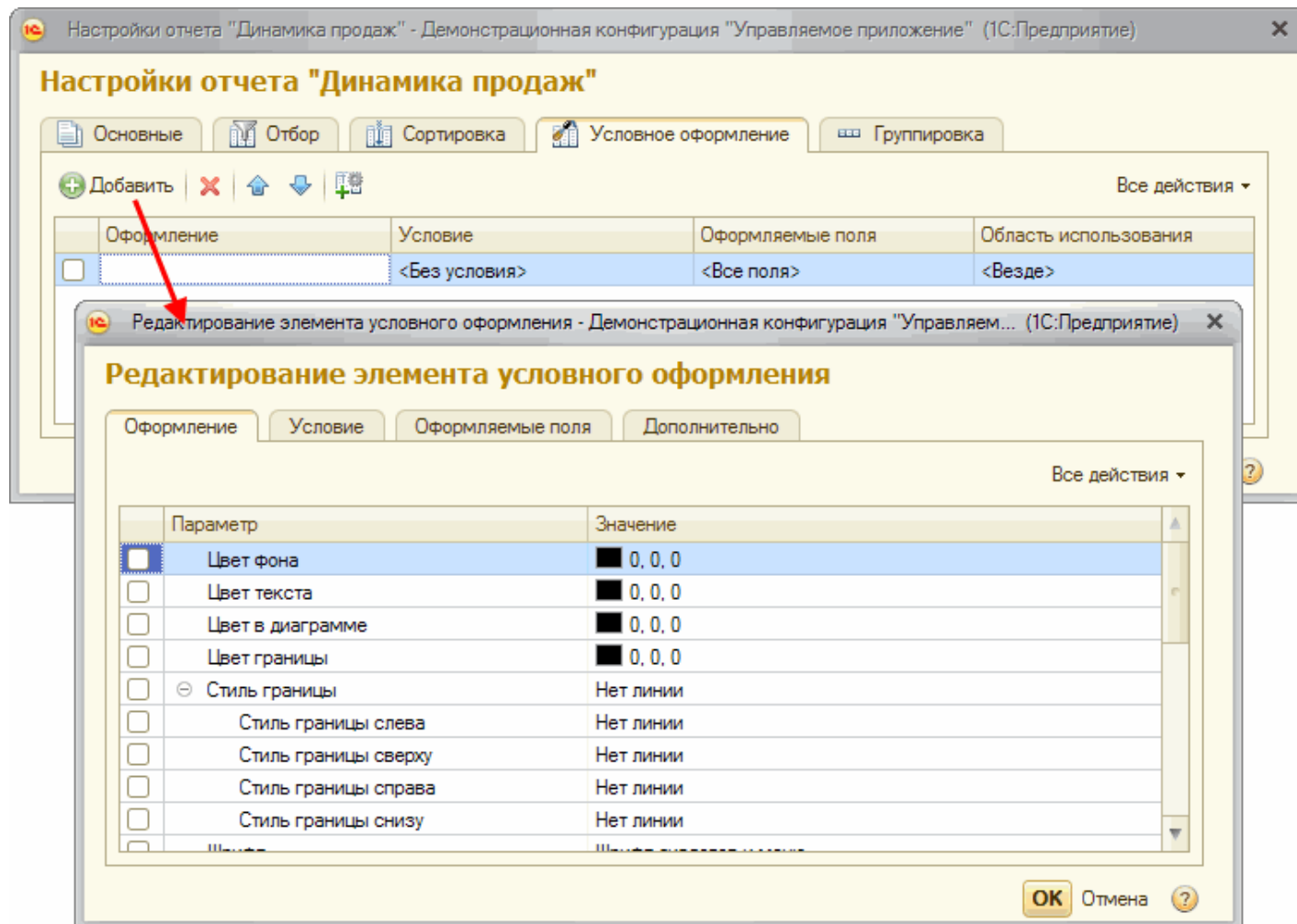


Рис. 127. Настройка условного оформления отчета

1. В открывшемся окне нажать кнопку **Добавить**.
2. На закладке **Оформление** выбрать требуемые параметры оформления: цвет фона, цвет текста и т. д.
3. На закладке **Условие** поставить условие, при выполнении которого оформление будет применяться к полям. Условие ставится аналогично отбору.
4. На закладке **Оформляемые поля** выбрать поля из списка доступных полей, к которым будет применяться оформление, если условие выполнено.
5. Если необходимо, установить порядок используемых элементов оформления с помощью кнопок **Переместить вверх** и **Переместить вниз**.

Порядок элементов оформления устанавливает последовательность, в которой оформление будет применяться к элементам отчета. Если по некоторым условиям к области отчета применяются два разных оформления, то в итоге применяется последнее из списка элементов условного оформления.

6. На закладке **Дополнительно** выбрать области отчета, к которым будет применен данный элемент условного оформления.
7. Нажать кнопку **OK**.

Подробнее о настройке условного оформления см. раздел [«Условное оформление»](#) [здесь](#).

Пример настройки условного оформления

В качестве примера в отчете **Отчет по взаиморасчетам** выделим документы, которые относятся к **ООО Животноводство**.

Для этого в настройках отчета выполним следующие действия:

1. На закладке **Условное оформление** добавим новый элемент.

2. В диалоге редактирования элемента условного оформления на закладке **Оформление** нажмем кнопку **Выбрать** и в диалоге **Выбор шрифта** установим флажок напротив пункта **Жирный** в группе **Начертание**.

3. На закладке **Условие** добавим элемент: **Документ.Организация Равно ООО Животноводство**.

4. Нажмем **ОК** и убедимся, что в колонке **Оформляемые поля** отображается **<Все поля>**, а в колонке **Область использования** отображается **<Везде>**.

Настройка условного оформления отчета примет вид, как показано на рисунке:

Оформление	Условие	Оформляемые поля	Область использования
<input checked="" type="checkbox"/> Шрифт	Контрагент Равно "Животноводст...	<Все поля>	<Везде>

Рис. 128. Пример настройки условного оформления

В диалоге настройки варианта нажмем кнопку **Завершить редактирование**. Отчет примет вид, как показано на рисунке.

Дата	Номер	Поставщик	Расчетный счет поставщика	Сумма
07.04.2008 12:18:23	000000003	Животноводство ООО	Основной	9 000,00
07.03.2008 12:19:03	000000004	Корнет ЗАО	Основной	60 000,00
06.05.2008 12:19:28	000000005	Мосхлеб ОАО	Основной	4 500,00
05.02.2008 12:19:50	000000006	Пантера АО	Основной	50 000,00
09.01.2008 12:20:14	000000007	ЭлектроБыт ЗАО	Основной	50 000,00
09.03.2008 12:20:32	000000008	ЭлектроБыт ЗАО	Основной	70 000,00
18.03.2008 12:57:57	000000009	Животноводство ООО	Основной	62 500,00
27.03.2008 13:20:03	000000010	Пантера АО	Основной	150 000,00
29.04.2008 14:06:12	000000011	Мосхлеб ОАО	Основной	4 500,00
23.04.2008 13:14:40	000000012	ЭлектроБыт ЗАО	Основной	1 100 000,00
15.04.2008 14:16:36	000000013	ЭлектроБыт ЗАО	Основной	500 000,00
18.03.2008 14:29:35	000000014	Животноводство ООО	Основной	50 000,00
29.05.2008 14:40:29	000000015	Животноводство ООО	Основной	50 000,00
11.01.2008 9:31:27	000000002	Скороход АО	Основной	2 000,00
10.07.2009 16:12:25	000000016	Магазин "Продукты"	Основной	12,00

Рис. 129. Настройка условного оформления отчета

8.3.2.4. Группировка

Чтобы создать группировку результата отчета, на форме настроек перейти на закладку **Группировка** и добавить элемент в табличном поле.

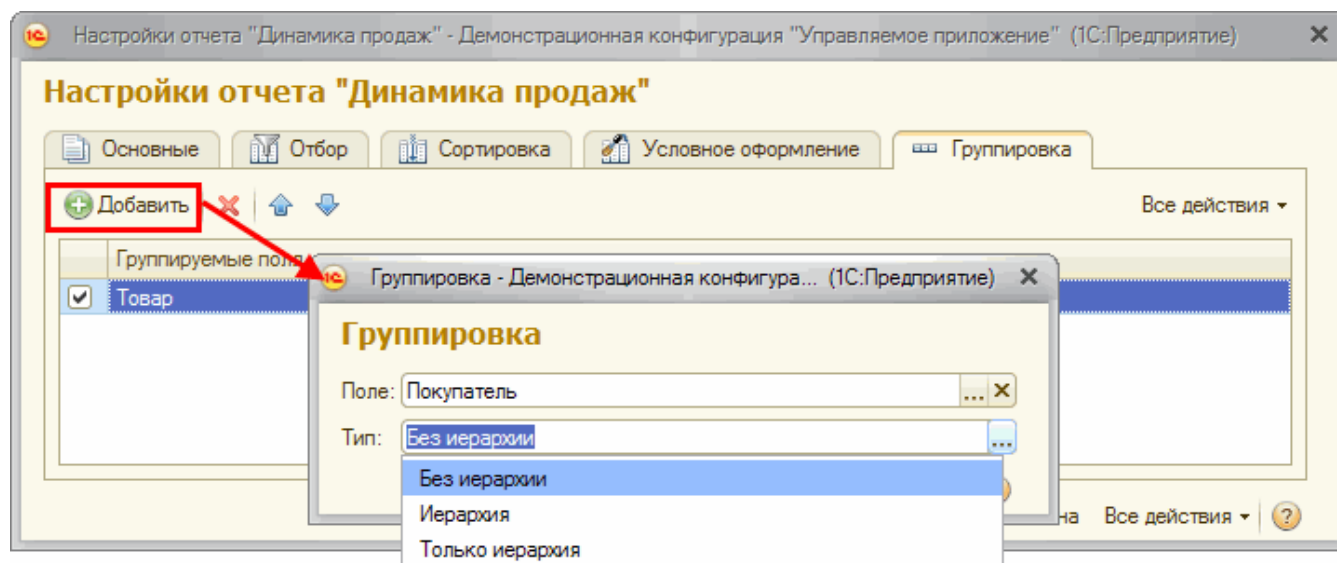


Рис. 130. Настройка группировки отчета

1. Нажать кнопку **Добавить**.

2. В открывшемся окне выбрать **Поле группировки** и указать тип группировки:

- **Без иерархии** – в группировку попадут только неиерархические записи (элементы);
- **Иерархия** – в группировку попадут иерархические записи и неиерархические записи (элементы и группы);
- **Только иерархия** – в группировке будут выводиться только иерархические записи (группы).

3. Нажать кнопку **ОК**.

4. Если необходимо, установить порядок группировок с помощью кнопок **Переместить вверх** и **Переместить вниз**.

5. Нажать кнопку **Завершить редактирование** в диалоге настроек.

Подробнее о работе с группировками см. раздел [«Группируемые поля»](#) [здесь](#).

Пример настройки группировки

В качестве примера сгруппируем результат отчета **Отчет по оплатам** по имени поставщика. Для этого в качестве поля группировки выберем поле **Поставщик** с типом **Иерархия**.

Результат выполнения отчета показан на рисунке:

Поставщик			
Дата	Номер	Расчетный счет поставщика	Сумма
Поставщики			
Животноводство ООО			
07.02.2009 12:18:23	000000003	Основной	9 000,00
18.01.2009 22:11:11	000000009	Основной	62 500,00
18.01.2009 14:29:35	000000014	Основной	50 000,00
03.03.2009 14:00:41	000000015	Основной	50 000,00
Корнет ЗАО			
07.01.2009 12:19:03	000000004	Основной	60 000,00
Мосхлеб ОАО			
02.03.2009 12:19:28	000000005	Основной	4 500,00
01.03.2009 14:06:12	000000011	Основной	4 500,00
Пантера АО			
05.12.2008 12:19:50	000000006	Основной	50 000,00
27.01.2009 13:20:03	000000010	Основной	150 000,00

Рис. 131. Отчет по оплатам, сгруппированный по полю «Поставщик»

8.4. Работа с расшифровкой отчета

При наведении указателя мыши на ячейку результата отчета, для которой возможна расшифровка, указатель изменит свою форму (см. рисунок). Теперь, если дважды щелкнуть левой кнопкой мыши на этой ячейке, то в новом окне откроется значение, находящееся в ячейке, а для ячейки с ресурсами будет предложено выбрать поле для расшифровки, после чего выполнится расшифровка.

При щелчке правой кнопкой мыши отобразится меню расшифровки.

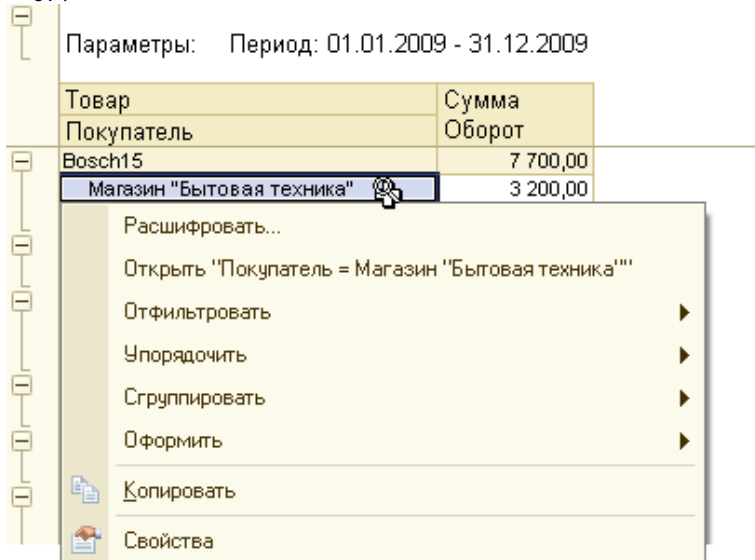


Рис. 132. Меню расшифровки для ячейки отчета

Расшифровать – получить более детальную информацию о содержимом поля. Любая расшифровка выполняется в новом окне.

Например, узнаем, когда продавались чайники Bosch15. Для этого в сформированном отчете **Динамика продаж** выберем ячейку Bosch15 и вызовем команду **Расшифровать**. В появившемся окне выбора поля выберем поле **Период**.

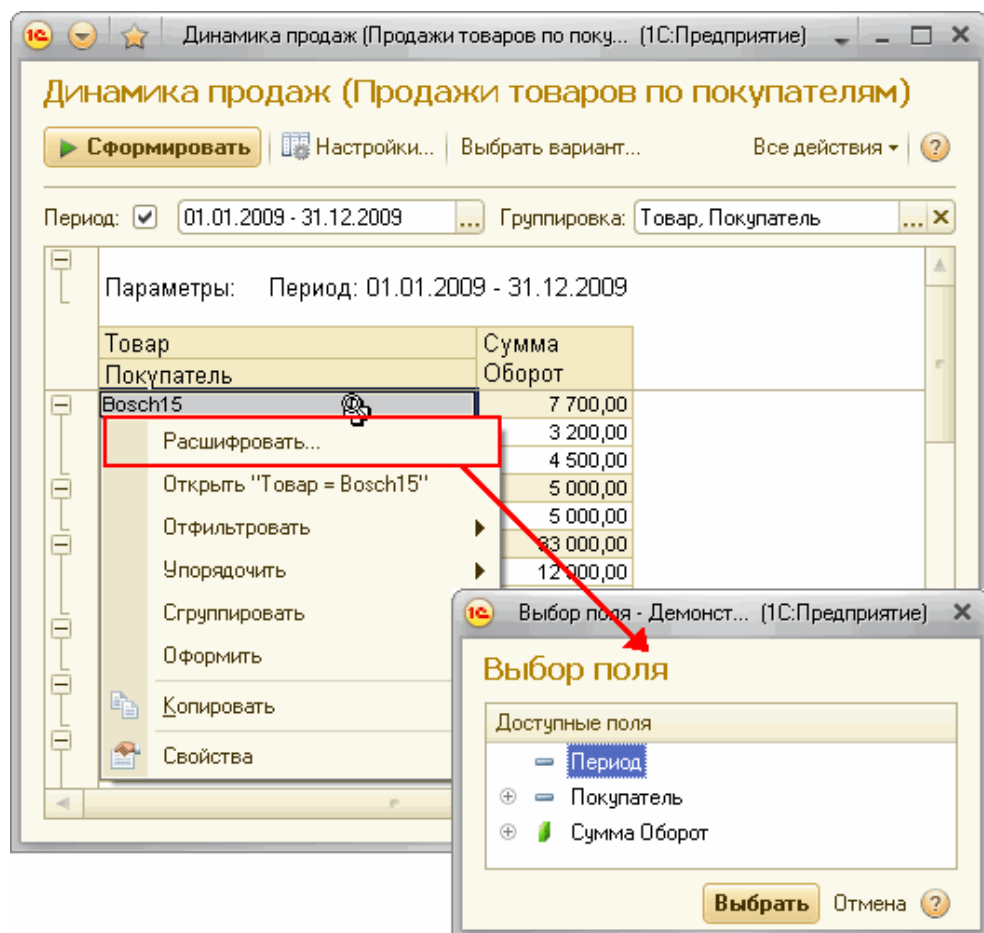


Рис. 133. Выбор поля расшифровки отчета

Нажмем кнопку **Выбрать**. Детали расшифровки откроются в новом окне.

Параметры: Период: 01.01.2009 - 31.12.2009
 Отбор: Товар Равно "Bosch15"

Период	Сумма Оборот
02.02.2009 0:00:00	3 200,00
23.02.2009 0:00:00	4 500,00
Итого	7 700,00

Рис. 134. Расшифровка отчета

Команда **Открыть "..."** открывает форму просмотра данных, показанных в ячейке. Например, окно с формой элемента списка.

The screenshot shows a window titled 'Bosch15 (Товар)'. On the left is a sidebar with 'Перейти' (Go) and links for 'Файлы', 'Характеристики товаров', and 'Цены товаров'. The main area has a header with 'Записать и закрыть', 'Печать', and 'Все действия'. Below are input fields for 'Группа' (Чайники), 'Код' (000000018), 'Наименование' (Bosch15), 'Вид' (Товар selected), 'Артикул' (4-0002), 'Поставщик' (ЭлектроБыт ЗАО), and 'Штрихкод'. At the bottom, there are tabs for 'Изображение' and 'Характеристики', with the 'Изображение' tab showing a picture of a blue and silver electric kettle.

Рис. 135. Форма элемента списка

Отфильтровать – позволяет отфильтровать текущий элемент структуры отчета по значению выбранного поля либо установить отбор по значению другого поля, используя команду **Дополнительно** и задав условие отбора. Например, отфильтруем строки отчета, в которых **Сумма (Оборот)** больше 50 000:

The screenshot shows a dialog box titled 'Редактирование условия отбора'. It has three main input areas: 'Левое значение' with a dropdown menu showing 'Сумма (Оборот)', 'Вид сравнения' with a dropdown menu showing 'Больше', and 'Правое значение' with a text input field containing '50 000,00'. At the bottom are 'OK', 'Отмена', and a help icon.

Рис. 136. Диалог редактирования условия отбора

Результат отображается в новом окне.

Параметры: Период: 01.01.2009 - 31.12.2009
 Отбор: Сумма Оборот Больше "50 000"

Товар	Сумма
Покупатель	Оборот
Кроссовки	85 000,00
Магазин "Обувь"	85 000,00
Итого	85 000,00

Рис. 137. Результат отбора

Упорядочить – выполняет упорядочивание содержимого текущего элемента отчета по значению текущего поля. По команде **Дополнительно** можно задать произвольное поле. Например, упорядочить таблицу в отчете по полю **Товар**.

Параметры: Период: 01.01.2009 - 31.12.2009

Товар	Сумма
Покупатель	Оборот
Bosch15	7 700,00
Магазин "Бытовая техника"	3 200,00
Магазин "Обувь"	4 500,00
Veko345MO	5 000,00
Пантера АО	5 000,00
Veko67NE	33 000,00
Магазин "Бытовая техника"	12 000,00
Пантера АО	21 000,00
Veko876N	12 000,00
Магазин "Бытовая техника"	12 000,00
VekoNT02	7 500,00
Магазин "Бытовая техника"	7 500,00
Ботинки	29 100,00
Магазин "Обувь"	25 000,00
Мосхлеб ОАО	2 100,00
Пантера АО	2 000,00
Валенки	1 800,00
Мосхлеб ОАО	1 800,00
Вихрь	7 000,00
Магазин "Бытовая техника"	7 000,00
Доставка	6 200,00
Магазин "Обувь"	200,00
Пантера АО	6 000,00

Рис. 138. Результат упорядочивания отчета

Сгруппировать – выполняет выбор поля и устанавливает тип группировки в текущем элементе структуры отчета. Например, в отчете **Остатки товаров** нужно узнать, какая обувь на каком складе находится. Для этого выберем поле группировки **Склад**.

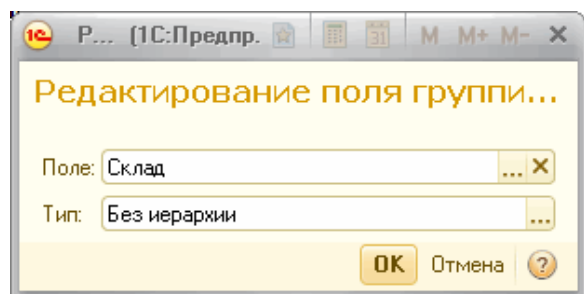


Рис. 139. Выбор поля группировки

Отфильтруем по значению **Товар В Группе "Обувь"**. Результат отчета приведен ниже.

Отбор: Товар В группе "Обувь"

Склад	Большой	Малый	Средний	Итого
Товар	Количество Остаток	Количество Остаток	Количество Остаток	Количество Остаток
Большой	88,00			88,00
Босоножки	51,00			51,00
Ботинки	12,00			12,00
Кроссовки	3,00			3,00
Сапоги	14,00			14,00
Тапочки	7,00			7,00
Туфли	1,00			1,00
Малый		14,00		14,00
Ботинки		5,00		5,00
Сапоги		4,00		4,00
Тапочки		5,00		5,00
Средний			8,00	8,00
Пинетки			5,00	5,00
Сапоги			1,00	1,00
Туфли			2,00	2,00
Итого	88,00	14,00	8,00	110,00

Рис. 140. Результат группировки отчета

Оформить – позволяет указать условие применения оформления и настроить оформление для текущего элемента отчета. Например, оформить поля, где **Количество Остаток < 10** единиц.

Отбор: Товар В группе "Обувь"

Склад	Большой	Малый	Средний	Итого
Товар	Количество Остаток	Количество Остаток	Количество Остаток	Количество Остаток
Большой	88,00			88,00
Босоножки	51,00			51,00
Ботинки	12,00			12,00
Кроссовки	3,00			3,00
Сапоги	14,00			14,00
Тапочки	7,00			7,00
Туфли	1,00			1,00
Малый		14,00		14,00
Ботинки		5,00		5,00
Сапоги		4,00		4,00
Тапочки		5,00		5,00
Средний			8,00	8,00
Пинетки			5,00	5,00
Сапоги			1,00	1,00
Туфли			2,00	2,00
Итого	88,00	14,00	8,00	110,00

Рис. 141. Результат оформления отчета

Глава 9. Изменение варианта отчета

Изменять вариант настройки отчета рекомендуется только опытным пользователям, знакомым с настройкой отчетов в системе компоновки данных. Для того чтобы отредактировать вариант отчета, следует использовать команду [Все действия – Изменить вариант...](#) При этом откроется форма варианта отчета.

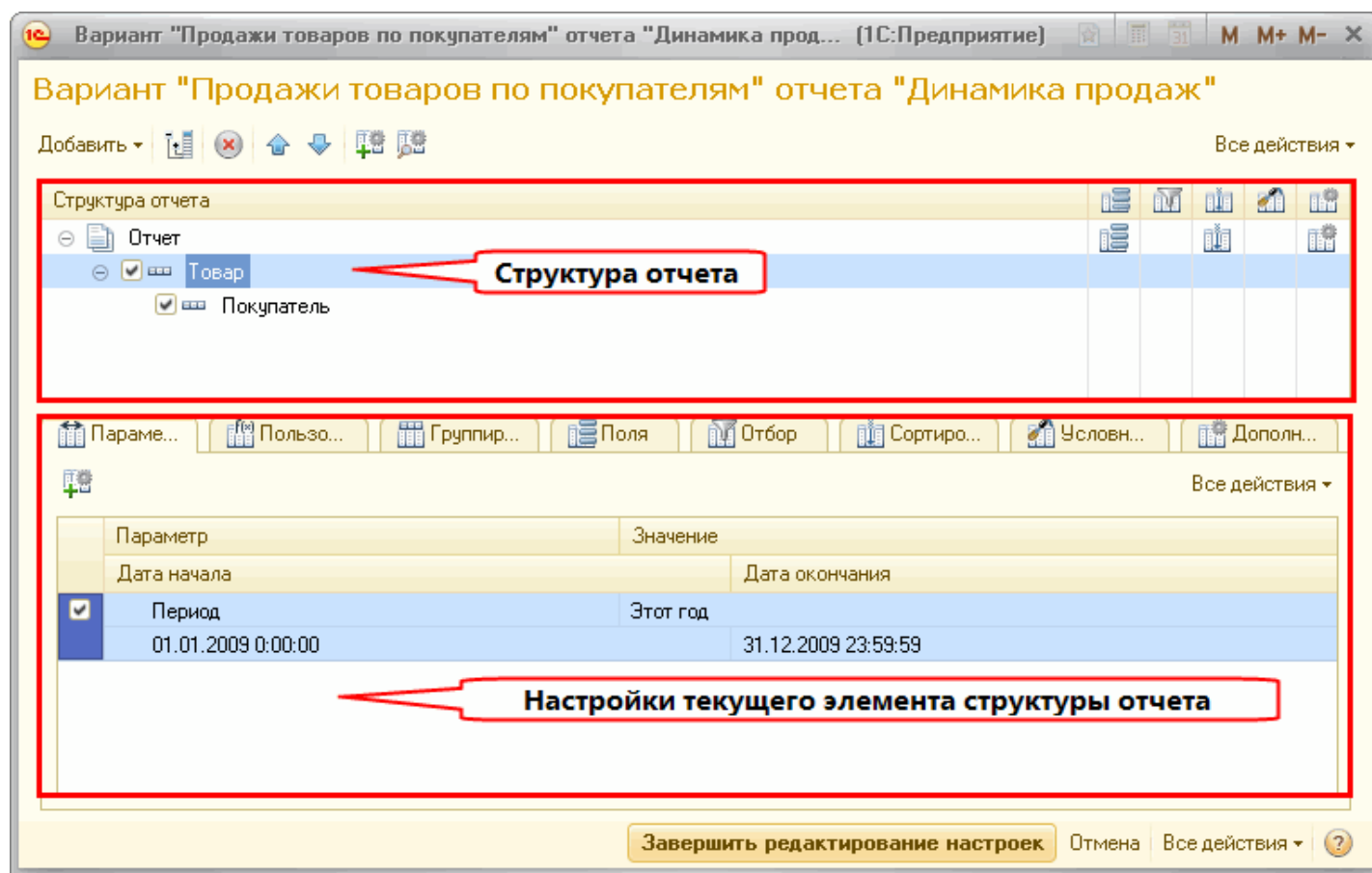


Рис. 142. Форма настройки варианта отчета

Настройка варианта отчета заключается в изменении структуры отчета и установке необходимых параметров, создании пользовательских полей, установке полей группировки, отбора, сортировки для каждого элемента отчета.

В любой момент измененный вариант настройки отчета можно вернуть в первоначальный вид, используя команду [Все действия – Стандартные настройки](#) командной панели формы варианта отчета.

Команда [Все действия – Сохранить настройки](#) позволяет сохранить настройки отчета в файл.

Настройки из файла могут быть загружены командой [Все действия – Загрузить настройки](#). Настройки загружаются с учетом имеющихся прав доступа у текущего пользователя.

Команда [Изменить форму](#) меню [Все действия](#) позволяет изменять саму форму настройки варианта отчета. Подробнее о настройке формы описано в разделе «Настройка формы» [здесь](#).

9.1. Настройка отчета в целом

Для того чтобы установить настройки для всего отчета, следует выбрать узел [Отчет](#) в структуре отчета, как это сделано на [рис. 143](#). Состав настроек (и соответствующих им закладок) зависит от наличия доступных полей и выбранного элемента настройки.

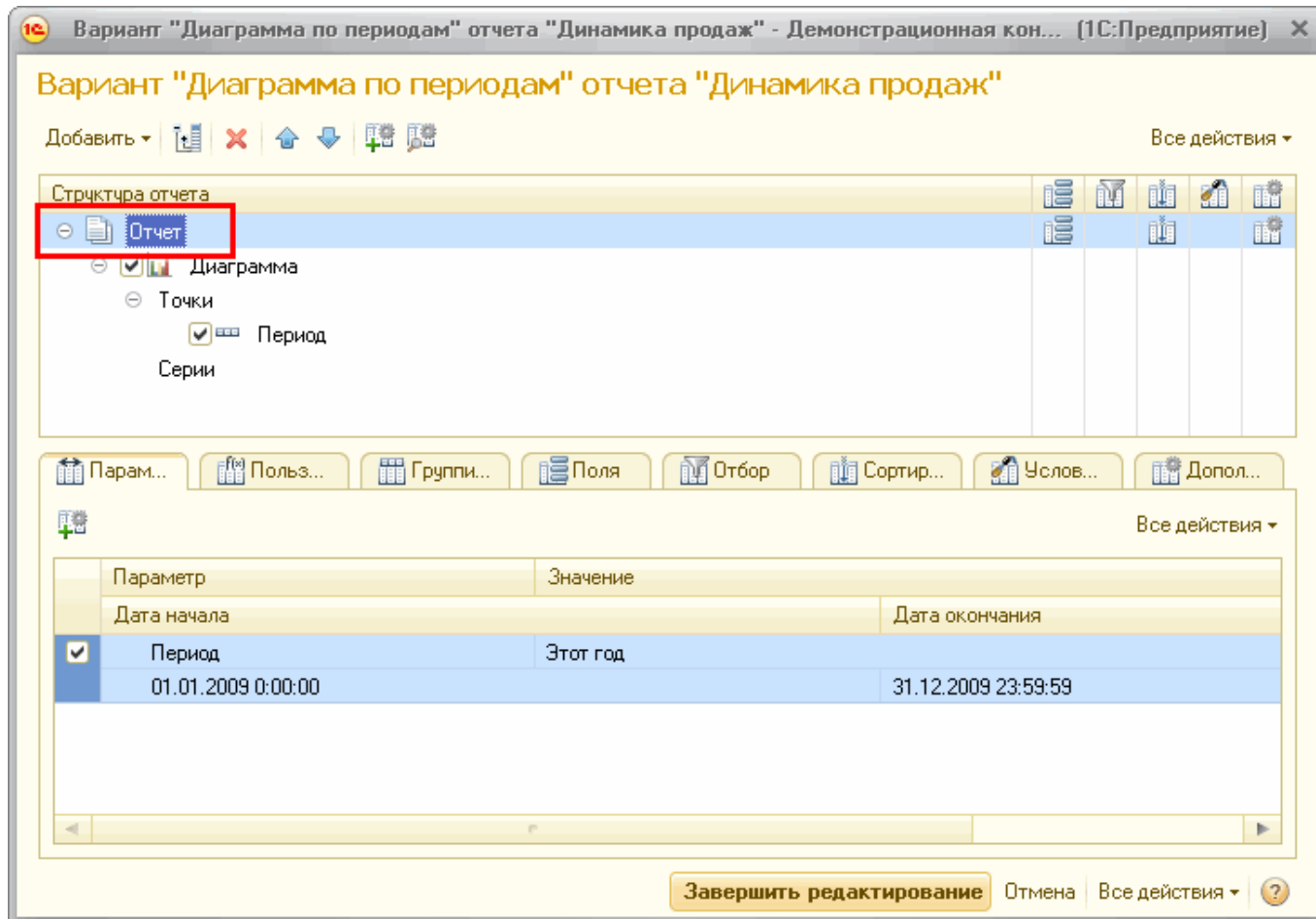


Рис. 143. Настройка варианта отчета

9.2. Изменение структуры отчета

Элементами отчета в системе компоновки данных могут быть группировка, таблица, диаграмма, а также вложенный отчет.

Группировка – выводит данные в виде списка. Чтобы добавить группировку в отчет, нужно выбрать пункт **Добавить – Новая группировка** или нажать клавишу **Ins**. В появившемся окне выбрать поле и тип группировки, как показано на рисунке.

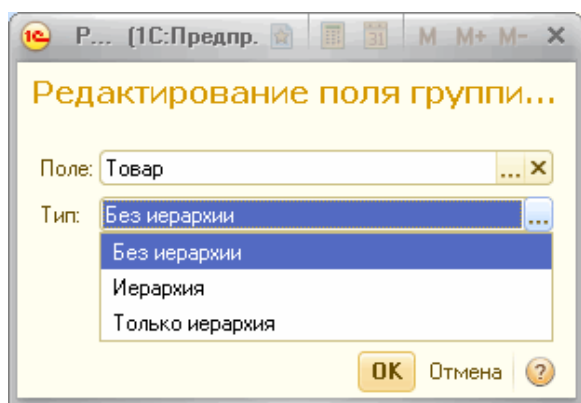


Рис. 144. Форма редактирования группировки

Если поле группировки не выбрано, то в отчет будут добавлены детальные записи – записи, полученные из базы данных в том порядке, в каком они расположены в базе.

В простейшем случае после добавления группировки по полю **Товар** отчет примет вид, как на рисунке ниже.

600

Отчет

Товар

Товар	Количество (Остаток)
Босоножки	100,00
Ботинки	13,00
Валенки	23,00
Кроссовки	3,00
Пинетки	5,00
Сапоги	18,00
Сланцы	4,00
Тапочки	12,00
Туфли	8,00
Итого	186,00

Рис. 145. Группировка в отчете

СОВЕТ. Также можно выбрать элементы структуры, которые требуется сгруппировать, и выбрать команду **Все действия – Сгруппировать**.

Команда **Разгруппировать** удаляет группировку, оставляя ее содержимое.

Таблица – выводит данные в виде таблицы. Чтобы добавить таблицу в структуру, нужно выбрать пункт **Добавить – Новая таблица**. В качестве строк и колонок таблицы используются группировки.

Например, в структуре отчета может быть создана таблица, строками которой является список товаров, а в колонках отображаются данные о количестве товаров на складе (см. рисунок ниже).

Отчет

Таблица

Строки

Товар

Колонки

Склад

Товар	Большой Количество Остаток	Малый Количество Остаток	Итого Количество Остаток
Bosch1234	56,00	1,00	57,00
Bosch15	1,00		1,00
Yeko345MO	89,00		89,00
Yeko67NE	6,00		6,00
Босоножки	100,00		100,00
Ботинки	12,00	1,00	13,00
Валенки	23,00		23,00
Кроссовки	3,00		3,00
Сапоги	15,00	2,00	17,00
Сланцы	2,00	2,00	4,00
Тапочки	7,00	5,00	12,00
Туфли	4,00	2,00	6,00
Итого	318,00	13,00	331,00

Рис. 146. Таблица в структуре отчета

Диаграмма – отображает данные в виде диаграммы. Чтобы добавить диаграмму в структуру, нужно выбрать пункт **Добавить – Новая диаграмма**. В зависимости от типа диаграммы необходимо добавить точки и серии. В качестве значений диаграммы будут использованы выбранные ресурсы.

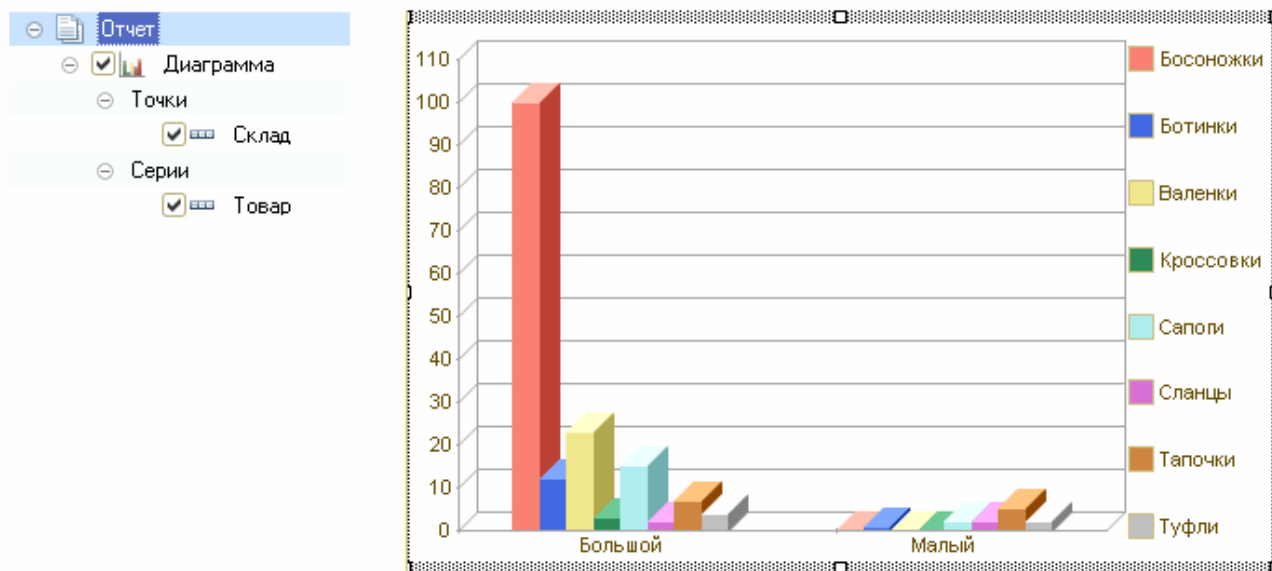


Рис. 147. Диаграмма в структуре отчета

Вложенный отчет – если при создании отчета на этапе конфигурирования для него был создан вложенный отчет, то он также может быть использован в структуре отчета в качестве элемента.

СОВЕТ. Наиболее часто используемые команды для работы со структурой отчета также можно вызывать из контекстного меню поля **Структуры**.

Чтобы удалить элемент из структуры, нужно выбрать его в поле структуры и нажать кнопку **Удалить элемент** или клавишу **Del**.

9.2.1. Настройка элемента структуры отчета

Для установки настроек для конкретного элемента структуры отчета нужно выбрать его в структуре и на закладке **Настройки** установить флажок **Текущий элемент имеет собственные настройки <имя настройки>** (для **детальной настройки элемента**).

Следует отметить, что возможности настройки зависят от типа элемента:

- для элемента типа **Группировка** можно настроить: **Поля группировки**, **Поля**, **Отбор**, **Сортировка**, **Условное оформление**, **Другие настройки**;
- для элементов типа **Таблица** и **Диаграмма** настраиваются: **Поля**, **Условное оформление**, **Другие настройки**.

ПРИМЕЧАНИЕ. Если в настройках отчета доступен вложенный отчет, то следует учитывать, что настройки отчетов разделяются.

9.2.2. Элементы настройки

9.2.2.1. Доступные поля

Список доступных полей (см. [рис. 148](#)) позволяет легко выбирать доступные поля и переносить их в списки полей группировки, выбора, отбора, сортировки с помощью стандартного механизма перетаскивания. Используя меню **Все действия** доступных полей, можно выбрать текущее или все доступные поля для того, чтобы поместить их в соответствующую коллекцию (команда **Выбрать все** выберет все поля, которые в дереве доступных полей располагаются на одном уровне с текущей строкой, при этом папки не выбираются). В списке доступных полей различаются поля, поля-ресурсы и папки полей.

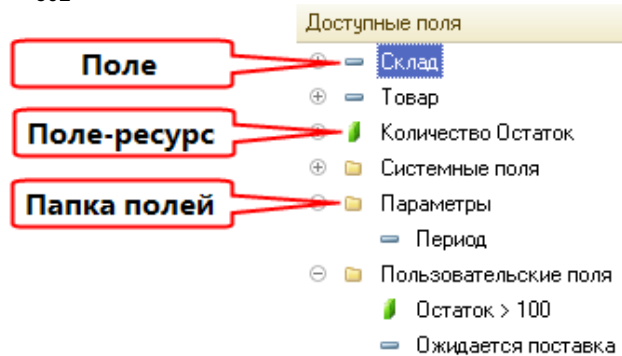


Рис. 148. Список доступных полей отчета

Для каждого числового ресурса в отчете могут быть использованы следующие вложенные поля:

- **% в строке или серии** – при выводе в таблице содержит отношение значения ресурса в текущей ячейке к значению ресурса в итоге по строке. При выводе вне таблицы выдает 100 %.
- **% в колонке или точке** – при выводе в таблице содержит отношение значения ресурса в текущей ячейке к значению ресурса в итоге по колонке. При выводе вне таблицы значение поля равно полю **% общий**.
- **% в группировке строки или серии** – при выводе в таблице содержит отношение ресурса в текущей ячейке к значению ресурса итога по текущей группировке по строке. При выводе вне таблицы выдает 100 %.
- **% в группировке колонки или точки** – при выводе в таблице содержит отношение ресурса в текущей ячейке к значению ресурса итога по текущей группировке по колонке, что равно полю **% в группировке**. При выводе вне таблицы равно полю **% в группировке**.
- **% в группе иерархии колонки или точки** – при выводе в таблице содержит отношение ресурса в текущей ячейке к значению ресурса итога на текущем уровне иерархии по текущей группировке по строке. При выводе вне таблицы содержит 100 %.
- **% в группе иерархии строки или серии** – при выводе в таблице содержит отношение ресурса в текущей ячейке к значению ресурса итога на текущем уровне иерархии по текущей группировке по колонке, что равно полю **% в группе иерархии**. При выводе вне таблицы равно полю **% в группе иерархии**.

При выводе этих полей в диаграмме в результате из заголовка поля убирается упоминание о строках и колонках. Например, поле **% в строке или серии** в результирующей диаграмме будет иметь заголовок **% в серии**.

При выводе этих полей вне диаграммы в результате из заголовка поля убирается упоминание о сериях и точках. Например, поле **% в строке или серии** в результирующей таблице будет иметь заголовок **% в строке**.

Для полей типа **Дата** доступен список дочерних полей, разбитый по группам: **Даты начала**, **Даты конца**, **Части дат**. С их помощью можно уточнить дату и время. Например, **Начало дня**, **Начало недели**, **Конец недели** и т.д.

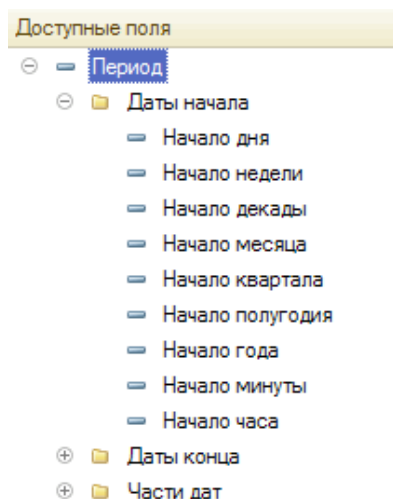


Рис. 149. Уточнение даты

Для полей, в составе которых есть дата:

- **Начало дня** – Возвращает дату и время начала дня.
- **Начало недели** – Возвращает дату и время начала недели.
- **Начало декады** – Возвращает дату и время начала декады.
- **Начало месяца** – Возвращает дату и время начала месяца.
- **Начало квартала** – Возвращает дату и время начала квартала.
- **Начало полугодия** – Возвращает дату и время начала полугодия.
- **Начало года** – Возвращает дату и время начала года.
- **Конец дня** – Возвращает дату и время конца дня.
- **Конец недели** – Возвращает дату и время конца недели.
- **Конец декады** – Возвращает дату и время декады.
- **Конец месяца** – Возвращает дату и время месяца.
- **Конец квартала** – Возвращает дату и время квартала.
- **Конец полугодия** – Возвращает дату и время полугодия.
- **Конец года** – Возвращает дату и время конца года.
- **День** – Возвращает номер дня месяца.
- **День недели** – Возвращает номер дня недели (неделя начинается с понедельника).
- **Название дня недели** – Возвращает представление дня недели.
- **День года** – Возвращает номер дня года.
- **Неделя года** – Возвращает номер недели года.
- **Месяц** – Возвращает номер месяца года.
- **Название месяца** – Возвращает представление месяца.
- **Квартал** – Возвращает номер квартала.
- **Год** – Возвращает четырехзначное значение года.

Для дат, в составе которых указано время:

- **Начало минуты** – Возвращает дату и время начала минуты.
- **Начало часа** – Возвращает дату и время начала часа.
- **Конец минуты** – Возвращает дату и время конца минуты.
- **Конец часа** – Возвращает дату и время конца часа.
- **Минута** – Возвращает номер минуты.
- **Час** – Возвращает номер часа.

9.2.2.2. Параметры

Если у отчета имеются параметры, доступные для настройки, они отобразятся в списке доступных полей, в папке **Параметры**, и также станут доступны для включения/исключения из отчета и выбора значения (если параметру было задано несколько значений) на закладке **Параметры**. Например, если используется период, за который выполняется отчет, значение периода должно быть установлено пользователем.

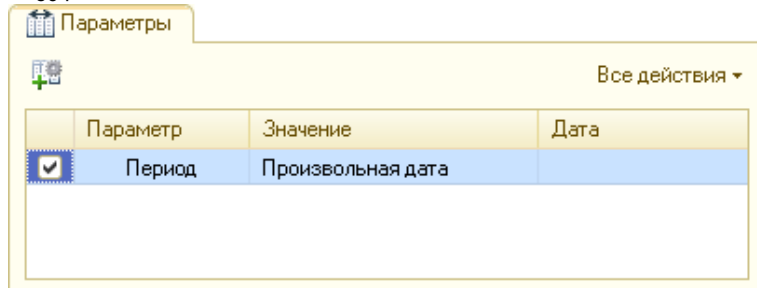


Рис. 150. Параметры отчета в форме настройки варианта

В различных настройках отчета параметры можно использовать как поля. В настройках, где будут использоваться эти поля, при выполнении отчета будут подставляться значения параметров.

Например, если в отчете есть параметр [Организация](#), то в отборе будет доступно поле [Параметры.Организация](#). При выборе этого поля в качестве значения отбора будет установлено значение параметра [Организация](#).

Для параметра типа [Дата](#) в качестве значения можно использовать не только конкретную дату, но и одну из стандартных дат: начало этого дня, начало следующего дня и т. д.

9.2.2.3. Пользовательские поля

На закладке [Пользовательские поля](#) можно создавать поля для использования в отчете. Для того чтобы вывести пользовательское поле в отчет, нужно добавить его в список выбранных полей. При этом возможность расположения пользовательского поля в элементе отчета определяется системой автоматически.

Пользовательские поля могут быть двух видов: поле-выбор или поле-выражение.

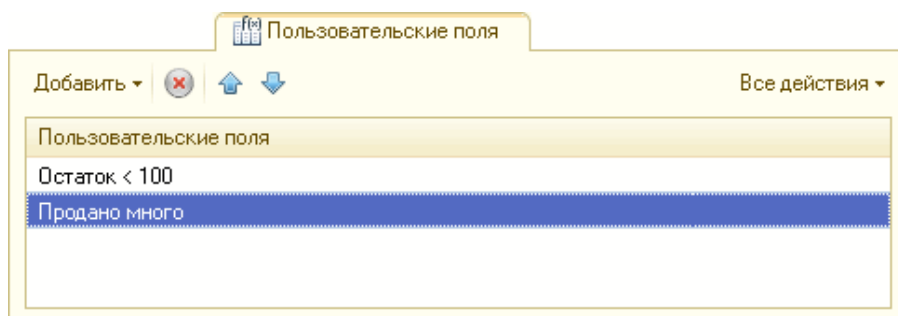


Рис. 151. Пользовательские поля в настройках варианта отчета

Полю-выбору можно задать несколько значений. Для вывода в отчет будет выбрано первое значение, для которого выполнится условие отбора. Нужно задать заголовок и выражение отбора, значение и, при необходимости, краткое представление отбора.

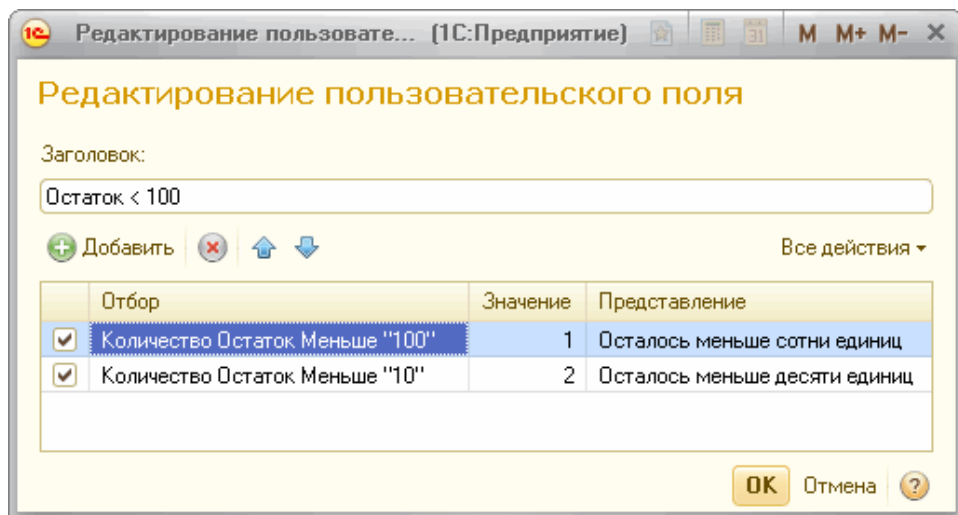


Рис. 152. Настройка отчета. Пользовательское поле-выбор

Условия отбора задаются в отдельном окне и добавляются нажатием кнопки [Добавить](#). После этого следует выбрать поле из списка доступных, вид сравнения и установить значение для сравнения.

Для того чтобы настроить последовательность элементов отбора, нужно использовать стрелки командной панели или же переместить элемент на требуемую строку с помощью мыши.

Пользовательское **поле-выражение** будет выводиться в отчете как результирующее значение выражений. Выражения составляются при помощи доступных полей отчета и выражений языка системы компоновки данных. Пользовательские поля могут использоваться в любом месте отчета. Пользовательские поля-ресурсы не могут использоваться в качестве поля группировки.

В диалоге **Редактирование пользовательского поля** следует задать заголовок и выражения для детальных и итоговых записей.

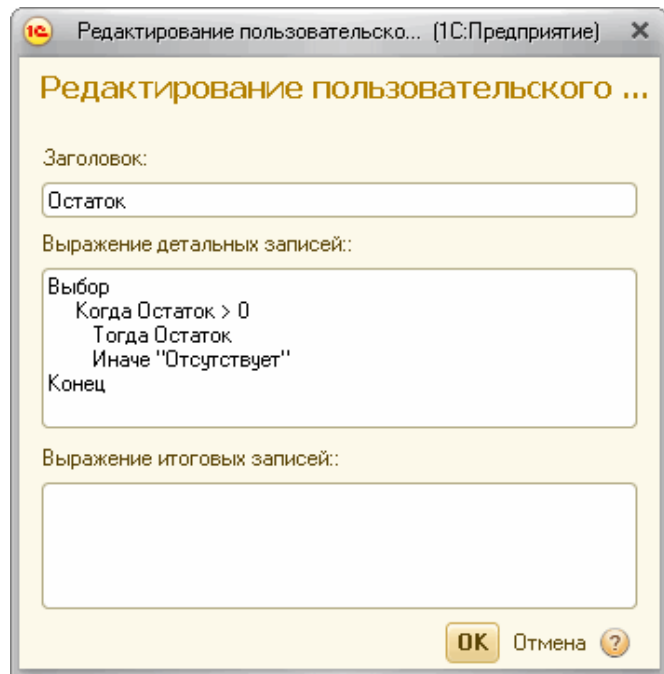


Рис. 153. Пользовательское поле-выражение

Названия полей, содержащие пробелы и/или спецсимволы, должны заключаться в квадратные скобки [].

Выражение для итоговых записей должно быть записано с помощью агрегатных функций, например: **Сумма (Поле) * 2**.

Подробнее о языке выражений компоновки данных см. Приложение 1 «Язык выражений системы компоновки данных», доступное в электронной версии документации.

9.2.2.4. Группируемые поля

На закладке **Группируемые поля** выбираются поля, по которым будет проводиться группировка, и тип группировки.

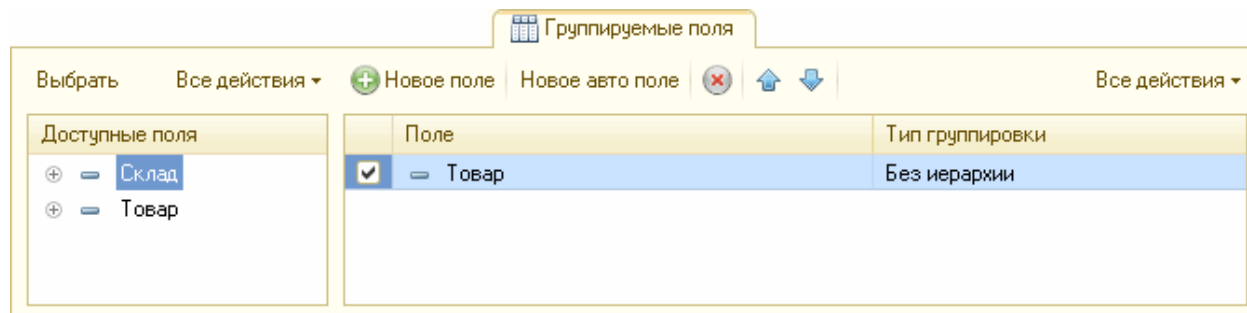


Рис. 154. Группируемые поля в элементе отчета

Группировка может иметь один из следующих типов:

- **Без иерархии** – в группировку попадут только неиерархические записи (элементы);
- **Иерархия** – в группировку попадут иерархические записи и неиерархические записи (элементы и группы);

- **Только иерархия** – в группировке будут выводиться только иерархические записи (группы).

С помощью меню можно добавить новое поле или новое автополе.

Автополе группировки

Группируемое **автополе** представляет собой набор полей, формируемый системой по следующему принципу. Выбираются используемые в отчете выбранные поля, которые:

- доступны для использования в полях группировки,
- не являются ресурсами,
- не являются реквизитами других выбранных полей,
- не являются реквизитами существующих полей группировки.

Чтобы просмотреть набор полей, в который будет преобразовано автополе группировки при выполнении отчета, в настройках отчета нужно вызвать команду [Все действия – Развернуть](#).

Если поле уже включено в данные поля группировки, повторно оно не добавляется.

Если допустимые типы поля группировки включают в себя дату, доступна настройка дополнения периода (для вывода в отчет дат, которые не попали в результат). Например, чтобы в отчете выводились данные для всех недель, выбирается тип дополнения [Неделя](#) и устанавливается начальная и конечная даты периода, как показано на рисунке ниже.

	Поле		Тип дополнения	
	Начальная дата периода		Конечная дата периода	
<input checked="" type="checkbox"/>	=	Период, день	Неделя	
		21.11.2008 0:00:00	12.11.2009 0:00:00	

Рис. 155. Настройка дополнения группировки по периоду

Проиллюстрируем на примере. Предположим, в отчете [Взаиморасчеты](#) отображаются данные о взаиморасчетах с магазинами «Мясная лавка» и «Продукты» за период с 1 января 2009 по 31 декабря 2009 года.

Параметры: Период = 01.01.2009 - 31.12.2009
 Отбор: Контрагент В списке "Магазин "Мясная лавка"; Магазин "Продукты"

Контрагент	Сумма Начальный остаток	Сумма Конечный остаток
Магазин "Мясная лавка"		71 562,86
01.01.2009 0:00:00		73 731,43
01.04.2009 0:00:00	73 731,43	71 562,86
01.12.2009 0:00:00	71 562,86	71 562,86
Магазин "Продукты"	2 702,85	97 609,99
01.01.2009 0:00:00	2 702,85	62 302,85
01.02.2009 0:00:00	62 302,85	72 609,99
01.04.2009 0:00:00	72 609,99	97 609,99
01.12.2009 0:00:00	97 609,99	97 609,99
Итого	2 702,85	169 172,85

Рис. 156. Отчет без дополнения

При этом необходимо вывести данные поквартально. Для этого группировке [Период, месяц](#) на закладке [Группировка](#) укажем [Дополнение](#).

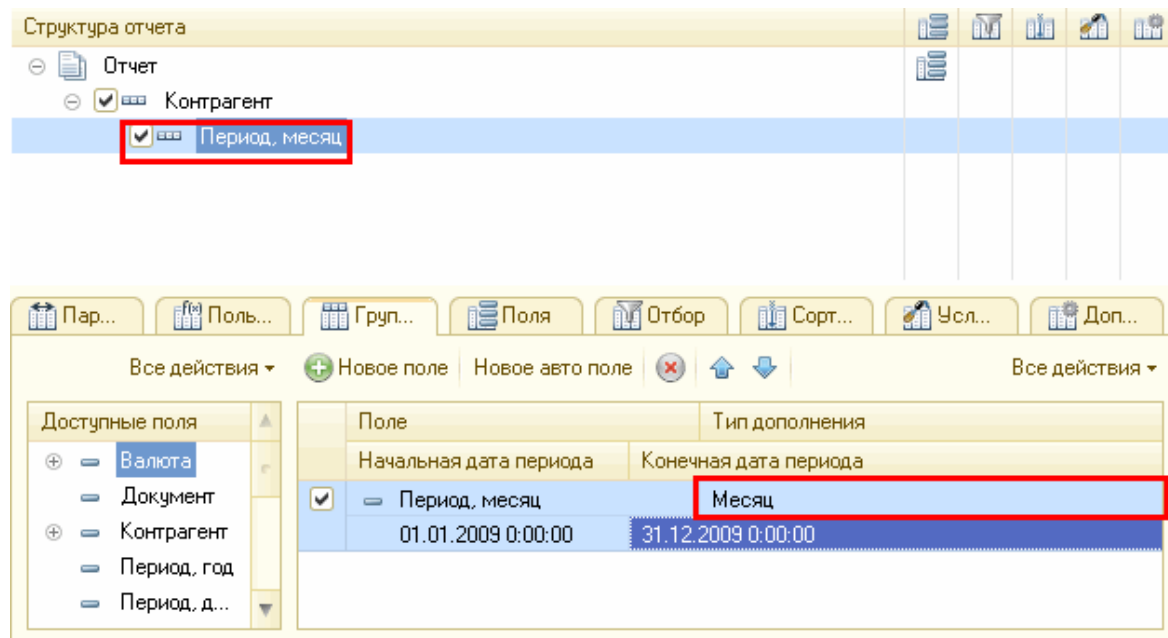


Рис. 157. Настройка дополнения в отчете

В результате отчет [Взаиморасчеты](#) примет вид:

Контрагент	Сумма Начальный остаток	Сумма Конечный остаток
Период, месяц		
Магазин "Мясная лавка"		71 562,86
01.01.2009 0:00:00		73 731,43
01.02.2009 0:00:00	73 731,43	73 731,43
01.03.2009 0:00:00	73 731,43	73 731,43
01.04.2009 0:00:00	73 731,43	71 562,86
01.05.2009 0:00:00	71 562,86	71 562,86
01.06.2009 0:00:00	71 562,86	71 562,86
01.07.2009 0:00:00	71 562,86	71 562,86
01.08.2009 0:00:00	71 562,86	71 562,86
01.09.2009 0:00:00	71 562,86	71 562,86
01.10.2009 0:00:00	71 562,86	71 562,86
01.11.2009 0:00:00	71 562,86	71 562,86
01.12.2009 0:00:00	71 562,86	71 562,86
Магазин "Продукты"	2 702,85	97 609,99
01.01.2009 0:00:00	2 702,85	62 302,85
01.02.2009 0:00:00	62 302,85	72 609,99
01.03.2009 0:00:00	72 609,99	72 609,99
01.04.2009 0:00:00	72 609,99	97 609,99
01.05.2009 0:00:00	97 609,99	97 609,99
01.06.2009 0:00:00	97 609,99	97 609,99
01.07.2009 0:00:00	97 609,99	97 609,99
01.08.2009 0:00:00	97 609,99	97 609,99
01.09.2009 0:00:00	97 609,99	97 609,99
01.10.2009 0:00:00	97 609,99	97 609,99
01.11.2009 0:00:00	97 609,99	97 609,99
01.12.2009 0:00:00	97 609,99	97 609,99
Итого	2 702,85	169 172,85

Рис. 158. Отчет с дополнением

9.2.2.5. Поля

На закладке [Поля](#) выбираются поля, которые будут отображены в отчете. Если выбранных полей нет, в отчет выводится пустой элемент. Имеется возможность с помощью меню или контекстного меню добавить новое поле, новую группу полей, новое автополе. Группе полей можно дать свое название и также можно управлять их размещением в текущем элементе отчета.

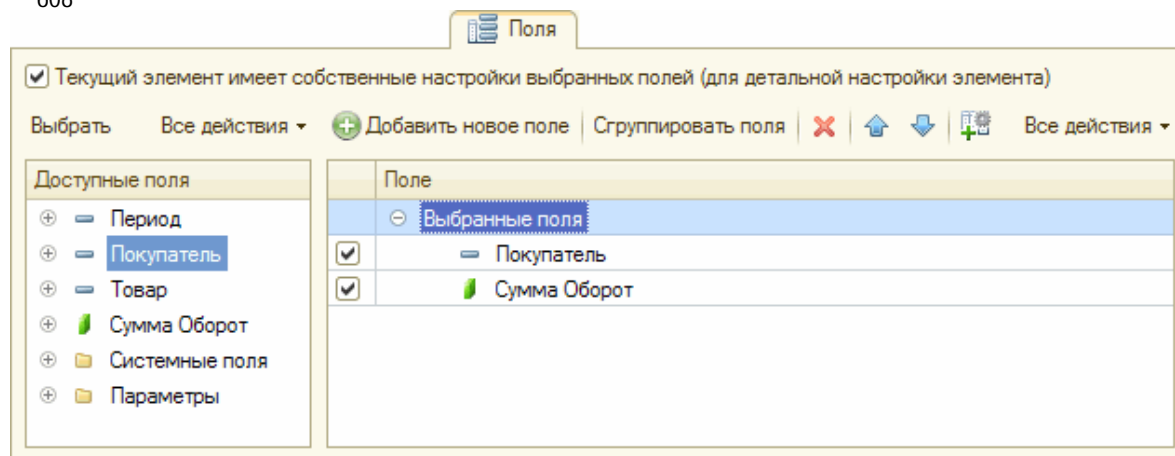


Рис. 159. Настройка отчета. Поля

9.2.2.6. Автополе выбора

В процессе настройки система позволяет использовать автоматические наборы полей в качестве полей. При создании элементов структуры отчета система автоматически добавляет поле **<Авто>** в качестве поля выбора. **Автополе выбора** представляет собой набор полей, в который оно будет преобразовано при выполнении отчета. Чтобы просмотреть набор полей, нужно использовать команду **Все действия – Развернуть**. Также если нужно добавить автополе, следует использовать кнопку **Все действия Новое авто поле**.

Состав набора полей зависит от того, какому элементу структуры принадлежит разворачиваемое автополе и в какой части структуры этот элемент располагается. Для каждого элемента система обходит все родительские элементы структуры отчета и из выбранных полей этих элементов отбирает ресурсы и поля по следующим правилам:

- Для **группировки** и **группировки таблицы** на место автополя подставляются все используемые поля этой группировки, которые доступны для использования в выбранных полях; поля, которые являются реквизитами ее полей группировки, и ресурсы родительских элементов.

ВНИМАНИЕ! Система учитывает при обходе только группировки с типом **Без иерархии** или **Иерархия**.

- Для **группировки диаграммы** ресурсы не выбираются, а обходятся все родительские элементы структуры настроек, и из выбранных полей этих элементов выбираются поля группировок, если по данному полю была задана группировка с типом **Только иерархия**.
- Для **группировок типа Детальные записи (группировка, группировка таблицы, группировка диаграммы)** из основных выбранных полей настроек, которым принадлежит группировка, выбираются все используемые поля, кроме полей, участвовавших в вышестоящих группировках, и реквизитов этих полей. Если же такая группировка имеет тип **Только иерархия**, то ее поля и реквизиты будут использоваться системой при формировании набора полей выбора. Для группировки диаграммы ресурсы также не выбираются.
- Для **диаграммы** автополе выбора заменяется всеми ресурсами, указанными для диаграммы. В зависимости от настроек прикладного решения автополе выбора может заменяться ресурсом, первым из встреченных при описанном выше обходе.
- Для **таблицы** автополе выбора преобразовывается в набор ресурсов, используемых родительскими элементами.

ПРИМЕЧАНИЕ. Если поле уже включено в данные выбранные поля, повторно оно не добавляется.

При этом поля добавляются в набор в следующем порядке: вначале поля собственных полей группировки (для группировок), потом поля из глобальных настроек (для группировок типа **Детальные записи**), и самыми последними – ресурсы и поля из родительских элементов.

Выбранные поля можно объединять в группы, которые добавляются командой **Сгруппировать** (чтобы команда была доступна, все выделенные строки таблицы должны иметь одного родителя) и удаляются, сохраняя вложенные поля, командой **Разгруппировать**. Для группы можно задавать расположение внутри текущего элемента структуры отчета. Для этого следует выбрать необходимый вид расположения из списка в колонке **Расположение**.

9.2.2.7. Отбор

На закладке [Отбор](#) выбираются поля для фильтрации записей результата отчета. С помощью меню, контекстного меню или двойным щелчком мыши по выбранному полю из доступных полей можно добавить элемент или группу элементов. На этой странице можно выбрать отбор из заданных разработчиком по представлению, если таковое устанавливалось для отчета, или добавить другой.

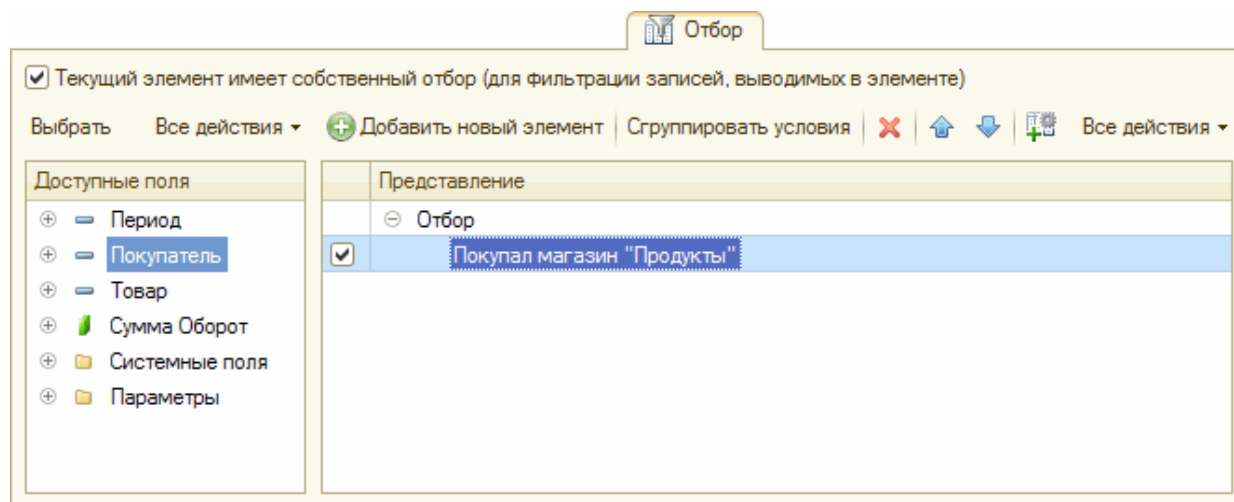


Рис. 160. Настройка элемента отчета. Представление отбора

При вызове команды [Все действия – Подробно](#) на закладке отобразятся условия отбора вместе с представлением.

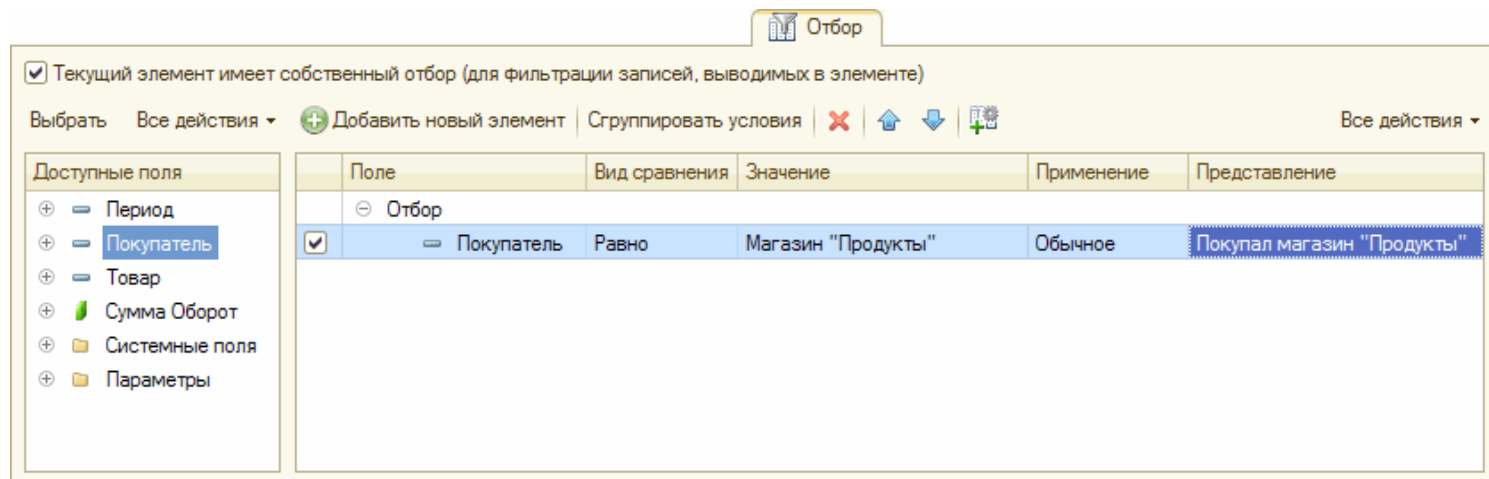


Рис. 161. Настройка элемента отчета. Отбор

В колонках этой закладки элементам можно установить значения, по которым будет проводиться отбор записей результата отчета.

Представление. С помощью команды контекстного меню [Установить представление](#) созданному элементу или группе отбора можно задать представление, которое будет показывать информацию об элементе отбора в списке, если кнопка [Подробно](#) командной панели отжата. Если для отбора не было задано представления, то его подробное и непдробное представления на закладке [Отбор](#) не отличаются.

Применение. Для группировки, группировки в таблице, группировки в диаграмме для элементов отбора доступно еще одно свойство – [Применение](#).

Поле	Вид сравнения	Значение	Применение
<ul style="list-style-type: none"> Отбор <input checked="" type="checkbox"/> Покупатель 	Равно	Магазин "Продукты"	Обычное

Рис. 162. Настройка применения для элемента отбора

Если свойство [Применение](#) установлено в значение [После группировки или для иерархии](#), то условие отбора проверяется после построения иерархии. Значение [Только иерархия](#) указывает, что условие отбора проверяется только для иерархических записей. Отфильтрованные записи продолжают влиять на итоговые значения ресурсов вышестоящих иерархических записей. Вложенные группировки записей, отфильтрованных при помощи этих свойств, также будут выводиться в отчет.

Виды сравнения

Данный раздел содержит описание видов сравнения, предоставляемых системой для отборов.

Равно – в результат попадут записи, для которых значение поля, указанного в качестве левого значения, равно правому значению.

Не равно – в результат попадут записи, для которых значение поля, указанного в качестве левого значения, не равно правому значению.

Меньше – в результат попадут записи, для которых значение поля, указанного в качестве левого значения, меньше правого значения.

Меньше или равно – в результат попадут записи, для которых значение поля, указанного в качестве левого значения, меньше или равно правому значению.

Больше – в результат попадут записи, для которых значение поля, указанного в качестве левого значения, больше правого значения.

Больше или равно – в результат попадут записи, для которых значение поля, указанного в качестве левого значения, больше или равно правому значению.

В списке – в результат попадут записи, в которых есть поля со значением, присутствующим в списке. Список значений устанавливается в отдельном окне, открываемом при выборе колонки **Левое значение**.

В группе из списка – в результат попадут записи, значения которых относятся к группам, присутствующим в списке, либо записи, которые указаны в списке. Список групп и элементов устанавливается в отдельном окне, открываемом при выборе.

В группе – в результирующий документ попадут записи, поле из левого значения которых находится в указанной группе элементов списка.

Не в списке – в результат попадут записи, в которых нет полей со значением, присутствующим в списке. Список значений устанавливается в отдельном окне, открываемом при выборе колонки **Левое значение**.

Не в группе из списка – в результирующий документ попадут записи, поле из левого значения которых не находится в указанной группе элементов списка.

Содержит – в результат попадут записи, содержащие подстроку, используемую в качестве значения.

Не содержит – в результат попадут записи, которые не содержат подстроки, используемой в качестве значения.

Заполнено – в результат попадут записи, у которых заполнено заданное поле.

Не заполнено – в результат попадут записи, у которых заданное поле не заполнено.

Начинается с – в результат попадут записи, у которых значение начинается на строку, заданную в условии сравнения.

Не начинается с – в результат попадут записи, у которых значение не начинается на строку, заданную в условии сравнения.

Соответствует шаблону – в результат попадут записи, значение которых удовлетворяет маске, заданной в условии сравнения.

Не соответствует шаблону – в результат попадут записи, значение которых не удовлетворяет маске, заданной в условии сравнения.

В строке шаблонов используется следующий синтаксис:

- / - следующий символ нужно интерпретировать как обычный символ;
- % - процент: последовательность, содержащая ноль и более произвольных символов;
- _ - подчеркивание: один произвольный символ.

Объединение условий отбора

Условия отборов можно объединять в логические группы по **И**, **НЕ** и группы по **ИЛИ**. Чтобы объединить условия, нужно выбрать их и нажать кнопку **Сгруппировать условия**:

- если отборы находятся в группе по **И**, то в отчет будут выводиться данные, для которых выполняются все условия, находящиеся в группе;
- если отборы объединены в группу по **НЕ**, то в результирующий документ не попадут записи, удовлетворяющие всем условиям группы;
- если отборы находятся в группе по **ИЛИ**, то данные будут выводиться при выполнении хотя бы одного условия. С помощью контекстного меню группам можно задавать представления; если оно есть у группы, то в кратком режиме вложенные элементы группы показываться не будут.

9.2.2.8. Сортировка

На закладке **Сортировка** выбираются поля, по которым результат выполнения отчета будет отсортирован. Возможно добавление как элемента порядка, так и автоэлемента порядка. При выполнении отчета система преобразует (разворачивает) автоэлемент порядка в поля, по которым и проведет сортировку.

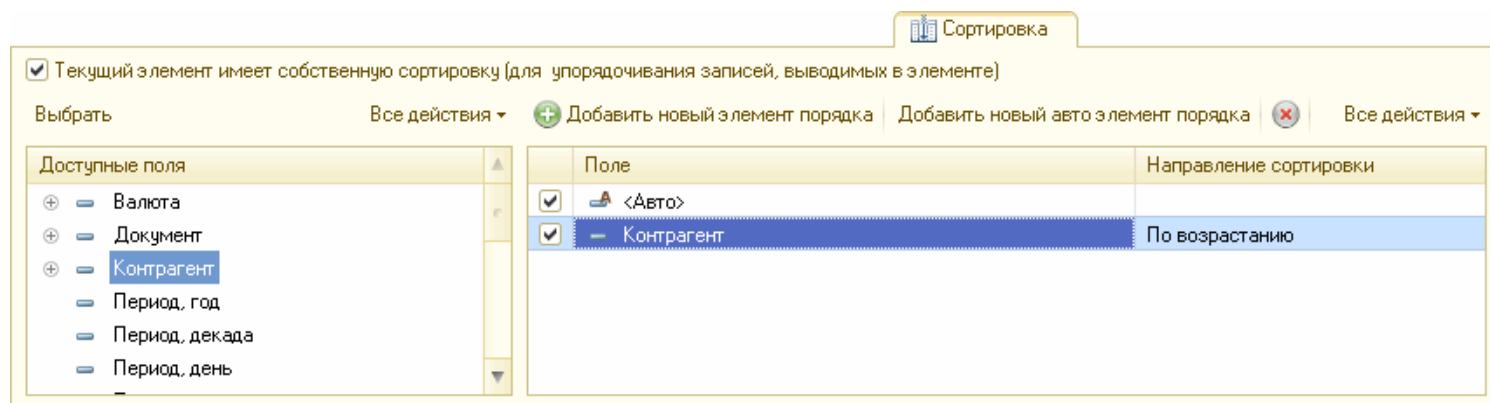


Рис. 163. Поля сортировки для элемента отчета

Для **автоэлемента** порядка в меню **Все действия** есть команда **Развернуть**, которая позволяет просмотреть, в какой набор полей это поле будет преобразовано при выполнении отчета. При развороте автоэлемента (порядка) из глобального упорядочивания безусловно будут добавлены поля-ресурсы, а из полей нересурсов в порядок будут добавлены поля, являющиеся реквизитами поля группировки, и само поле группировки (для детальных записей будут занесены все поля). Поля группировки, которые не были указаны в глобальном упорядочивании, попадут в конец порядка.

9.2.2.9. Условное оформление

На закладке **Условное оформление** можно установить оформление для различных элементов отчета в зависимости от значений данных, которые выводятся в отчет. Например, выделить цветом отрицательные значения и т. д. Таким образом может быть оформлено несколько элементов структуры. При этом для каждого элемента задаются области, которые будут оформлены.

Условное оформление может состоять из нескольких элементов. Каждый элемент описывает область, которую нужно оформить.

Область представляет собой список полей, выбираемых из списка доступных. Если область не задана, то условное оформление применится ко всему элементу отчета.

Для каждой области могут задаваться условия, при выполнении которых к области будет применяться выбранное оформление.

Условия отбора полей указываются в отдельном окне, похожем на окно отбора пользовательского поля выбора. Подробнее про отборы можно прочитать [здесь](#).

Если по некоторым условиям к области применяются два разных оформления, то в итоге выбирается последнее из них в списке элементов условного оформления.

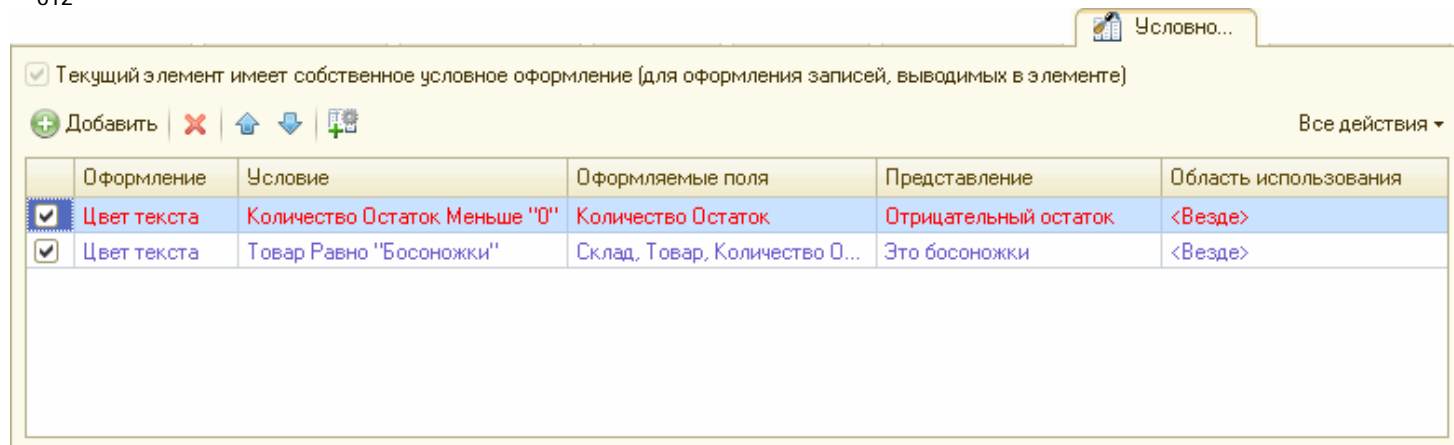


Рис. 164. Список элементов условного оформления в режиме «Подробно»

В колонке **Представление** задается обозначение условного оформления области, с которым оно будет показано в списке условных оформлений, если кнопка напротив команды **Подробно** меню **Все действия** отжата.

Параметры условного оформления задаются в отдельном окне. Можно выбрать цвета фона, текста, границы ячейки, стиль границ ячейки, шрифт текста в ячейке и др.

- **Цвет фона** – устанавливает цвет фона;
- **Цвет текста** – устанавливает цвет текста для текстовых данных;
- **Цвет в диаграмме** – устанавливает цвет серии данных в диаграмме;
- **Цвет границы** – цвет, которым будет отображаться граница элемента данных;
- **Стиль границы** – стиль, которым будет отображаться граница элементов. Система позволяет настроить вид для каждой границы в отдельности;
- **Шрифт** – устанавливает шрифт, которым будут отображаться текстовые данные;
- **Отступ** – устанавливает смещение на заданное количество символов;
- **Автоотступ** – если указано значение больше 0, то при отображении группировок отступ будет определяться как произведение уровня группировки и значения, указанного в свойстве;
- **Горизонтальное положение** – в данном свойстве выбирается выравнивание текста по горизонтали;
- **Вертикальное положение** – в данном свойстве выбирается выравнивание текста по вертикали;
- **Размещение** – устанавливает способ отображения текста, если он не помещается в ячейке целиком;
- **Ориентация текста** – число, устанавливает угол наклона текста в ячейках группировок и таблиц;
- **Формат** – устанавливает формат отображения данных;
- **Выделять отрицательные** – если свойство установлено в значение **Да**, то в результате отчета отрицательные значения будут отображаться особым цветом;
- **Отметка незаполненного** – если свойство установлено в значение **Да**, то в результате отчета незаполненные значения будут отображаться особым образом;
- **Минимальная ширина** – число, устанавливает минимальное количество символов для отображения данных по ширине;
- **Максимальная ширина** – число, устанавливает максимальное количество символов для отображения данных по ширине;
- **Минимальная высота** – число, устанавливает минимальное количество символов для отображения данных по высоте;
- **Максимальная высота** – число, устанавливает максимальное количество символов для отображения данных по высоте;

по высоте;

- **Текст** – произвольный текст, описывающий данные.

Чтобы установить область применения элемента оформления, при двойном щелчке в диалоге редактирования элемента условного оформления в колонке **Область использования** откроется закладка **Дополнительно**. Если флажок напротив области установлен, то элемент оформления будет использован при отображении этой области.

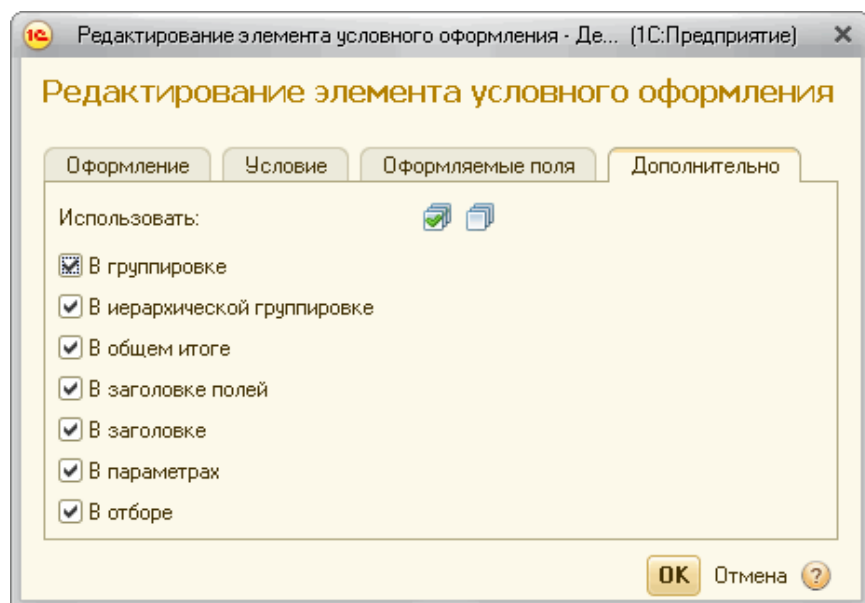


Рис. 165. Установка области использования элемента оформления

По умолчанию все флажки установлены. Для быстрого редактирования можно использовать кнопки **Установить все** и **Сбросить все**.

ПРИМЕЧАНИЕ. Возможность установки области использования элемента условного редактирования недоступна при настройке динамического списка.

9.2.2.10. Дополнительные настройки

На закладке **Дополнительные настройки** для выбранного элемента структуры задаются параметры вывода. Например, для группировки содержимое закладки будет иметь вид:

Параметр	Значение
<input checked="" type="checkbox"/> Макет оформления	Основной
<input type="checkbox"/> Расположение итогов	Авто
<input type="checkbox"/> Расположение полей группировок	Вместе
<input type="checkbox"/> Расположение группировок	Начало
<input type="checkbox"/> Расположение реквизитов	Вместе с владельцем
<input type="checkbox"/> Расположение ресурсов	Горизонтально
<input type="checkbox"/> Расположение общих итогов по го...	Авто
<input type="checkbox"/> Расположение общих итогов по ве...	Авто
<input type="checkbox"/> Тип заголовка полей	Авто
<input type="checkbox"/> ⊕ Тип диаграммы	Гистограмма объемная
<input type="checkbox"/> Выводить заголовок	Авто
<input type="checkbox"/> Заголовок	
<input type="checkbox"/> Выводить параметры данных	Авто
<input type="checkbox"/> Выводить отбор	Авто

Рис. 166. Дополнительные настройки элемента отчета

ПРИМЕЧАНИЕ 1. При компоновке отчета система выдаст ошибку, если во вложенном отчете используется поле верхнего отчета, значение которого невозможно определить.

ПРИМЕЧАНИЕ 2. При размещении диаграммы в структуре отчета следует иметь в виду, что при выборе вида

диаграммы [Круговая](#), если не заданы серии, но заданы точки, в качестве серий системой будут автоматически использоваться точки. А при выборе вида диаграммы [График](#), если не заданы точки, но заданы серии, в качестве точек будут использоваться серии.

9.3. Выбор элементов настроек

Чтобы указать, что элемент настройки доступен на форме настроек отчета, в настройке элемента используется команда [Свойства элемента пользовательских настроек](#).

В форме настройки пользовательского элемента можно установить признак того, что элемент является пользовательским, а также указать представление, которое будет использоваться для элемента, и режим его редактирования (обычный, быстрый доступ, недоступный), см. [рис. 167](#).

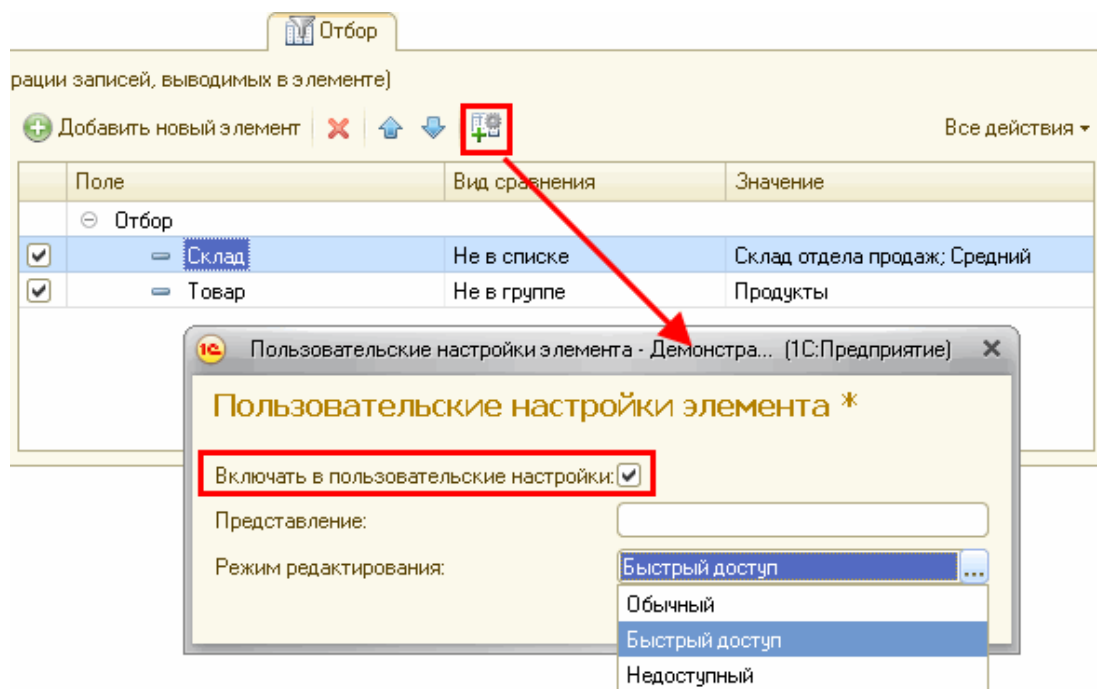


Рис. 167. Добавление элемента пользовательских настроек

В списке структуры настроек команда [Свойства элемента пользовательских настроек](#) позволяет настраивать пользовательские настройки для текущего элемента структуры.

Для каждого элемента структуры определен свой состав настраиваемых элементов.

Объект	Настраиваемые элементы
Отчет	Выбранные поля, порядок, отбор, условное оформление, состав группировок
Группировка/группировка таблицы/группировка диаграммы	Группировка, выбранные поля, отбор, порядок, условное оформление, состав вложенных группировок
Диаграмма	Диаграмма, выбранные поля, условное оформление, состав группировок серий, состав группировок точек
Таблица	Таблица, выбранные поля, условное оформление, состав группировок строк, состав группировок колонок
Вложенная схема	Вложенный отчет, выбранные поля, отбор, порядок, условное оформление, состав группировок

В зависимости от того, где происходит вызов, команда [Свойства элемента пользовательских настроек](#) позволяет изменять различные настройки:

- список отбора – настройки для текущего элемента/групп отбора;
- список параметров вывода и параметров данных – настройки для текущего параметра;
- список условного оформления – настройки для текущего элемента условного оформления.

ПРИМЕЧАНИЕ. Если настройка включена в пользовательские настройки, то при выполнении отчета ее значение будет получаться из пользовательских настроек, а не из варианта отчета.

При редактировании варианта отчета изменение настройки, значение которой было ранее сохранено в пользовательских настройках, не повлияет на результат отчета, поскольку значение настройки будет получено из ранее сохраненной пользовательской настройки.

Кроме того, вызванная из командной панели структуры отчета команда [Предварительный просмотр пользовательских настроек](#) (см. рисунок) позволяет открыть модальную форму, в которой будут отображаться настройки со своими значениями по умолчанию.

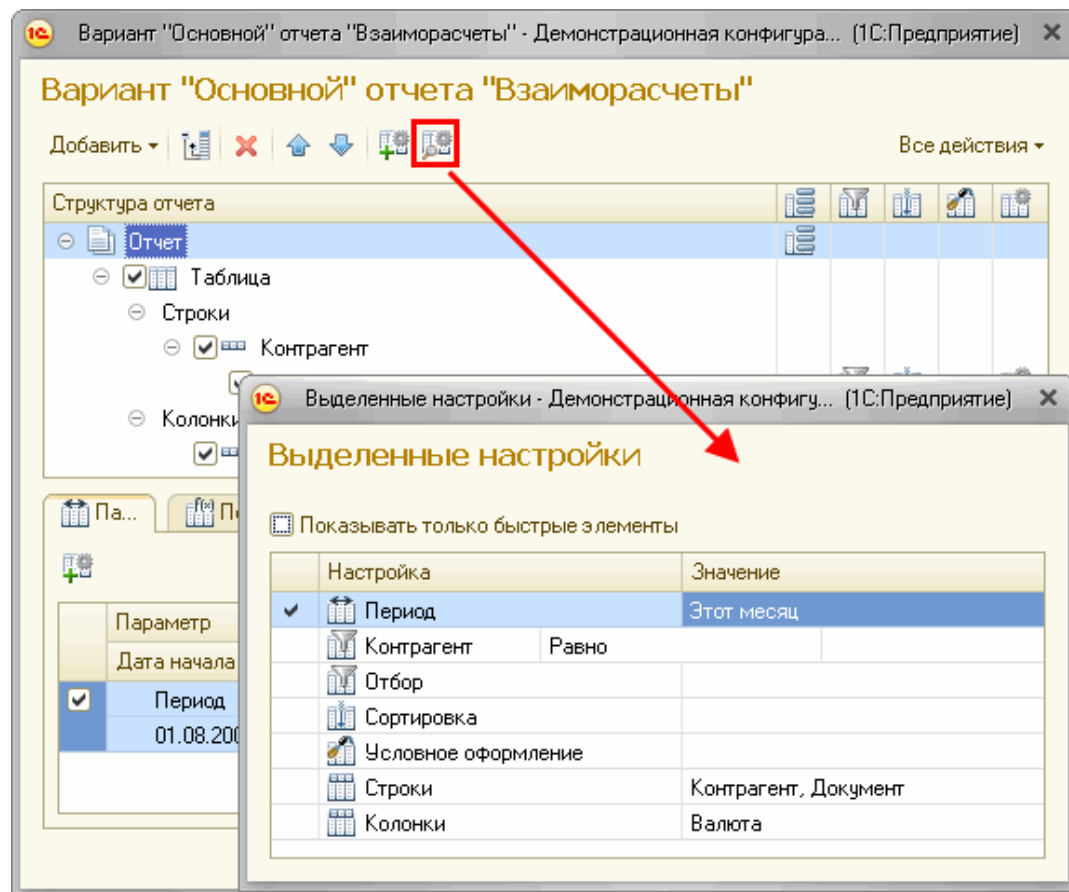


Рис. 168. Просмотр пользовательских настроек

Глава 10. Сервисные возможности

Система «1С:Предприятие» предоставляет сервисные механизмы для навигации по приложению: механизм ссылок, история работы, избранное.

Также пользователю доступна возможность использования встроенных калькулятора, календаря, механизма для сравнения файлов и т. д.

Данная глава описывает работу с перечисленными механизмами.

10.1. Ссылки

В системе имеется возможность получить текстовую ссылку на любой раздел интерфейса программы, отчет, обработку и на объекты информационной базы (документы, элементы списков и т. д.).

ПРИМЕЧАНИЕ. При работе с закладками и в интерфейсе [Такси](#) ссылки на разделы системой не выдаются.

Полученную ссылку можно сохранить и использовать в дальнейшем для перехода по ней. Так как ссылки являются текстовыми, то их можно отправлять другим пользователям, например, по электронной почте.

Для получения ссылки следует использовать пункт главного меню [Сервис – Получить ссылку](#) или нажать кнопку [Получить ссылку](#) области системных команд.

Рассмотрим пример получения ссылки на документ [Заказ 00000020](#):

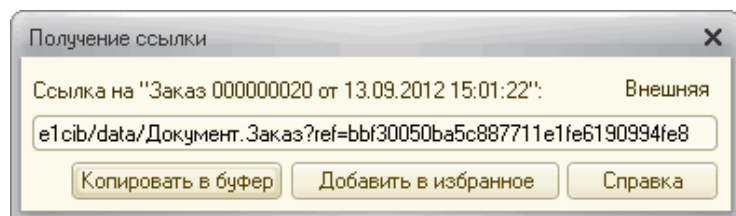


Рис. 169. Получение ссылки на документ «Заказ»

Чтобы получить ссылку для использования в адресной строке браузера (для веб-клиента), следует нажать кнопку [Внешняя](#).

Чтобы поместить ссылку в буфер обмена, следует нажать кнопку [Копировать в буфер](#).

Для добавления ссылки в список избранных используется кнопка [Добавить в избранное](#).

Для перехода по ссылке следует использовать команду [Сервис – Перейти по ссылке](#), в открывшемся диалоге указать сохраненный ранее текст ссылки и нажать кнопку [Перейти](#) или клавишу [Enter](#). Для того чтобы отличать ссылки «1С:Предприятия» от других (например, [http](#) или [file](#)), используется идентификатор схемы [e1c](#). Если в диалог перехода по ссылке передается внешняя ссылка «1С:Предприятия», начало которой не совпадает с навигационной ссылкой информационной базы, то при переходе по ссылке выполняется попытка перейти по внутренней части, указанной после идентификатора [e1c](#). Если переход не удастся, то выполняется переход по переданной внешней ссылке.

При переходе по ссылке пробельные символы в начале и в конце строки не учитываются.

10.2. Избранное

В «1С:Предприятии» **избранное** – это список ссылок, выбранных самим пользователем для быстрого перехода к тем или иным разделам конфигурации, формам объектов информационной базы, а также формам отчетов и обработок.

Чтобы добавить ссылку на объект в список избранных, можно воспользоваться одним из следующих способов:

- выбрать пункт в области системных команд [Избранное – Добавить в избранное](#);
- выбрать пункт главного меню [Сервис – Добавить в избранное](#);
- при получении ссылки следует нажать кнопку [Добавить в избранное](#) (подробнее о работе со ссылками см. раздел [«Ссылки» выше](#)).

При добавлении ссылки в избранное система отобразит оповещение.

Для открытия списка избранных в области системных команд следует нажать кнопку [Избранное](#).

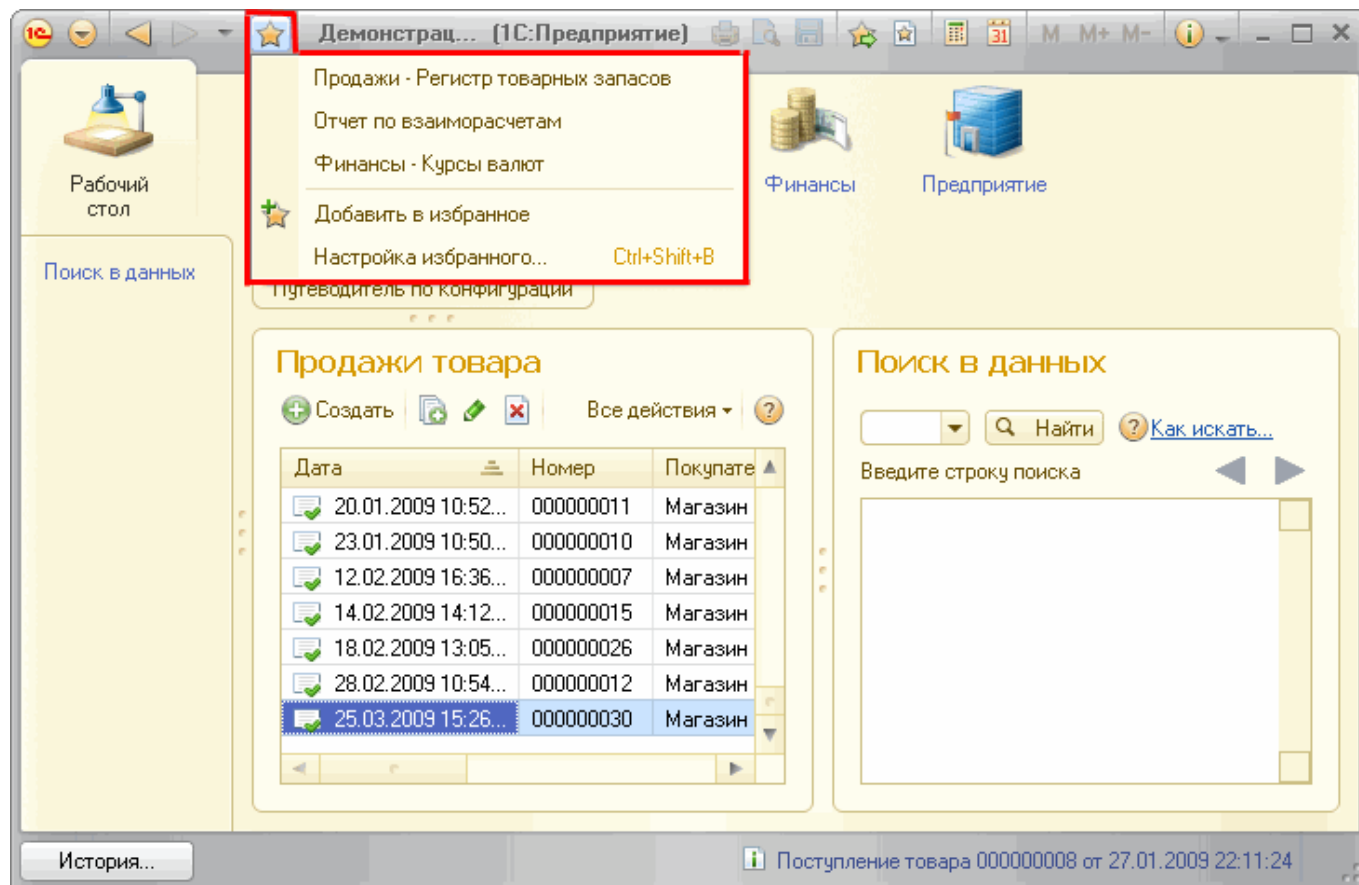


Рис. 170. Избранное

Чтобы перейти по ссылке из списка избранного, следует нажать кнопку [Избранное](#) и выбрать ссылку в списке нажатием левой кнопки мыши.

10.2.1. Настройка избранного

Для управления составом и порядком избранных элементов используется форма, вызываемая пунктом меню [Избранное – Настройка избранного](#) области системных команд.

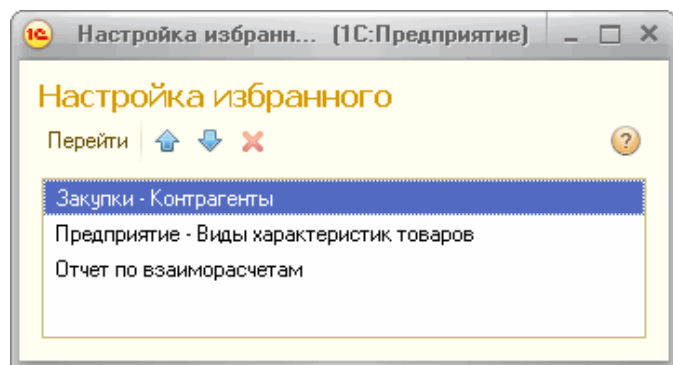


Рис. 171. Диалог настройки списка избранных ссылок

Чтобы перейти по ссылке, нужно выбрать ее с помощью мыши или выбрать строку и нажать кнопку [Перейти](#).

Упорядочить список избранных можно с помощью кнопок [Переместить текущий элемент вверх](#) и [Переместить текущий элемент вниз](#).

Для удаления элемента избранного следует нажать кнопку [Удалить текущий элемент](#).

10.3. История

«1С:Предприятие» сохраняет историю работы пользователя, которую можно использовать для быстрого доступа к недавно созданным или отредактированным объектам информационной базы (документам, элементам списков и др.).

При интерактивном создании или изменении объектов информация об этом отображается в виде оповещения и попадает в историю (подробнее о механизме оповещений см. в разделе «[Оповещения](#)» данной главы).

История работы пользователя хранится в информационной базе. При этом в истории хранится одна запись на один объект информационной базы за каждый день (запись о последующем изменении замещает запись о предыдущем изменении данного объекта).

Просмотр истории. Следует нажать кнопку [История](#), которая расположена в левом нижнем углу основного окна приложения. При этом откроется список всех событий (см. [рис. 172](#)), а при нажатии левой кнопки мыши на событии откроется форма объекта, к которому это событие относится.

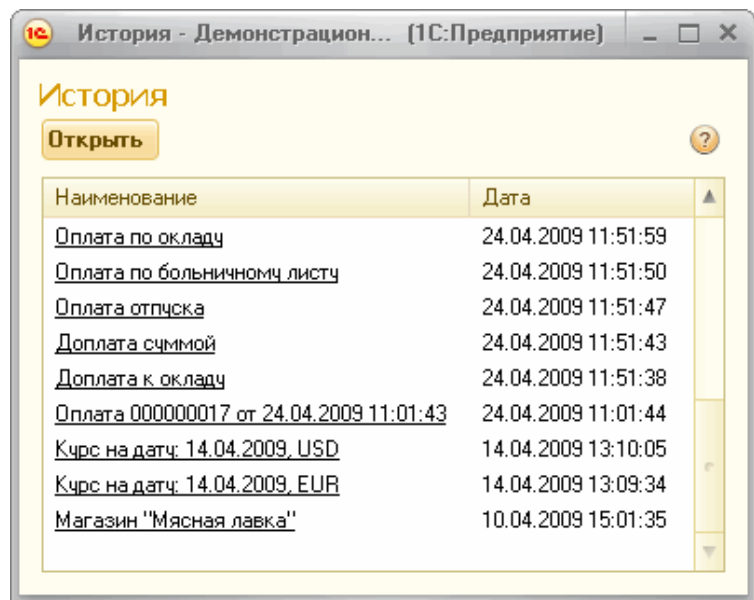


Рис. 172. История

Поиск элемента истории. Для поиска события в списке следует нажать правую кнопку мыши в списке и в контекстном меню выбрать пункт [Искать](#). Поиск события выполняется аналогично поиску в списке. Также в списке событий поддерживается быстрый поиск. Описание поиска в списках содержится [здесь](#).

Обновление списка истории. Однажды открытый список событий считывается программой только в момент открытия окна и в дальнейшем не обновляется. Для того чтобы обновить окно истории, его нужно закрыть и открыть заново.

При нажатии на заголовок панели или на иконку в заголовке откроется форма истории.

10.4. Все функции

Система предоставляет возможность открывать все доступные формы. Для этого используется список, отображающийся при вызове команды главного меню [Все функции](#). Команда доступна, если установлена соответствующая настройка в параметрах системы и у пользователя есть право вызова данного режима (подробнее см. раздел «[Системные параметры](#)» [здесь](#)).

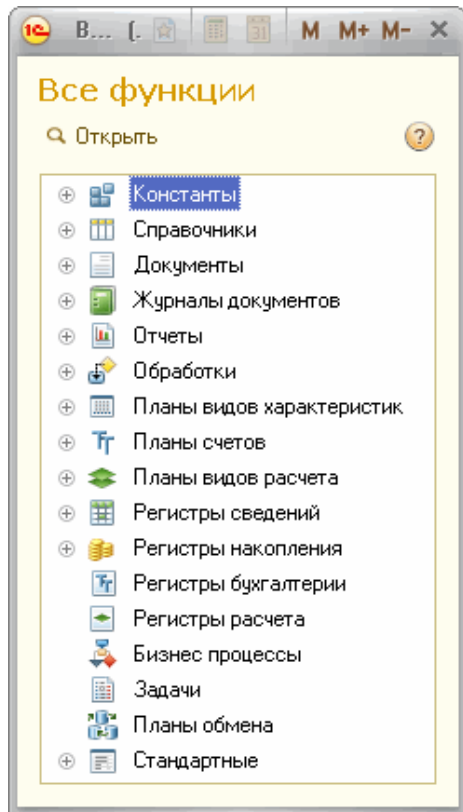


Рис. 173. Список всех функций системы

В список не включены объекты, на которые у пользователя нет права на просмотр.

В группе **Стандартные** отображаются стандартные функции (список активных пользователей, журнал регистрации и т. д.). Подробнее о стандартных функциях см. книгу «1С:Предприятие 8.3. Руководство администратора».

Чтобы перейти к объекту, нужно дважды щелкнуть на объекте в дереве объектов или выбрать объект в дереве и нажать кнопку **Открыть**.

Для быстрого перехода к нужному объекту в списке можно воспользоваться функцией поиска. При вводе текста с клавиатуры поиск будет выполняться по началам названий объектов. Также можно использовать сочетание клавиш **Ctrl + F** для поиска в любой части названия функции. При этом будет использован стандартный диалог поиска в списке.

10.5. Оповещения

Механизм оповещений предназначен для информирования пользователя о том, что система выполнила то или иное действие. Оповещения могут создаваться системой или разработчиком прикладного решения.

Оповещение отображается в окне, которое по умолчанию привязано к системному лотку панели задач операционной системы. Если с оповещением связана какая-либо ссылка, то пояснение будет одновременно являться гиперссылкой. При ее нажатии откроется объект, на который указывает ссылка. Пояснение также будет представлено гиперссылкой, если оповещение сформировано системой автоматически (при интерактивной записи/изменении объекта).

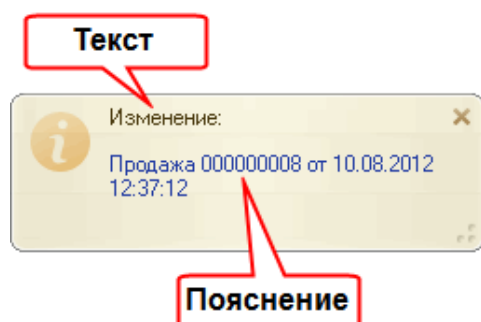


Рис. 174. Оповещение об изменении документа «Поступление денег»

- **Текст** – причина оповещения. Если было изменено несколько объектов, то в скобках будет указано количество измененных объектов;
- **Пояснение** – уточняющая информация.

Окно оповещения исчезает с экрана по истечении некоторого времени. Однако если навести на него курсор мыши, то окно будет отображаться до тех пор, пока его не закрыть или не отвести курсор мыши.

Если в процессе работы системой сгенерировано несколько разных оповещений, то в тексте оповещения в скобках будет указано количество измененных объектов, а в качестве пояснения будет использовано название списка.

При отображении в информационной панели список оповещений выравнивается по правому краю панели. Наиболее позднее оповещение будет отображаться у правого края панели. Если оповещений больше пяти, то отображаются только пять последних оповещений.

ПРИМЕЧАНИЕ. Список оповещений очищается при закрытии основного окна приложения.

10.6. Сообщения

В процессе работы с программой могут возникать ситуации, когда действие, запрошенное пользователем, не может быть выполнено. Например, если при проведении документа поле, обязательное для заполнения, окажется незаполненным. В подобных случаях в программе может быть предусмотрено оповещение пользователя в специальном окне сообщений. Окно сообщений будет содержать список причин возникшей ошибки.

По двойному щелчку в строке списка сообщений откроется специальное окно с указателем на поле, в заполнении которого возникла ошибка. Если сообщение не связано ни с каким отдельным полем, то отобразится стандартное предупреждение. Если сообщение не может быть отображено у требуемой ячейки таблицы формы, то сообщение позиционируется на самой таблице.

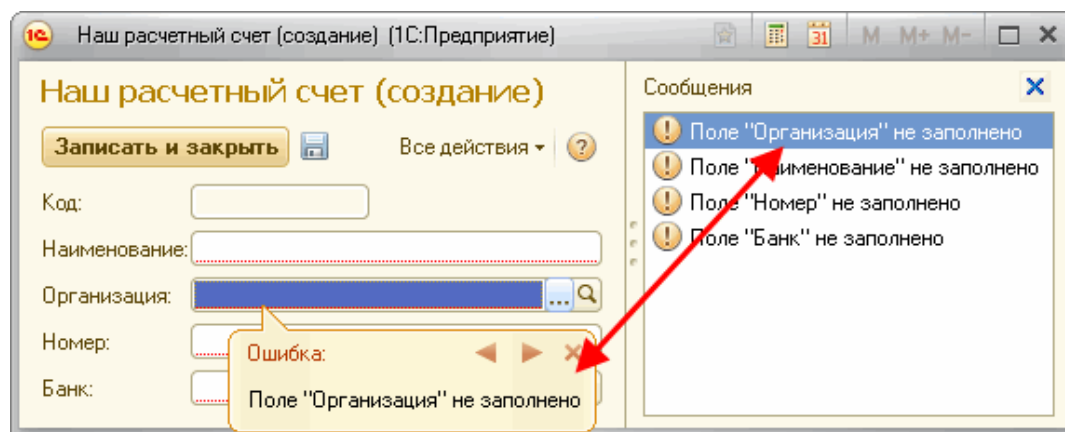


Рис. 175. Сообщения об ошибке

Если сообщение связано с каким-либо реквизитом другого объекта информационной базы, то будет открыта форма этого объекта, и сообщение с указателем на конкретное поле отобразится уже в этом окне.

Если в окне сообщений отображается более одного сообщения, то в нижней части окна, указывающего на поле с ошибкой, появляются кнопки **Вперед/Назад**. С их помощью можно перемещаться между полями, с которыми связаны сообщения.

При выполнении таких операций, как запись элемента или проведение документа, окно сообщений очищается и заполняется новыми сообщениями, которые были сформированы в ходе выполнения выбранного действия.

Если в таблице формы был выполнен поиск в данных, то при выборе сообщения пользователю поиск данных в таблице отменяется и сообщение отображается у требуемой ячейки.

ПРИМЕЧАНИЕ. Чтобы открыть и закрыть окно сообщений в процессе работы с формой, используются клавиши **Ctrl + Shift + Z**.

10.7. Состояние длительного процесса

Во время работы с программой могут возникать ситуации, когда системе требуется время для выполнения

некоторого процесса. Например, при проведении большого количества документов, выполнении тяжелых расчетов. При этом на экране отображается сообщение, содержащее состояние длительного процесса, см. [рис. 176](#).

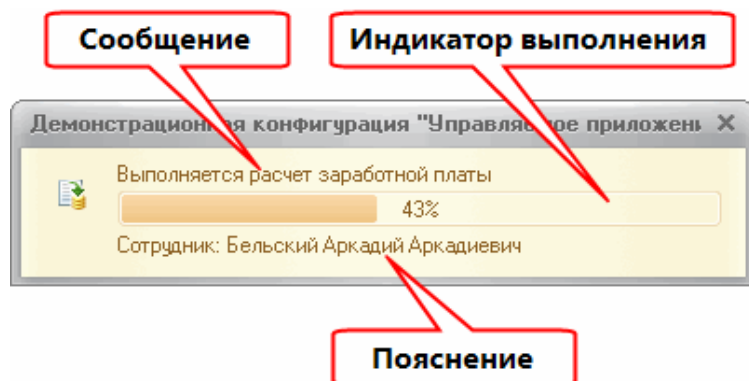


Рис. 176. Состояние выполнения процесса

Сообщение автоматически «гаснет» по истечении некоторого времени. При наведении курсора на сообщение оно снова отображается. В зависимости от настроек программы выполнение процесса может быть прервано по нажатию клавиш **Ctrl + Break**.

10.8. Калькулятор

Для выполнения несложных текущих расчетов система «1С:Предприятие» имеет встроенный калькулятор. Его можно вызвать в любом режиме программы (кроме случаев, когда программа ждет ответа на запрос), выбрав пункт главного меню [Сервис – Калькулятор](#).

Калькулятор можно перетащить мышью в любое место экрана, а также изменить размер окна. При этом все элементы управления, расположенные в форме калькулятора, изменяют свой размер пропорционально изменению размеров окна.

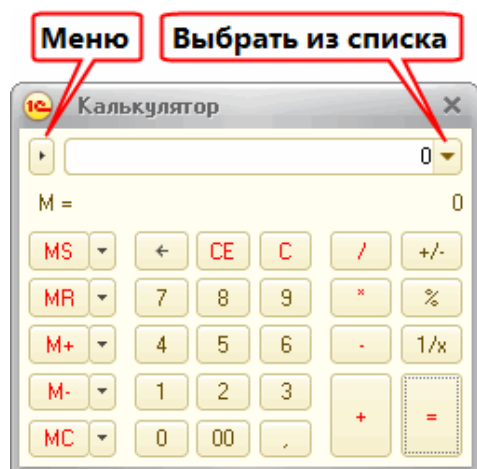


Рис. 177. Калькулятор

Калькулятор позволяет выполнять все арифметические действия, вычислять проценты и обратные значения, запоминать числа и выполнять сложение и вычитание с запомненными значениями. При вводе числа калькулятор позволяет редактировать введенное значение, удаляя неверно введенные разряды.

Для работы с калькулятором можно использовать как мышь, так и клавиатуру. По выполняемым функциям цифровой калькулятор напоминает «настоящие» калькуляторы.

Калькулятор запоминает последние 15 вычислений. Для просмотра результата любого вычисления следует нажать кнопку [Выбрать из списка](#), расположенную справа от поля ввода, и выбрать нужное вычисление.

Калькулятор имеет 10 регистров памяти для запоминания чисел и оперирования ими при вычислениях. С помощью кнопок [MR](#), [M+](#), [M-](#) и [MC](#) оперируют с первым регистром.

Для запоминания результата вычисления в первом регистре следует нажать кнопку [MS](#).

В поле **M =** отображается значение буфера обмена, если в нем содержится число.

Нажатием кнопки **MR** запомненное значение читается из первого регистра в поле ввода. По кнопке **M+** запомненное в первом регистре значение прибавляется к результату вычислений, по кнопке **M-** запомненное в первом регистре значение вычитается из результата вычислений. Кнопка **MC** сбрасывает содержимое первого регистра.

Для запоминания числа в другие регистры число должно быть набрано в поле ввода (нужно ввести его или произвести расчеты). Затем следует нажать кнопку **Выбрать**, расположенную рядом с кнопкой **MS**.

На экран выводится список регистров.

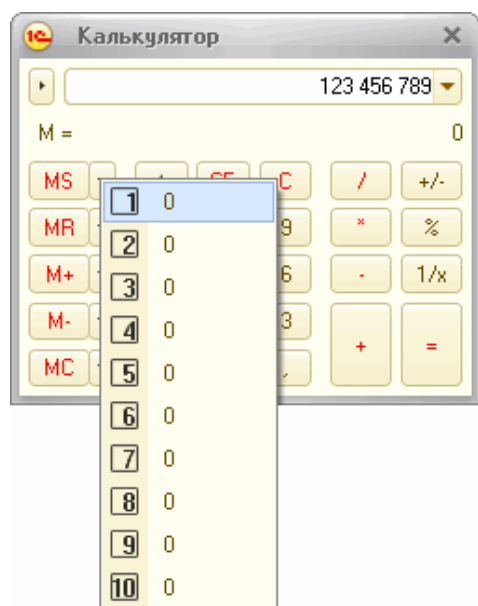


Рис. 178. Список доступных регистров

Для записи числа в регистр следует нажать кнопку соответствующего регистра.

Для извлечения числа из регистра нужно нажать кнопку, расположенную рядом с кнопками **MR**, **M+** или **M-**, в зависимости от вида действия, и выбрать регистр. Аналогично выполняется процедура для очистки регистра (кнопка **MC**).

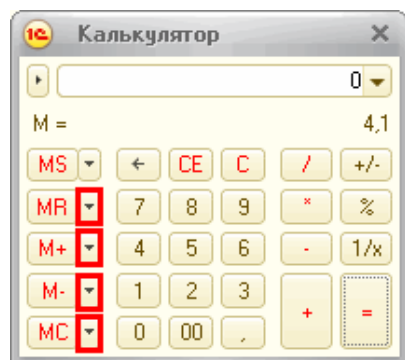


Рис. 179. Кнопки для работы с регистрами

Калькулятор может работать в различных режимах. Выбор режима производится нажатием кнопки **Меню** калькулятора (см. [рис. 177](#)) и выбором в контекстном меню нужного пункта. Пункты меню **Формульный калькулятор** и **Микрокалькулятор** позволяют включить или выключить соответствующий режим работы калькулятора.

Пункт контекстного меню **Настройка** предназначен для установки точности калькулятора. На экран выводится диалог настройки, в котором указывается число знаков после запятой для представления результата.

В режиме **Микрокалькулятор** все действия выполняются с клавиатуры. Калькулятор имеет упрощенную форму.

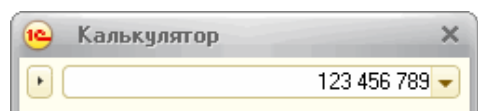


Рис. 180. Калькулятор в режиме «Микрокалькулятор»

Для ввода чисел используются цифровые клавиши клавиатуры. Для очистки поля ввода – кнопка **C**.

Формульный калькулятор. В этом режиме все действия записываются в строку калькулятора последовательно. Для организации определенной последовательности действий и ввода сложных выражений допускается использование скобок.

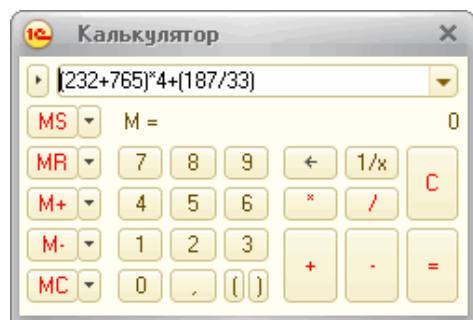


Рис. 181. Формульный калькулятор

Калькулятор вызывается для выбора значения реквизита типа **Число** в поле ввода нажатием кнопки выбора или по клавише **F4**. Калькулятор позволяет вводить большее число знаков после десятичной точки, чем определено для реквизита. При нажатии кнопки **OK** производится округление по установленным правилам. Для настройки точности калькулятора следует нажать кнопку **Меню** и выбрать **Настройка**. На экран выводится диалог настройки, в котором указывается число знаков после запятой для представления результата.

10.8.1. Работа с буфером обмена

При работе с буфером обмена система «1С:Предприятие» предоставляет расширенный сервис работы с числами. Кроме запоминания числа в буфер число можно добавить к значению в буфере или вычесть из значения в буфере.

Для этого обычно используются кнопки стандартной командной панели **M**, **M+** и **M-**.

При нажатии кнопки **M** (можно использовать сочетание клавиш **Shift + Num***) выделенное значение помещается в буфер обмена.

Кнопка **M+** (сочетание клавиш **Shift + Num+**) позволяет добавить текущее выделенное значение к значению в буфере обмена.

Кнопка **M-** (сочетание клавиш **Shift + Num-**) позволяет вычесть текущее значение из буфера обмена.

Данные действия доступны во всех формах для числовых значений и при работе с таблицами. Кроме того, при работе с табличным документом существует возможность выполнять одновременные действия с несколькими ячейками: например, можно выделить несколько ячеек и, используя указанные действия, поместить или добавить в буфер обмена их сумму.

Полученные результаты в буфере обмена можно использовать в любых формах, например, вставляя значения с помощью клавиш **Shift + Ins** или соответствующих пунктов меню.

Числовое выражение текущего значения буфера обмена можно просмотреть в режиме формульного калькулятора в поле **M=**.

Если требуется вставить число из среды «1С:Предприятия» в калькулятор **Windows** или в ячейку **Microsoft Office Excel**, то для запоминания следует использовать команды **M**, **M+**, **M-**, а не команды запоминания в буфер обмена.

Нужно иметь в виду, что для формульного калькулятора и табло число сохраняется в соответствии с региональными установками информационной базы, но без разделителей групп и замены разделителя целой и дробной части на точку.

Для всех остальных приемников буфера обмена число сохраняется с региональными установками операционной системы и также без разделителей групп.

10.9. Календарь

Система «1С:Предприятие» содержит встроенный календарь. Его можно вызвать в любом режиме программы, выбрав пункт главного меню **Сервис – Календарь**. Календарь также используется для ввода дат.

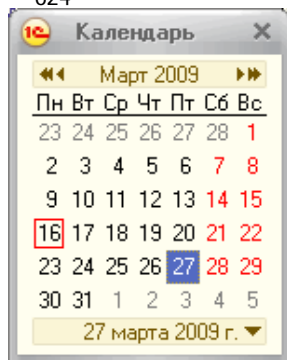


Рис. 182. Окно календаря

В окне выводится календарь на текущий месяц. Его наименование и год выводятся в верхней строке. Ниже расположена строка дней недели. Красной рамкой выделяется текущая дата. При выборе указателем мыши другой даты она выделяется рамкой серого цвета, а выбранная дата выводится в нижней строке окна.

Для перехода между месяцами и для смены года используются кнопки, расположенные вокруг заголовка.

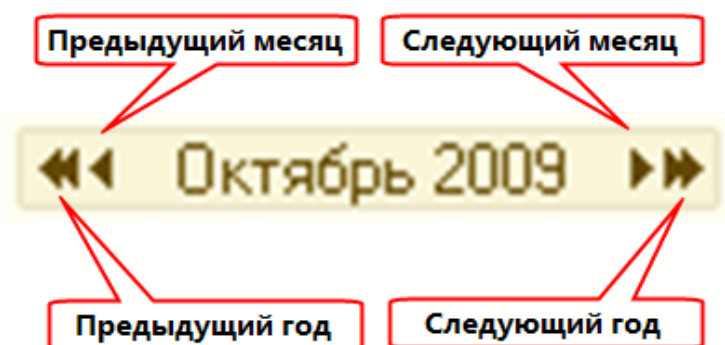


Рис. 183. Кнопки перехода

Для быстрого выбора текущей даты, даты начала и конца месяца, начала и конца года рекомендуется использовать контекстное меню, вызываемое кнопкой, расположенной в нижней строке справа от наименования выбранной даты.

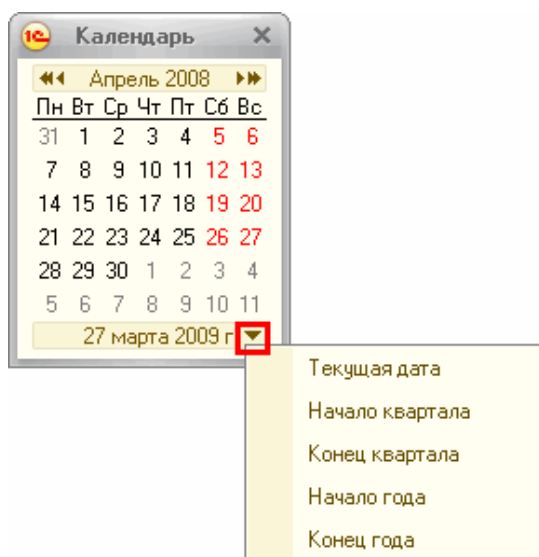


Рис. 184. Календарь. Контекстное меню выбора

Если щелкнуть мышью строку, содержащую наименование месяца и года, то на экран выводится список месяцев для выбора. Перемещая указатель мыши вверх или вниз, можно прокручивать список до нужного значения. Для смены месяца выбирается нужная строка списка.

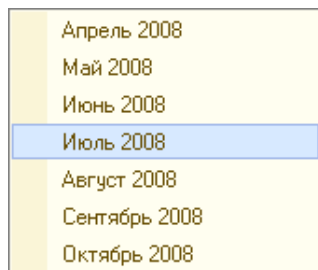


Рис. 185. Календарь. Список месяцев

Календарь можно перетащить мышью в любое место экрана, а также изменить размер окна.

При увеличении размера окна показываются календари на несколько месяцев.

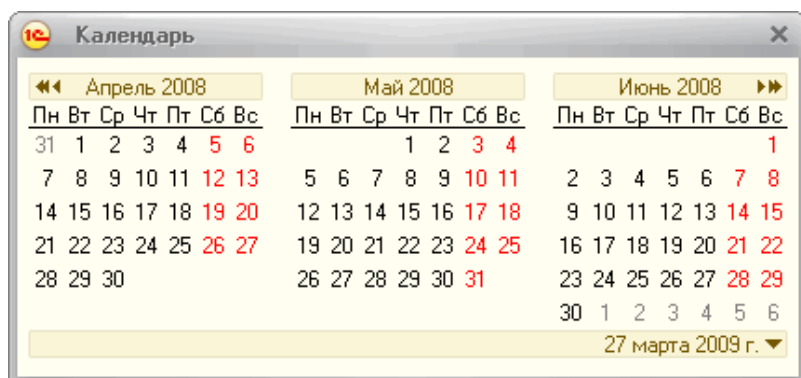


Рис. 186. Календарь на три месяца

Добавление календарей производится при любом направлении увеличения размеров окна. При уменьшении размеров дополнительные календари убираются.

Календарь также может быть размещен в форме как элемент формы, связанный с определенным реквизитом, содержащим дату. Кроме того, для дат, расположенных в форме в виде поля ввода, выбор значения может производиться с помощью календаря, который вызывается нажатием кнопки выбора или клавиши **F4**.

Приемы работы с этими календарями аналогичны описанным выше.

10.10. Сравнение файлов

Режим сравнения файлов дает возможность пользователю сравнить два любых файла. Для сравнения файлов необходимо выбрать пункт главного меню **Файл – Сравнить файлы**. На экран будет выдан диалог для выбора сравниваемых файлов.

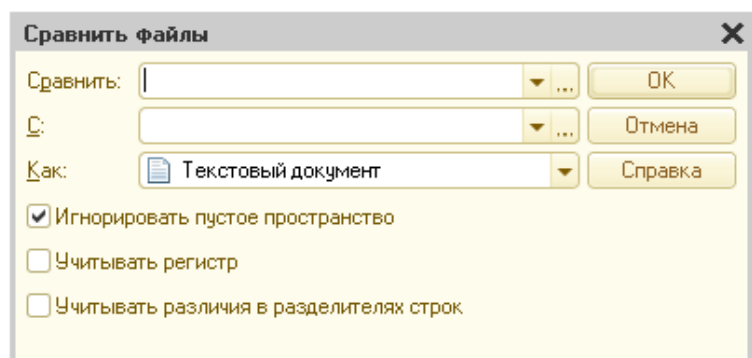


Рис. 187. Диалог выбора файлов для сравнения

В полях **Сравнить** и **С** необходимо указать имена сравниваемых файлов. Их можно ввести вручную, выбрать из выпадающего списка или нажатием кнопки **Выбрать** вызвать стандартный диалог открытия файла и выбрать файл в этом диалоге. В поле **Тип файлов** диалога можно указать любой тип исходного файла.

В поле **Как** указывается вид сравнения файлов. Текстовый и табличный документы сравниваются. Допускается двоичное сравнение. В этом случае результатом сравнения будет сообщение о совпадении или различии файлов.

Если в полях **Сравнить** и **С** указаны файлы различных типов, то выбранные файлы при сравнении будут приводиться к виду, указанному в поле **Как**. Затем выполнится сравнение, и на экран будет выведена информация о его результатах.

Если установлен флажок **Учитывать различия в разделителях строк**, то при сравнении будут учитываться символы, используемые в качестве разделителей строк.

Для начала сравнения следует нажать кнопку **ОК**.

Если выбран вид сравнения **Текстовый документ**, то выбранные файлы будут интерпретироваться при сравнении как текстовые, несмотря на то, что в полях **Сравнить** и **С** могут быть выбраны нетекстовые (по расширению) файлы. Если указанные файлы или один из них не могут быть интерпретированы как текстовые, то производится двоичное сравнение.

10.10.1. Сравнение текстовых файлов

Для текстовых файлов установка флажка **Игнорировать пустое пространство** означает, что пробелы сравниваться не будут.

Флажок **Различать регистр** позволяет при сравнении различать прописные и строчные буквы. Если флажок установлен, одна и та же прописная и строчная буквы будут считаться разными символами.

Флажок **Учитывать разделители строк** означает, что при сравнении файлов будут учитываться разделители строк.

Перед сравнением программа последовательно выполняет попытку чтения файлов в кодировках UTF-8, UTF-16 и ANSI. Если файлы прочитаны и если при чтении в ANSI файлы не содержат 0, то программа сравнивает их как текстовые документы. Если попытки чтения были неудачными (или для ANSI содержится 0), то программа сравнивает их как двоичные данные.

Для текстовых файлов на экран будет выдано окно просмотра различий. В двух колонках окна просмотра различий будут помещены сравниваемые тексты. В текстах разными цветами выделены удаленные, измененные и добавленные (новые) строки. В окне имеется панель инструментов со стандартными для текстового редактора «1С:Предприятия» кнопками поиска и закладок, а также двумя кнопками, специфическими для данного окна.

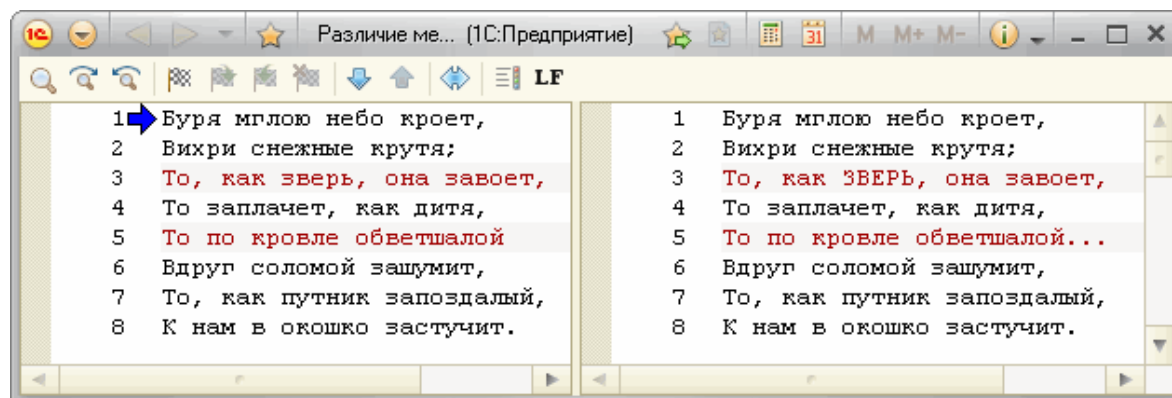


Рис. 188. Диалог сравнения текстовых файлов

Это большие стрелки, направленные одна вверх, другая вниз. Они служат для быстрого перехода на начало ближайшего предыдущего или последующего измененного текста.

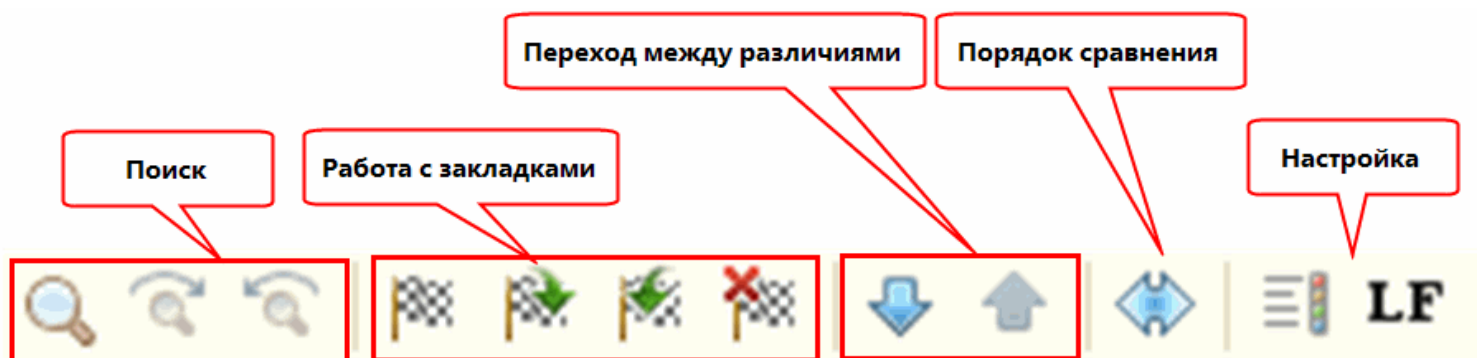


Рис. 189. Панель инструментов для сравнения файлов

При нажатии кнопки **LF** в текстах отображаются разделители строк в виде букв **LF** и **CR**.

Различия в сравниваемых текстах показываются с помощью цвета текста. Цвет текста можно настроить (по кнопке **Настройка** запускается окно настройки цветов). По умолчанию удаленный текст выделяется синим цветом, добавленный – зеленым, а измененный – красным.

В левой части окна показывается текст файла, указанного в поле **Сравнить**, в правой – указанного в поле **С**. Нажатием кнопки **Порядок сравнения** можно поменять местами сравниваемые файлы. При этом меняется и цвет пометки. Если первоначально в левой части был показан удаленный текст, то при смене порядка сравнения этот текст в правой части будет показан как добавленный.

10.10.2. Сравнение табличных документов

Если сравниваются табличные документы, то на экран будет выдано окно просмотра различий.

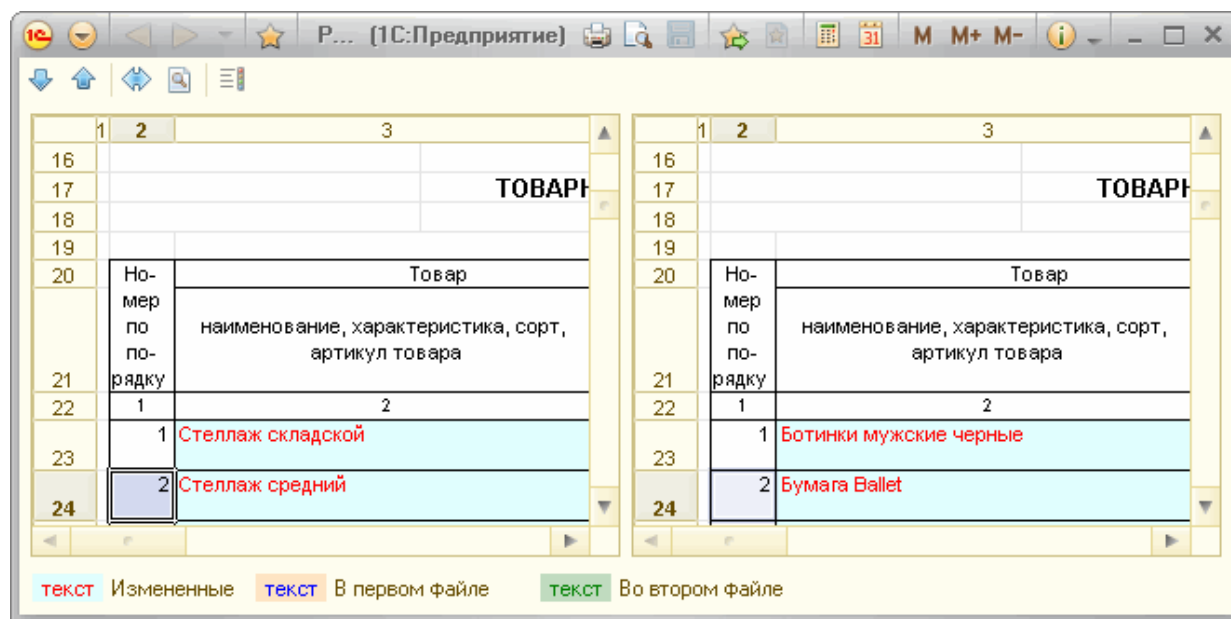


Рис. 190. Окно сравнения табличных документов

Окно состоит из двух частей (левой и правой). В каждую часть окна просмотра различий помещается исходный документ. С помощью кнопок командной панели производится управление просмотром и настройка цветов.

Кнопками **Следующее различие** и **Предыдущее различие** осуществляется переход к следующему или предыдущему различию.

Различия показываются с помощью цвета фона и текста. Обозначения цветов перечислены в легенде.

С помощью кнопки **Настройка параметров просмотра сравнения** открывается диалог настройки параметров показа различий.

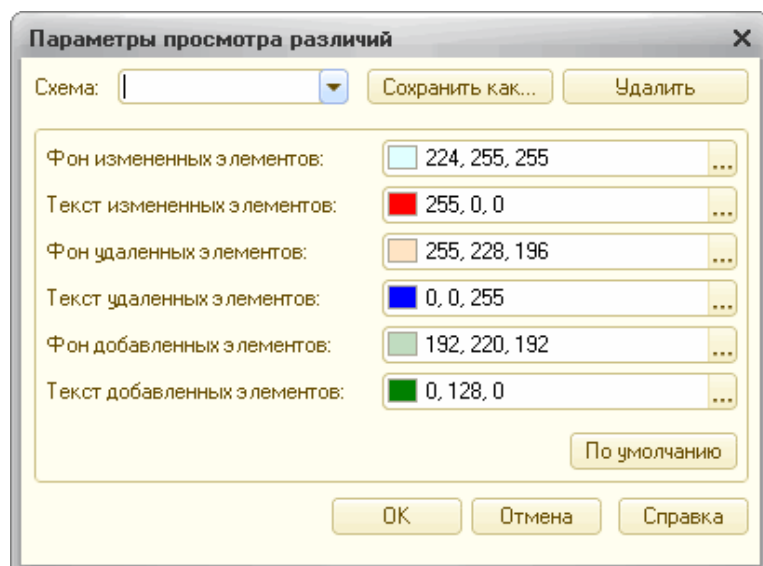


Рис. 191. Настройка параметров показа различий

Значения настроек можно сохранять как схемы. Для этого следует нажать кнопку [Сохранить как](#), в открывшемся диалоге ввести наименование настройки и нажать кнопку [ОК](#).

Если есть несколько схем настроек, то выбор нужной осуществляется в поле [Схема диалога настройки](#).

Для восстановления стандартных настроек цветов следует нажать кнопку [По умолчанию](#).

Различия в ячейках, строках, свойствах строк и колонок показываются цветом фона и цветом текста.

Различия в объединениях ячеек показываются обводящей рамкой цвета текста.

Различия в группировках и именованных областях обозначаются цветом текста.

Различия в рисунках показываются обводящей рамкой цвета текста, кроме прямоугольника, линии и овала, которые рисуются цветом текста.

В окне разными цветами выделены измененные, удаленные и добавленные (присутствуют в первом или втором файлах) строки.

Для выделенных ячеек можно получить подробный отчет о невидимых различиях (текст, шрифт, формат, объединение, имена и т. д.). Для этого следует нажать кнопку [Подробнее...](#)

10.11. Сообщения об ошибках

В случае возникновения ошибки при выполнении программы на экран выводится предупреждение (может содержать разное число кнопок, назначение кнопок также может быть разным), например, следующего вида:

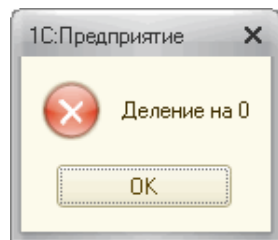


Рис. 192. Ошибка деления на ноль

Чтобы получить более подробную информацию об ошибке в окне "[О программе](#)" следует нажать ссылку [Информация для технической поддержки](#). Подробнее см. [здесь](#).

При аварийном завершении программы «1С:Предприятие» система отображает диалог с информацией о процессе записи дампа, который автоматически закрывается после завершения записи.

Глава 11. Настройка программы

Система «1С:Предприятие» позволяет настраивать интерфейс приложения, как глобальный, так и каждую форму в отдельности (если эта возможность есть в конфигурации), задавать системные параметры, управлять отображением текстовых и табличных документов и расположением окон.

11.1. Интерфейс

При работе с «1С:Предприятием» существует возможность настраивать интерфейс приложения, в частности:

- рабочий стол;
- панель разделов;
- панели навигации (как основного, так и вспомогательного окна);
- панель действий;
- командную панель вспомогательного окна.

Пользователь может выбирать команды из доступных и управлять их расположением.

Диалоги настройки вызываются из главного меню "1С:Предприятия" (Пункт главного меню [Вид](#)).

Также окно настройки для каждой панели пользователь может вызвать из ее контекстного меню любой из панелей.

11.1.1. Рабочий стол

Система позволяет устанавливать состав форм, которые отображаются на рабочем столе. При этом можно управлять только теми формами, которые расположены на рабочем столе.

Для перехода к диалогу настройки рабочего стола на рабочем столе в контекстном меню панели разделов нужно выбрать пункт [Настройка рабочего стола](#). Также можно использовать команду главного меню [Вид – Настройка рабочего стола](#).

Настройка производится в диалоге, состоящем из трех списков: [Доступные формы](#) и группы колонок [Левая колонка](#), [Правая колонка](#). В списке [Доступные формы](#) отображаются формы, не используемые ни в левой, ни в правой колонке.

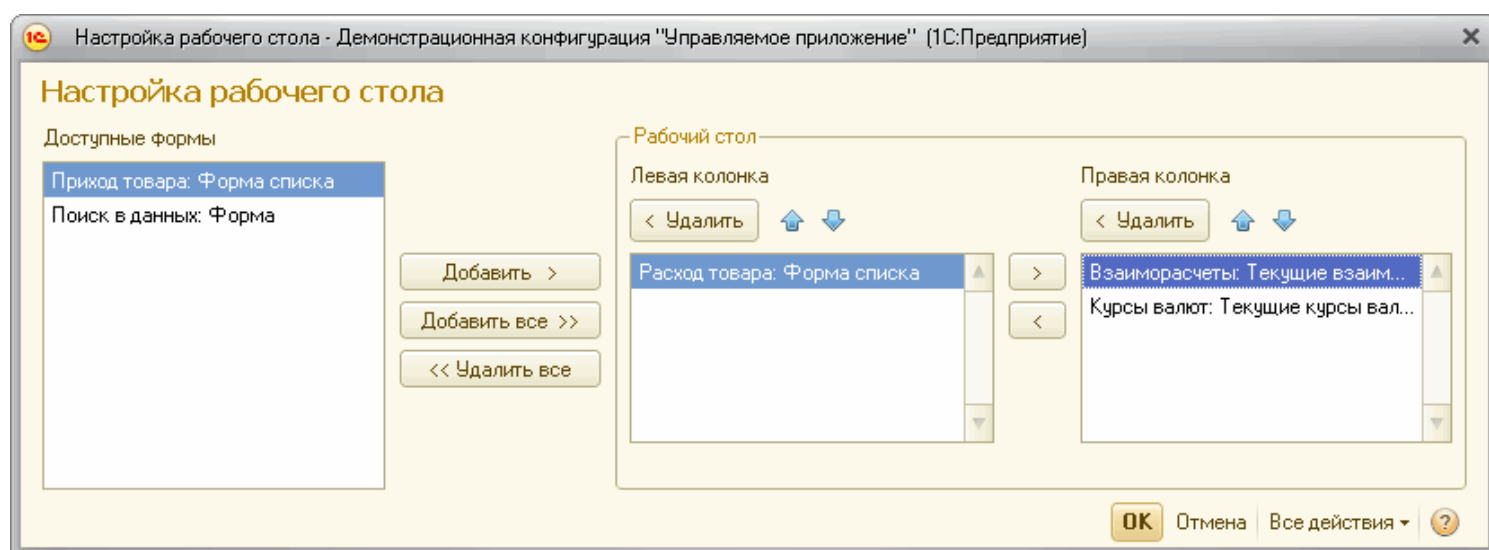


Рис. 193. Диалог настройки рабочего стола

Формы можно перетаскивать из одного списка в другой с помощью кнопок [Добавить >](#), [Добавить все >>](#), перемещать между колонками с помощью кнопок [>](#) и [<](#). Чтобы удалить форму из колонки (поместить ее обратно в список доступных форм), нужно выбрать ее в списке и нажать кнопку [Удалить <](#) или дважды щелкнуть по названию формы в списке. Чтобы удалить все формы с рабочего стола, следует нажать кнопку [<< Удалить все](#). Также можно изменять порядок форм в колонках с помощью кнопок [Вверх – Вниз](#). Формы выводятся на рабочий стол в том же порядке, который установлен в настройках.

В процессе настройки можно отменить изменения, сделанные как в данном сеансе работы, так и ранее, и вернуться к настройкам, заданным в конфигурации. Для этого в меню [Все действия](#) имеется команда [Установить стандартные настройки](#). Эта команда не вызывает немедленного изменения командного интерфейса. После ее выполнения настройку можно продолжить. Применение настроек происходит по нажатию кнопки [OK](#).

11.1.2. Настройка отображения панелей

Панель навигации и панель действий могут отображаться, даже если они не содержат команд. Также следует отметить, что при переключении разделов высота панели действий не изменяется.

Чтобы настроить или спрятать панели, используются соответствующие команды главного меню [Вид](#).

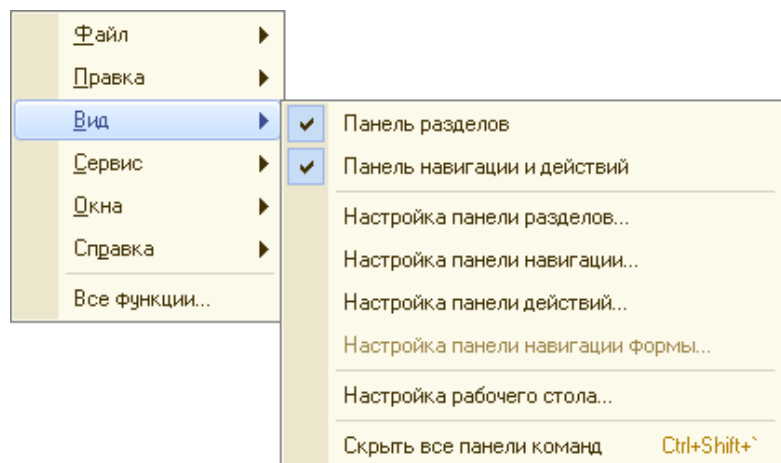


Рис. 194. Меню «Вид» в интерфейсе «Формы в закладках»

Данное подменю позволяет управлять видимостью панелей и содержимым панели разделов, панелей навигации, панели действий и рабочего стола. Это позволяет увеличить место под формы и уменьшить вероятность появления полос прокрутки. Чтобы спрятать все панели, используется команда главного меню [Вид – Скрыть все панели команд](#) или сочетание клавиш [Ctrl + Shift + `](#). Нужно учитывать следующую особенность: команда [Скрыть все панели команд](#) не будет работать, если панель разделов, панель навигации и панель действий скрыты с помощью меню [Вид](#).

11.1.3. Панель разделов

При настройке панели разделов пользователь устанавливает список команд перехода к разделам.

Диалог настройки панели разделов открывается командой [Настройка панели разделов...](#) контекстного меню панели или с помощью меню [Сервис](#).

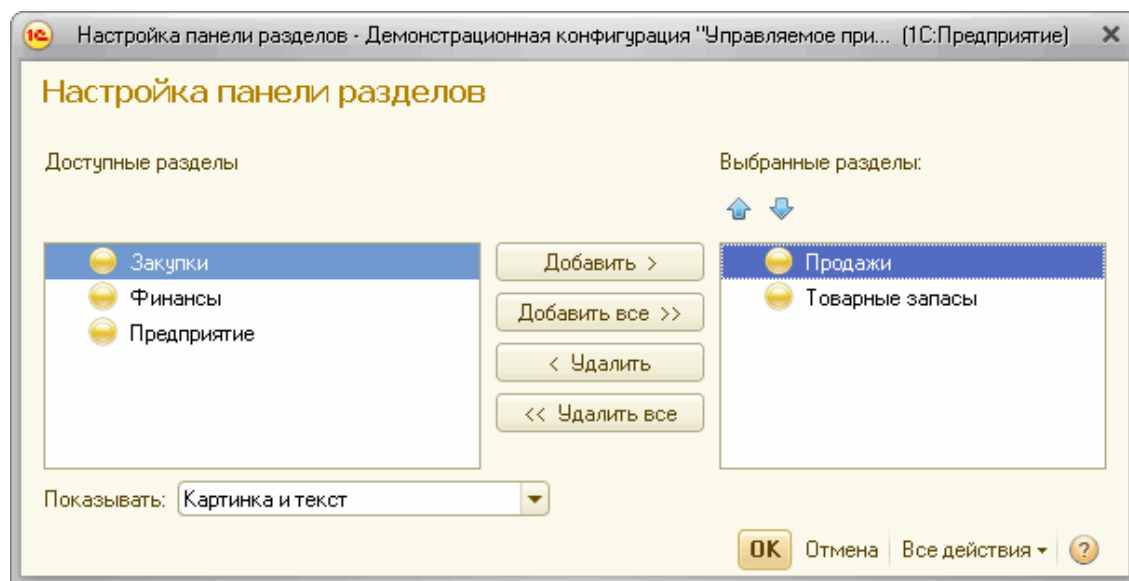


Рис. 195. Форма настройки панели разделов

Чтобы добавить раздел на панель, нужно выбрать его в списке доступных разделов и нажать кнопку [Добавить >](#). В диалоге настройки поддерживается перемещение элементов списка с помощью мыши и выбор нескольких

строк.

Чтобы добавить на панель все доступные разделы, следует нажать кнопку [Добавить все >>](#). Также можно перемещать разделы между списками двойным нажатием левой кнопки мыши по разделу в списке.

Чтобы удалить команды из панели, используются кнопки [Удалить](#) и [Удалить все](#).

Последовательность разделов на панели регулируется с помощью кнопок командной панели списка выбранных (см. [рис. 195](#)).

Для любого раздела можно настроить представление. Для этого нужно выбрать пункт и установить требуемый способ отображения в списке [Показывать: Картинка, Текст, Картинка и текст](#).

Чтобы применить выбранные настройки и закрыть форму, следует нажать кнопку [OK](#). При этом происходит сохранение настроек в информационной базе.

ПРИМЕЧАНИЕ. Если с панели разделов удалены все разделы, она автоматически скрывается и основное окно программы переключается на [Рабочий стол](#).

В процессе настройки можно отменить изменения, сделанные как в данном сеансе работы, так и ранее, и вернуться к настройкам, заданным в конфигурации. Для этого нужно выбрать пункт [Все действия – Установить стандартные настройки](#). Исполнение этой команды не вызывает немедленного изменения командного интерфейса. После ее выполнения можно продолжить настройку.

11.1.4. Панель навигации

Настройка расположения команд на панели навигации производится в диалоге [Настройка панели навигации](#).

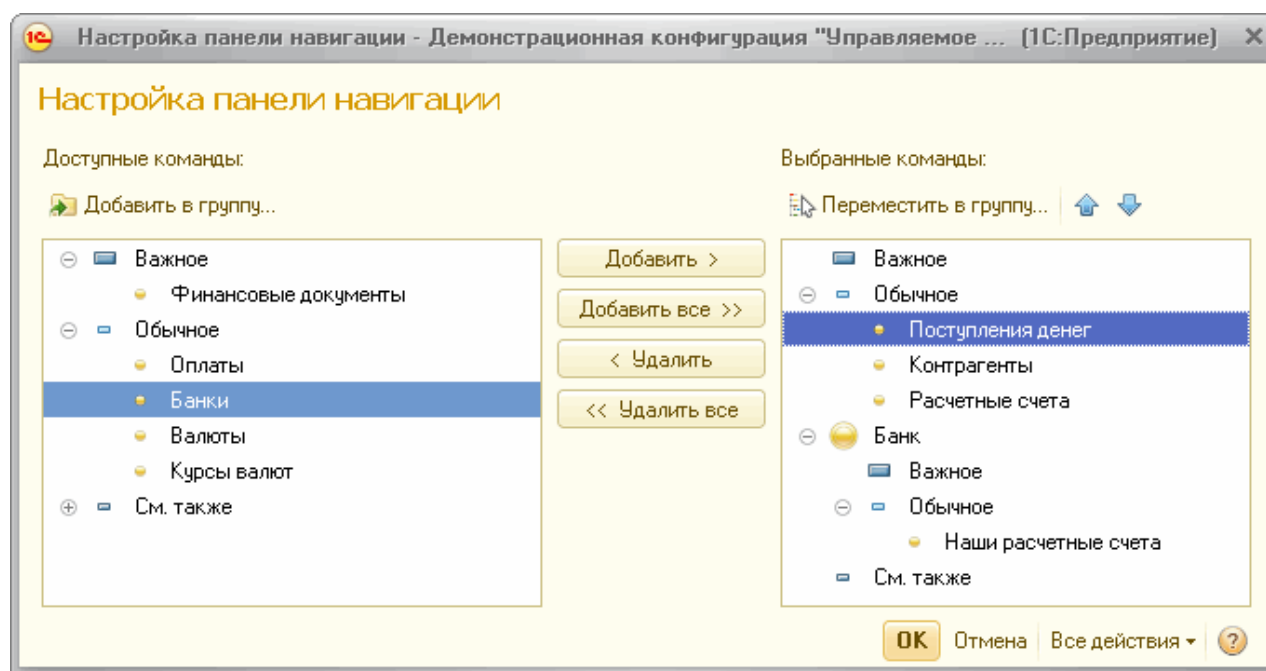


Рис. 196. Настройка панели навигации

- команды перехода к формам списков, например, [Банки](#), [Контрагенты](#);
- подчиненные разделы (например, на [рис. 196](#) текущему разделу подчинен раздел [Банк](#), для которого определена команда [Наши расчетные счета](#)).

Все команды распределены по группам:

- [Важное](#),
- [Обычное](#),
- [См. также](#).

Чтобы добавить команду на панель, нужно выбрать ее в списке доступных команд и нажать кнопку [Добавить >](#).

Чтобы добавить на панель все доступные команды, следует нажать кнопку [Добавить все >>](#). Также можно перемещать команды между списками двойным нажатием левой кнопки мыши по разделу в списке.

Для удаления команды из панели используются кнопки [Удалить](#) и [Удалить все](#).

Последовательность команд регулируется с помощью стрелок на командной панели списка выбранных команд (см. [рис. 196](#)).

Чтобы добавить команду из списка доступных в группу выбранных команд, нужно выбрать ее и нажать кнопку [Добавить в группу](#).

Для перемещения команды в необходимую группу в списке выбранных команд нужно выбрать команду и нажать кнопку [Переместить в группу](#) или перетащить команду в требуемую группу с помощью мыши. Если при этом выбрана группа или подраздел, то операция выполняется для всех команд в группе/подразделе.

Диалоги настройки поддерживают перетаскивание команд между списками и множественный выбор.

ПРИМЕЧАНИЕ. Если некоторая команда относится к подразделу, то она может быть перемещена в другую группу только внутри подраздела, к которому она относится. Например, на [рис. 196](#) команда [Наши расчетные счета](#) может быть перемещена только в группу [Важное](#) подраздела [Банк](#).

Исключение составляет группа [См. также](#), которая может содержать любые команды.

Чтобы применить выбранные настройки и закрыть форму, следует нажать кнопку [ОК](#).

Заметим, что порядок может быть настроен только для команд.

11.1.5. Панель действий

Настройка расположения команд на панели действий производится в диалоге [Настройка панели действий](#).

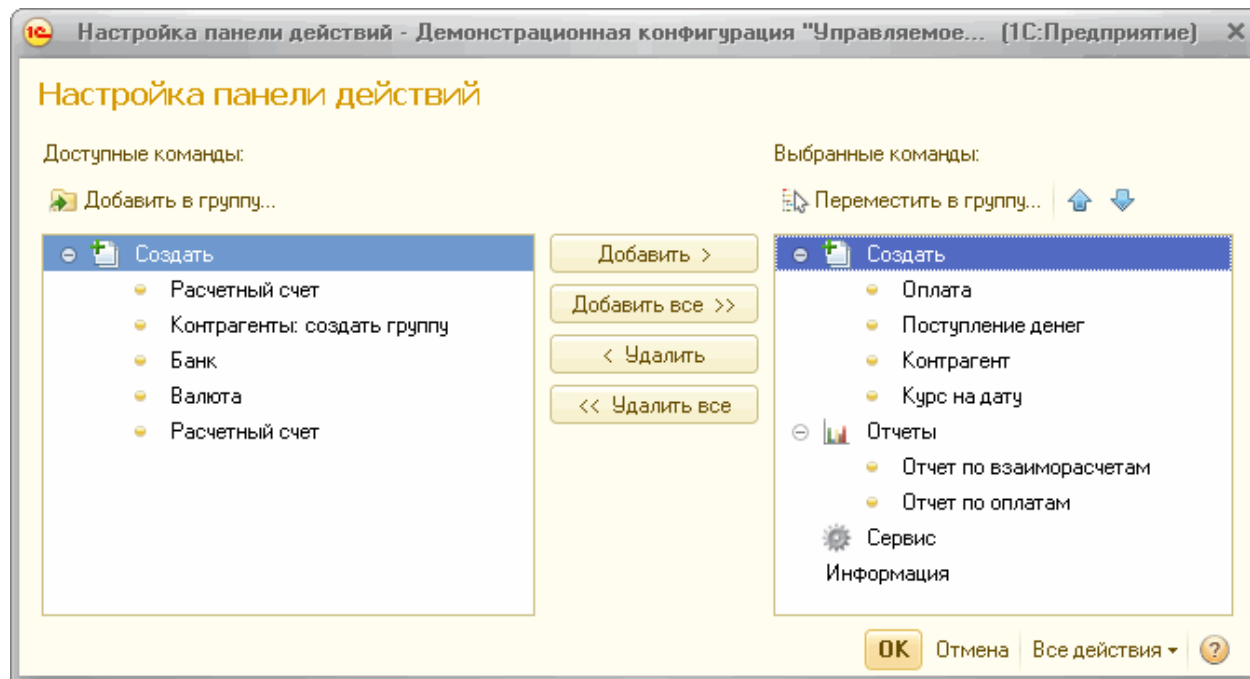


Рис. 197. Настройка панели действий

- команды создания новых объектов (например, [Контрагент](#)) – в стандартной группе [Создать](#);
- команды открытия отчетов (например, [Отчет по взаиморасчетам](#)) – в стандартной группе [Отчеты](#);
- команды вызова обработок – в стандартной группе [Сервис](#);
- другие команды, созданные в конфигурации.

Настройка панели действий производится аналогично настройке панели навигации (см. выше).

11.1.6. Настройка системных команд основного и вспомогательного окна

Чтобы добавить или удалить кнопки на стандартной командной панели основного или вспомогательного окна, следует нажать кнопку [Другие кнопки](#), в появившемся меню навести курсор мыши на пункт [Добавить или удалить кнопки](#) и в отобразившемся меню установить или снять пометку напротив названия команды.

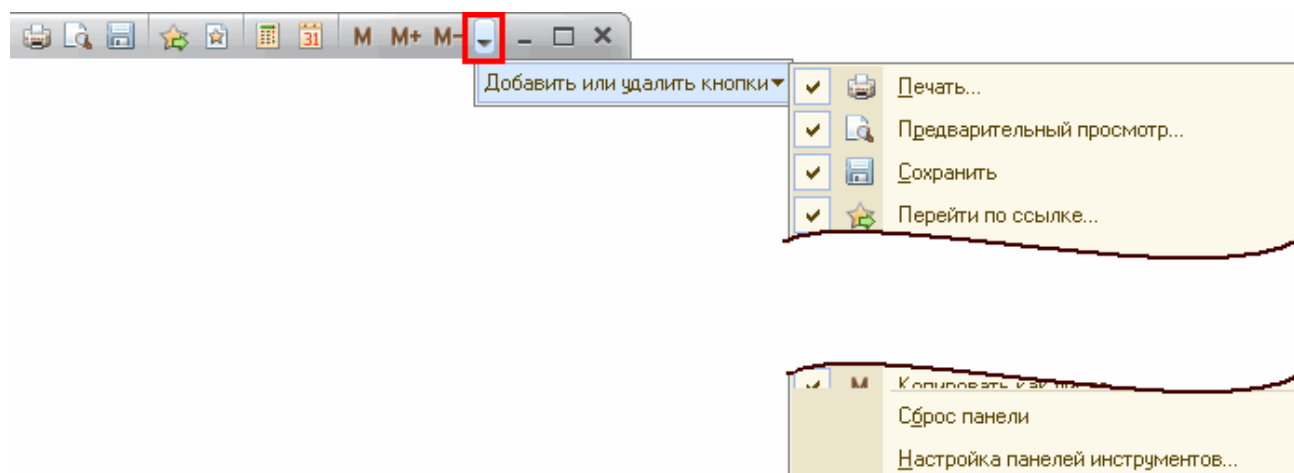


Рис. 198. Настройка командной панели вспомогательного окна

Команда, для которой пометка установлена, отобразится в области системных команд окна.

Команда, для которой пометка снята, скроется из области.

Для возвращения панели первоначального вида используется пункт [Сброс панели](#).

11.1.6.1. Настройка области системных команд вспомогательного окна

Управление областью системных команд **вспомогательного окна** ведется при помощи пункта [Настройка панелей инструментов](#) контекстного меню области.

Диалог [Настройка панелей инструментов](#) содержит две закладки.

Закладка «Панели инструментов»

На данной закладке выполняется выбор и редактирование панелей инструментов.

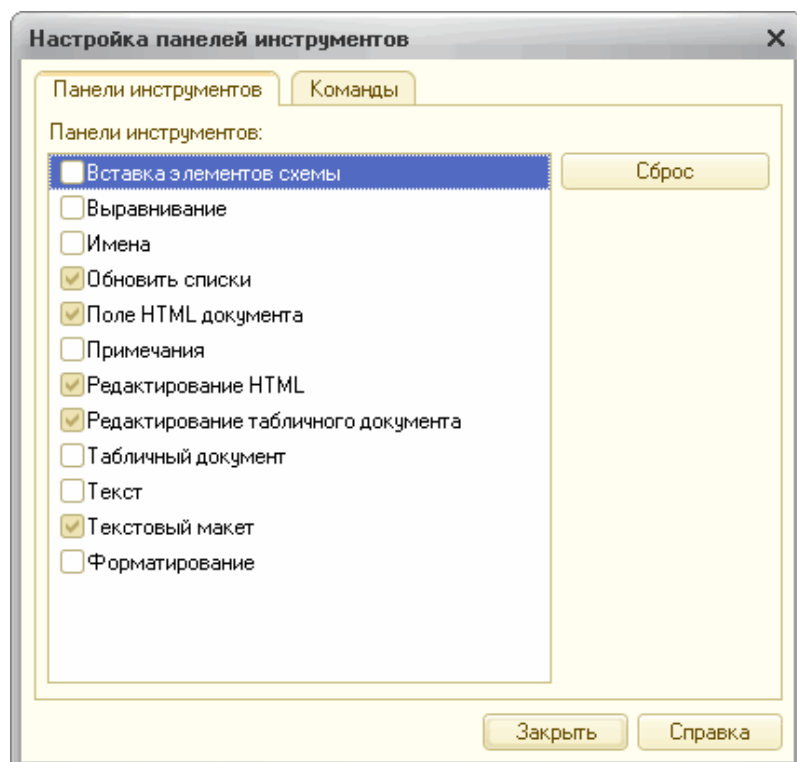


Рис. 199. Диалог настройки панелей инструментов

В списке панелей инструментов флажком отмечены панели, находящиеся в настоящий момент на экране. Установка флажка в строке с именем панели инструментов вызовет выбранную панель инструментов на экран, снятие – уберет ее.

При первом открытии этого диалога в списке панелей инструментов будут присутствовать только системные панели и панели, созданные в процессе настройки конфигурации.

Системные панели инструментов, измененные в результате редактирования, можно восстановить в первоначальном виде. Для этого необходимо выбрать имя системной панели в списке и нажать кнопку [Сброс](#).

Закладка «Команды»

При редактировании панели инструментов можно изменять состав и взаимное расположение кнопок, что позволяет организовать панели инструментов наиболее удобным образом. Редактирование выполняется при помощи закладки [Команды](#) диалога [Настройка панелей инструментов](#).

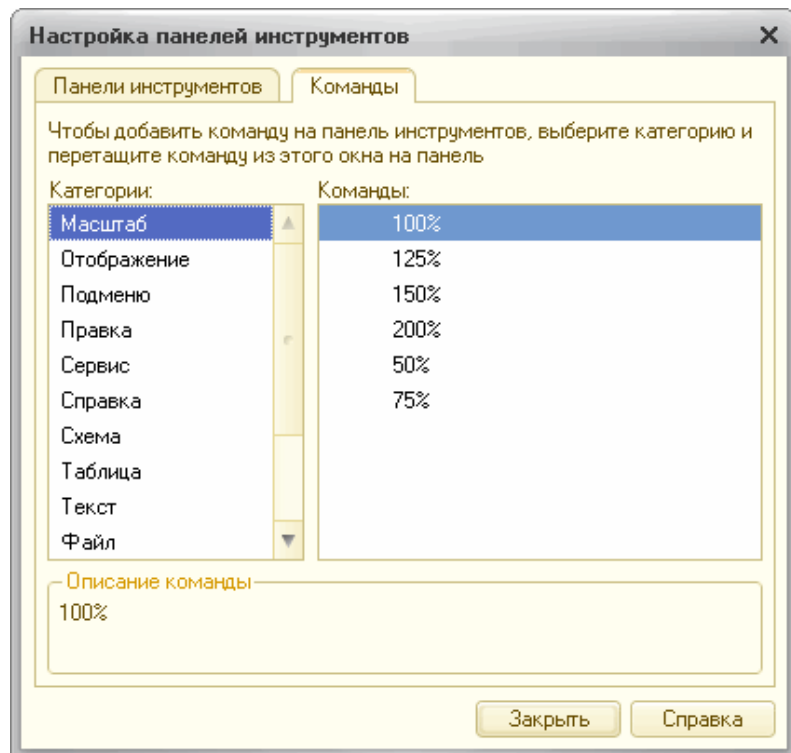


Рис. 200. Выбор команд из категории

Все кнопки, которые могут быть размещены на панелях инструментов, для удобства объединены в несколько категорий. Список категорий размещается в левой верхней части закладки.

Если выбрать в этом списке имя какой-либо категории, в правой части закладки выдается набор кнопок, входящих в эту категорию. Этот набор не может быть изменен.

Чтобы узнать назначение конкретной кнопки, необходимо щелкнуть на ней мышью – в нижней части закладки будет выдано описание кнопки.

Все действия по редактированию панели инструментов выполняются с помощью перемещения элементов панели мышью.

Для того чтобы добавить кнопку на панель инструментов, необходимо перетащить ее мышью на эту панель. Можно добавлять кнопки, перетаскивая их внутри панели и между панелями инструментов.

Для удаления кнопки с панели инструментов необходимо перетащить ее мышью в любое место окна программы.

11.2. Настройка формы

При работе в системе «1С:Предприятие» может быть доступна пользовательская настройка формы. Если иного не было установлено в конфигурации, то при вызове команды [Все действия – Изменить форму](#) на командной панели формы откроется окно редактора.

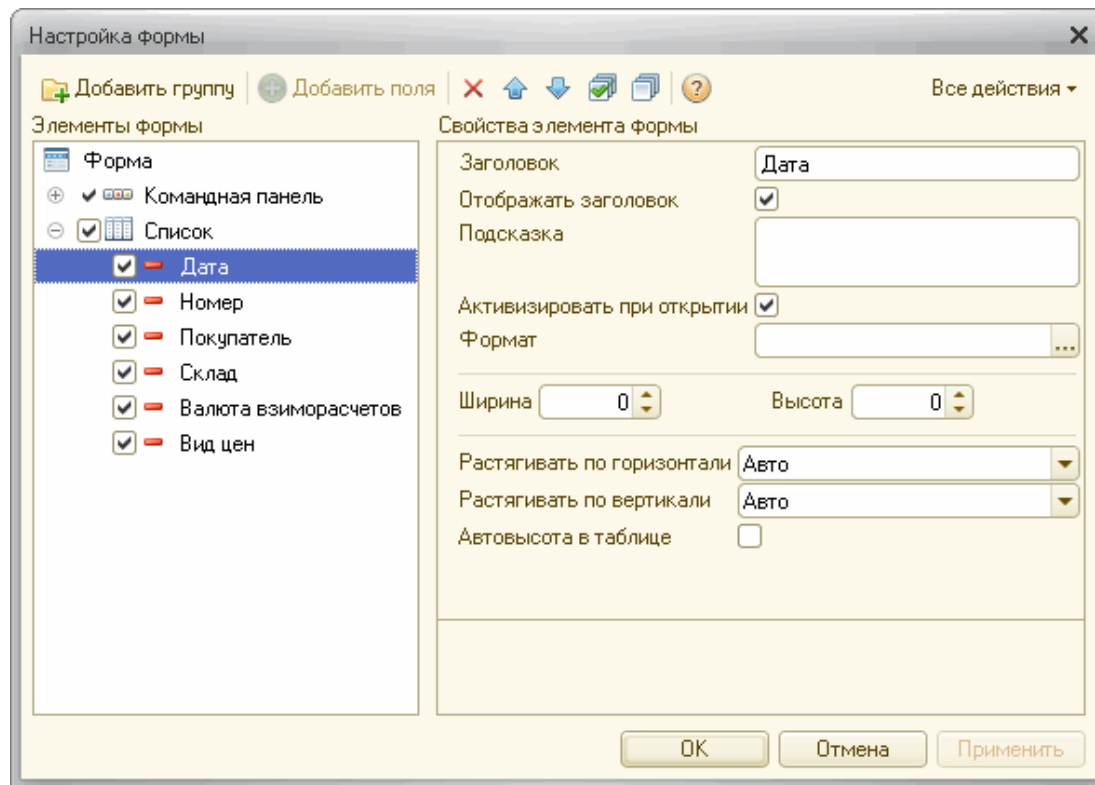


Рис. 201. Настройки формы

В левой части окна в виде дерева отображается структура элементов формы. В правой части – список свойств выбранного элемента формы.

Содержание и внешний вид формы определяются набором составляющих форму элементов нескольких видов.

- группы (их несколько видов),
- таблицы,
- поля (нескольких видов),
- кнопки.

Эти элементы обладают свойствами, влияющими на способ их отображения на форме. Иерархическая организация элементов определяет их взаимное расположение на форме.

Пользователь может изменять состав, взаимное расположение, группировку и свойства элементов.

Перестановка элементов. Чтобы изменить положение элементов в иерархии, используются кнопки [Вверх](#) и [Вниз](#). Или можно перетащить элемент на нужное место с помощью мыши.

Свойства элемента. Для каждого элемента свойства настраиваются в свойствах элемента формы. Описание каждого свойства отображается в нижней части панели при выборе свойства.

Настройка видимости. Чтобы изменить видимость элементов на форме (скрыть или показать), используются флажки напротив элементов. Следует отметить, что нельзя управлять видимостью корневого элемента [Форма](#) и кнопки настройки формы. Также не может быть изменена видимость командной панели, на которой расположена кнопка [Изменить форму...](#)

Создание групп. Чтобы добавить группу на форму, выберите в дереве структуры корневой элемент [Форма](#) и следует нажать кнопку [Добавить группу](#). В свойствах группы установите требуемый вид ([Обычная группа](#), [Страницы](#), [Командная панель](#)).

Добавление полей. Для некоторых полей может быть доступна кнопка [Добавить поля](#). Это означает, что у данных, отображаемых в поле/колонке, есть реквизиты, которые также могут быть отображены в поле/колонке. Для того чтобы добавить на форму реквизиты полей, следует выбрать элемент формы, нажать кнопку [Добавить поля](#) и в появившемся окне выбрать требуемые реквизиты.

Применение настроек. Применение настроек происходит при выходе из редактора настройки с помощью кнопки [ОК](#) или по команде [Применить](#). При этом настройки сохраняются в информационной базе, чтобы

впоследствии их можно было применить при открытии формы.

В процессе настройки можно отменить изменения, сделанные как в данном сеансе работы, так и ранее, и вернуться к настройкам формы, заданным в конфигурации. Для этого используется команда **Установить стандартные настройки** меню **Все действия**. Эта команда не вызывает немедленного изменения формы. После ее выполнения настройку можно продолжить. Применение настроек происходит только при нажатии кнопки **OK** или **Применить**.

11.2.1. Пример настройки формы

Рассмотрим процесс настройки формы на примере формы документа **Продажа**.

Продажа 000000012 от 16.03.2009 15:19:43 - Демонстрационная конфигурация "Управляемое прил... (1С:Предприятие)

Продажа 000000012 от 16.03.2009 15:19:43

Провести и закрыть | Провести | Создать на основании | Печать | Остатки по складу | Пересчитать | Все действия

Номер: 000000012 | Вид цен: Мелкооптовая

Дата: 16.03.2009 15:19:43

Организация: ООО "1000 мелочей"

Покупатель: Магазин "Бытовая техника" | Новый покупатель

Склад: Большой

Товары (2) | Прочее

+ Добавить | Подбор

N	Товар	Цена	Количество	Сумма
1	Veko67NE	12 000,00	4,00	48 000,00
2	Veko345MD	17 500,00	4,00	70 000,00

Количество (итог): 8,00 | Сумма (итог): 118 000,00

Рис. 202. Форма документа «Продажа»

Предположим, что в таблице **Товары** требуется отобразить артикул товара и поставщика, от которого был получен товар. Для этого на форме документа вызовем пункт **Все действия – Изменить форму**.

В форме настроек раскроем группу **Группа табличные части – Товары**, выберем поле **Товар** и нажмем кнопку **Добавить поля**, как показано на рисунке ниже.

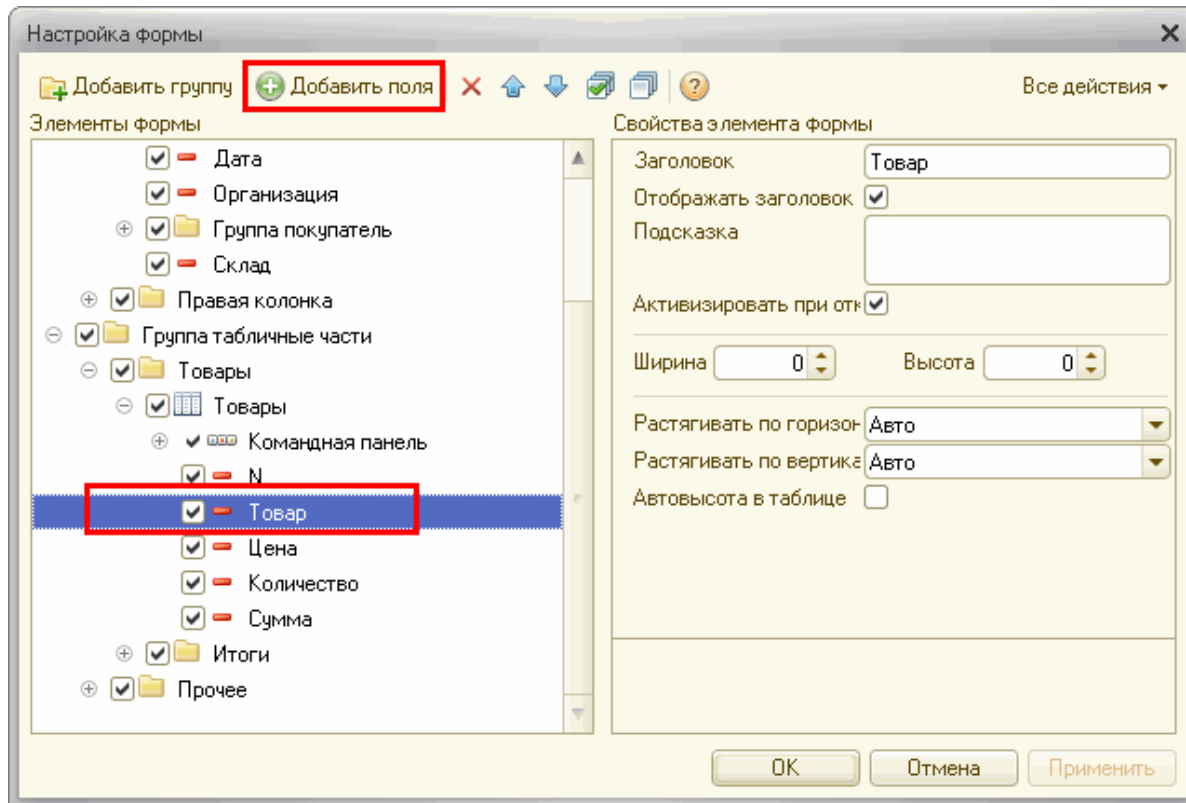


Рис. 203. Добавление полей в настройках формы

В появившемся окне выберем поля [Артикул](#) и [Поставщик](#), как показано на рисунке ниже.

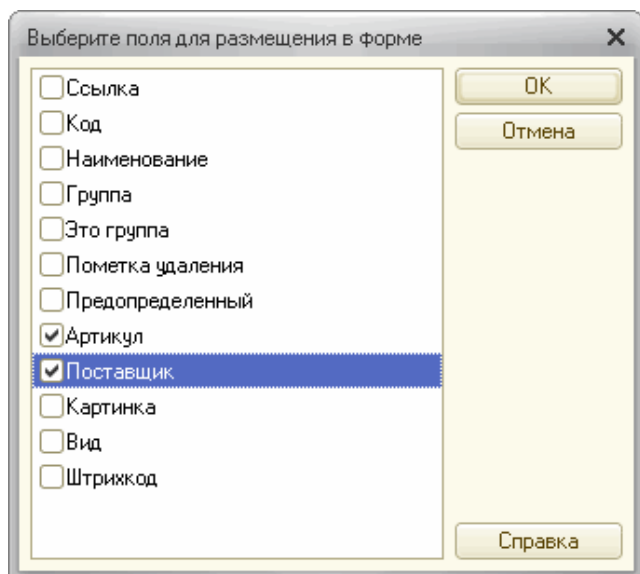


Рис. 204. Выбор полей

И нажмем **OK**. В диалоге настройки формы также нажмем **OK** или **Применить**. Форма документа [Продажа](#) примет вид:

Продажа 000000012 от 16.03.2009 15:19:43 - Демонстрационная конфигурация "Управляемое прил... (1С:Предприятие)

Продажа 000000012 от 16.03.2009 15:19:43

Провести и закрыть | Провести | Создать на основании | Печать | Остатки по складу | Пересчитать | Все действия

Номер: 000000012 | Вид цен: Мелкооптовая

Дата: 16.03.2009 15:19:43

Организация: ООО "1000 мелочей"

Покупатель: Магазин "Бытовая техника" | Новый покупатель

Склад: Большой

Товары (2) | Прочее

Добавить | Подбор

N	Товар	Артикул	Поставщик	Цена	Количество	Сумма
1	Veко67NE	4-0003	ЭлектроБыт ЗАО	12 000,00	4,00	48 000,00
2	Veко345MD	VEKO00001	ЭлектроБыт ЗАО	17 500,00	4,00	70 000,00

Количество (итог): 8,00 | Сумма (итог): 118 000,00

Рис. 205. Форма документа «Продажа»

Предположим, что требуется отображать, из какого региона поставщик товара. Для этого в настройках формы выберем поле **Поставщик** и нажмем кнопку **Добавить поля**. В открывшемся окне выберем поле **Регион** и нажмем **OK**.

Структура таблицы примет следующий вид:

- Группа табличные части
 - Товары
 - Товары
 - Командная панель
 - N
 - Товар
 - Артикул (Товар)
 - Поставщик (Товар)
 - Регион (Товар, Поставщик)
 - Цена
 - Количество
 - Сумма

Рис. 206. Структура таблицы в документе «Продажа»

Сгруппируем колонки таблицы. Для этого в таблице **Товары** создадим две группы (нажатием кнопки **Добавить группу**). В одной группе будет отображаться информация о товаре: поля **Товар** и **Артикул**. В другой – информация о поставщике: поля **Поставщик** и **Регион**. Установим порядок отображения полей в колонках с помощью стрелок панели формы настройки.

Структура таблицы документа примет вид:

- ☐ Группа табличные части
- ☐ Товары
 - ☐ Товары
 - ⊕ Командная панель
 - N
 - ☐ Группа
 - Товар
 - Артикул (Товар)
 - ☐ Группа1
 - Поставщик (Товар)
 - Регион (Товар, Поставщик)
 - Цена
 - Количество
 - Сумма
- ⊕ Итоги

Рис. 207. Структура таблицы в документе «Продажа»

А измененная форма будет выглядеть, как показано на рисунке ниже.

Продажа 000000012 от 16.03.2008 15:19:43 · Демонстрационная конфигурация "Управляемое прил... (1С:Предприятие)

Продажа 000000012 от 16.03.2008 15:19:43

Провести и закрыть | Провести | Создать на основании | Печать | Остатки по складу | Пересчитать | Все действия

Номер: 000000012 | Вид цен: Мелкооптовая

Дата: 16.03.2008 15:19:43

Организация: ООО "1000 мелочей"

Покупатель: Магазин "Бытовая техника" | Новый покупатель

Склад: Большой

Товары (2) | Прочее

Добавить | Подбор | Все действия

N	Товар	Поставщик	Цена	Количество	Сумма
	Артикул	Регион			
1	Veko67NE 4-0003	ЭлектроБыт ЗАО Урал	12 000,00	4,00	48 000,00
2	Veko345MO VEKO00001	ЭлектроБыт ЗАО Урал	17 500,00	4,00	70 000,00

Количество (итог): 8,00 | Сумма (итог): 118 000,00

Рис. 208. Форма документа «Продажа»

При изменении количества товара в колонке **Количество** сумма по строке и итоговая сумма по документу автоматически пересчитываются. А значит, кнопка **Пересчитать** не нужна. Скроем кнопку **Пересчитать** в настройках формы, сняв флажок видимости напротив кнопки **Пересчитать**, как показано на [рис. 209](#), и нажмем **OK**.

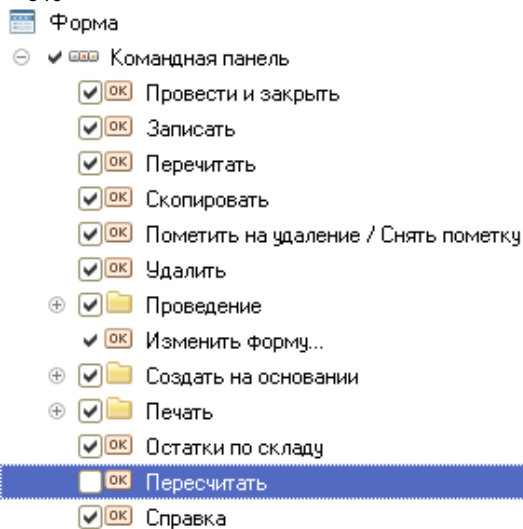


Рис. 209. Установка невидимости для кнопки «Пересчитать»

Командная панель формы документа примет следующий вид:



Рис. 210. Шапка документа «Продажа»

По документам **Продажа** проходят в основном неделимые предметы. И значит, точность до двух знаков после запятой в колонке **Количество (Итог)** нам не нужна.

В настройках формы выберем поле **Количество (итог)** и в свойстве **Формат** нажмем кнопку выбора, как показано на рисунке.

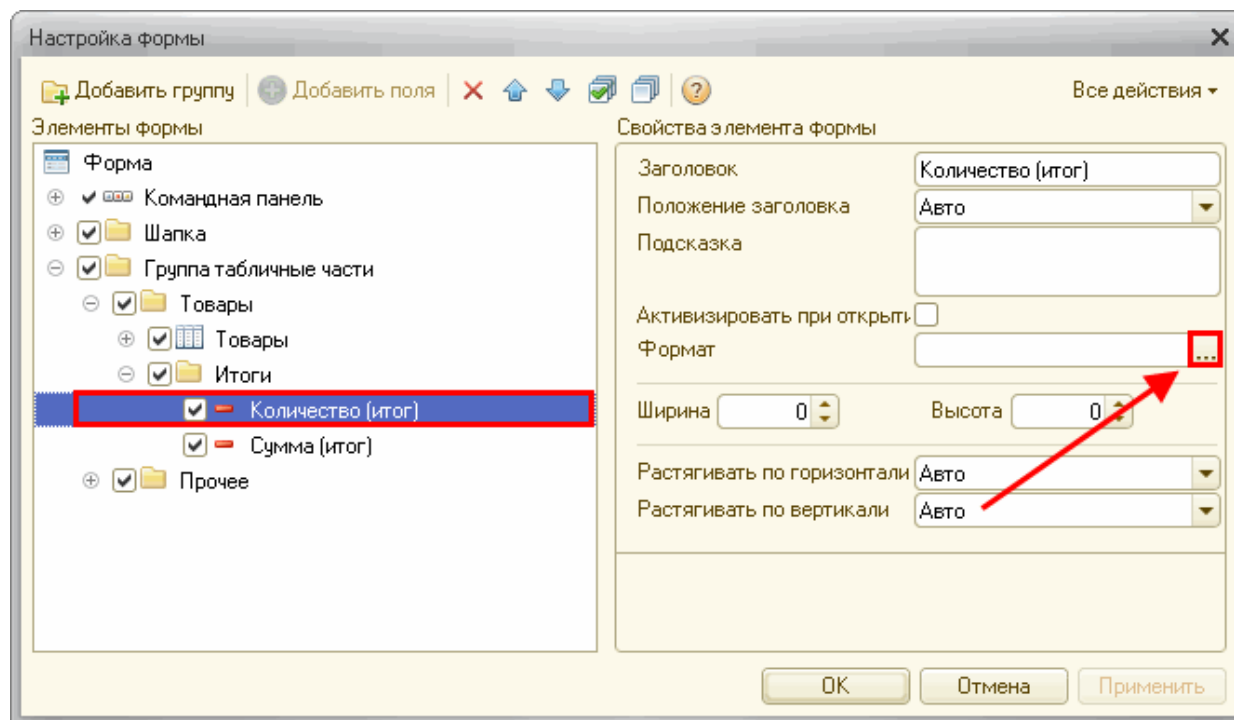


Рис. 211. Вызов конструктора форматной строки

В открывшемся окне конструктора форматной строки установим флажок напротив параметра **Точность** и нажмем **OK**. Применим настройки формы.

После применения всех перечисленных выше настроек документ **Продажа** будет выглядеть следующим образом:

Продажа 000000012 от 16.03.2008 15:19:43

Провести и закрыть Провести Создать на основании Печать Остатки по складу Все действия

Номер: Вид цен:

Дата:

Организация:

Покупатель:

Склад:

Товары (2) Подбор Все действия

N	Товар	Поставщик	Цена	Количество	Сумма
	Артикул	Регион			
1	Veko67NE	ЭлектроБьт ЗАО	12 000,00	4,00	48 000,00
	Ч-0003	Урал			
2	Veko345MO	ЭлектроБьт ЗАО	17 500,00	4,00	70 000,00
	VEKO00001	Урал			

Количество (итог): **8** Сумма (итог): **118 000,00**

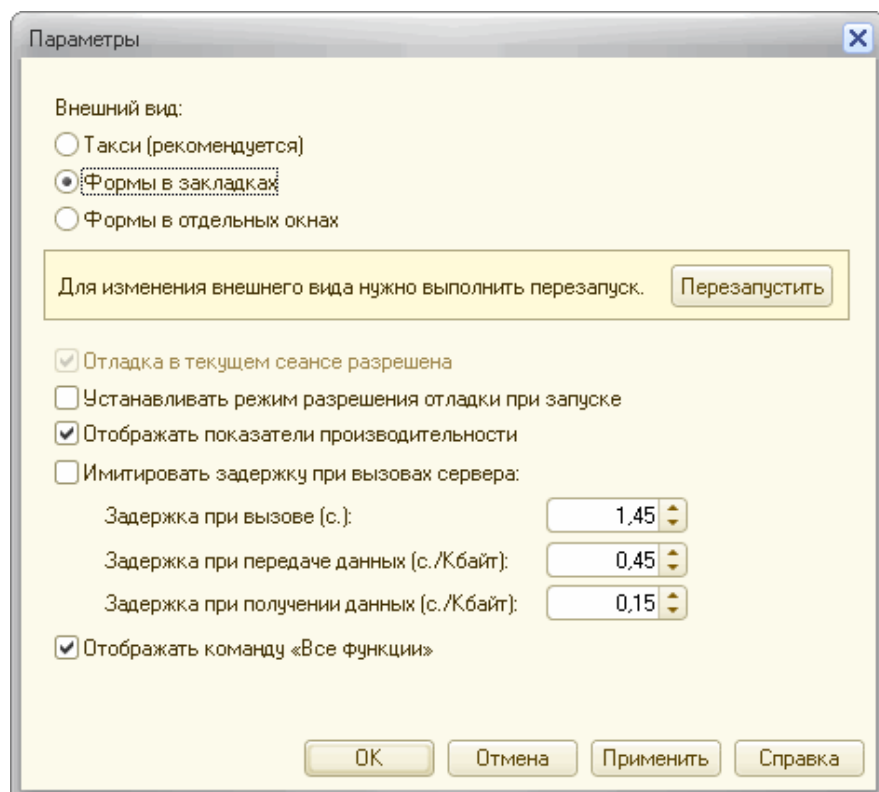
Рис. 212. Результирующая форма документа «Продажа»

Таким образом, в режиме **1С:Предприятие** можно группировать связанные по смыслу элементы, располагать их в удобном для себя порядке, отключать ненужные для работы элементы, изменять способ отображения элементов на форме.

11.3. Системные параметры

Диалог установки системных параметров предназначен для выбора внешнего вида программы и настройки режимов отладки разработчиком или администратором.

Для вызова диалога настройки параметров системы «1С:Предприятие» используется команда главного меню **Сервис – Параметры**.



В диалоге возможна настройка следующих параметров:

- **Внешний вид** – Чтобы параметр вступил в силу, требуется перезапуск программы. Доступны следующие варианты:
 - **Такси (рекомендуется)** – доступность данного внешнего вида зависит от настроек программы. Подробнее об особенностях работы в интерфейсе «Такси» см. книгу «1С:Предприятие 8.3. Руководство пользователя. Интерфейс «Такси»;
 - **Формы в закладках** – Подробнее о данном режиме открытия форм см. раздел «[Формы в закладках](#)», [здесь](#);
 - **Формы в отдельных окнах** – подробнее о данном режиме открытия форм см. раздел «[Формы в отдельных окнах](#)», [здесь](#);
- **Отладка в текущем сеансе разрешена** – если флажок установлен, то в исполняемом экземпляре «1С:Предприятия» разрешена отладка;
- **Устанавливать режим разрешения отладки при запуске** – если флажок установлен, то отладка будет разрешена при последующем запуске экземпляров приложения;
- **Отображать показатели производительности** – если установлен флажок, то в отдельном специальном окне, которое по умолчанию расположено в левом нижнем углу экрана, будут отображаться показатели производительности;
- **Имитировать задержку при вызовах сервера** – данный параметр используется для проверки работы конфигурации в условиях соединения по медленным каналам;
- **Отображать команду "Все функции"** – если данный флажок установлен, то в меню добавляется команда **Все функции**, с помощью которой возможно выполнение различных действий. В список включены все объекты конфигурации независимо от того, включены ли они в интерфейс системы или нет. Данная настройка доступна, если пользователь имеет право на использование режима **Все функции**.

Подробнее о влиянии значений системных параметров см. в книге «1С:Предприятие 8.3. Руководство разработчика».

11.4. Управление окнами

В процессе работы помимо основного окна может быть открыто несколько вспомогательных. Платформа предоставляет механизм управления окнами и перехода между ними.

Для вызова диалога управления окнами нужно выбрать пункт **Окна – Все окна** главного меню. В диалоге отображается список окон, открытых к настоящему моменту.

Чтобы перейти к требуемому окну, нужно выбрать его в списке и нажать кнопку **Перейти**. Диалог **Окна** закроется автоматически.

Чтобы закрыть несколько окон, следует выбрать их с помощью мыши, удерживая клавишу **Shift**, и нажать кнопку **Закрыть окна**. Основное окно не может быть закрыто с помощью этого диалога.

Также для переключения между окнами одного сеанса используются сочетания клавиш **Ctrl + Tab**, **Ctrl + Shift + Tab**, **Ctrl + Shift + F6**. Для перехода к следующему активному окну следует нажать клавиши **Ctrl + Tab** или **Ctrl + F6**, для перехода к предыдущему окну – **Ctrl + Shift + Tab** или **Ctrl + Shift + F6**. При этом учитываются открытые модальные и блокирующие окна – окно, перекрытое блокирующим окном, активировано не будет.

В режиме интерфейса с закладками в списке окон отображаются и открытые закладки, но их активность в списке не отображается. Подробнее о режиме интерфейса с закладками см. раздел «[Формы в закладках](#)», [здесь](#).

11.4.1. Окна текстовых и табличных документов

Система «1С:Предприятие» помимо различных форм, предназначенных для учетных целей, предоставляет возможность работы с текстовыми и табличными документами (отчеты и созданные пользователем табличные документы). Для этих типов окон становятся доступными пункты меню, предоставляющие дополнительный сервис при просмотре и редактировании.

Пункты [Горизонтально разделить окно](#) и [Вертикально разделить окно](#) позволяют в пределах одного окна просматривать различные части данного окна. Пункты меню доступны, только если активным является окно с текстовым или табличным документом. Выбранное окно разделяется на две одинаковые части. Чтобы изменить соотношение частей, нужно подвести указатель мыши к разделителю окна. В зависимости от выбранного вида разделения указатель мыши сменит свою форму на раздвоенную вертикальную или горизонтальную стрелку. Нажав левую клавишу и перемещая мышь, изменить размеры подокон.

Отмена разделения окна редактора текстов. Чтобы отменить разделение окна редактора текстов, нужно выбрать пункт [Окна – Удалить разделение окна](#) или перетащить мышью разделительную линию к одному из краев окна редактора.

11.4.2. Восстановление положения окна

Положение, размер и состояние окон можно изменять. При закрытии последние параметры отображения окна сохраняются, и при повторном открытии окно открывается с последними значениями параметров показа.

Для восстановления первоначального положения, размера и состояния окна следует в главном меню выбрать пункт [Окна – Восстановить положение окна](#). При этом запомненные значения параметров показа сбрасываются и восстанавливаются первоначальные, какие были при первом открытии.

Глава 12. Получение справочной информации

Информацию по работе с системой «1С:Предприятие» и по конкретным объектам и режимам можно получить, используя систему справочной информации (пункт [Справка](#) или клавиша **F1**). В главном меню для вызова помощи существует подменю [Справка](#).

При вызове справки из какого-либо режима отображается раздел справочной информации, соответствующий тому режиму, в котором сейчас работает пользователь. Пункт меню [Содержание](#) и клавиши **Shift + F1** позволяют обратиться к общему содержанию справочной информации.

В режиме интерфейса [В отдельных окнах](#) справка откроется в отдельном окне.

В режиме интерфейса [В закладках](#) окно справки откроется на новой закладке.

12.1. Окно справки

Окно справочной информации состоит из двух частей. Вверху расположена командная панель, с помощью которой осуществляется навигация и поиск информации. Ниже командной панели отображается содержание справки.

Если текст не помещается в окне целиком, справа и внизу окна появляются линейки прокрутки, которые можно использовать для вывода на экран невидимой части.

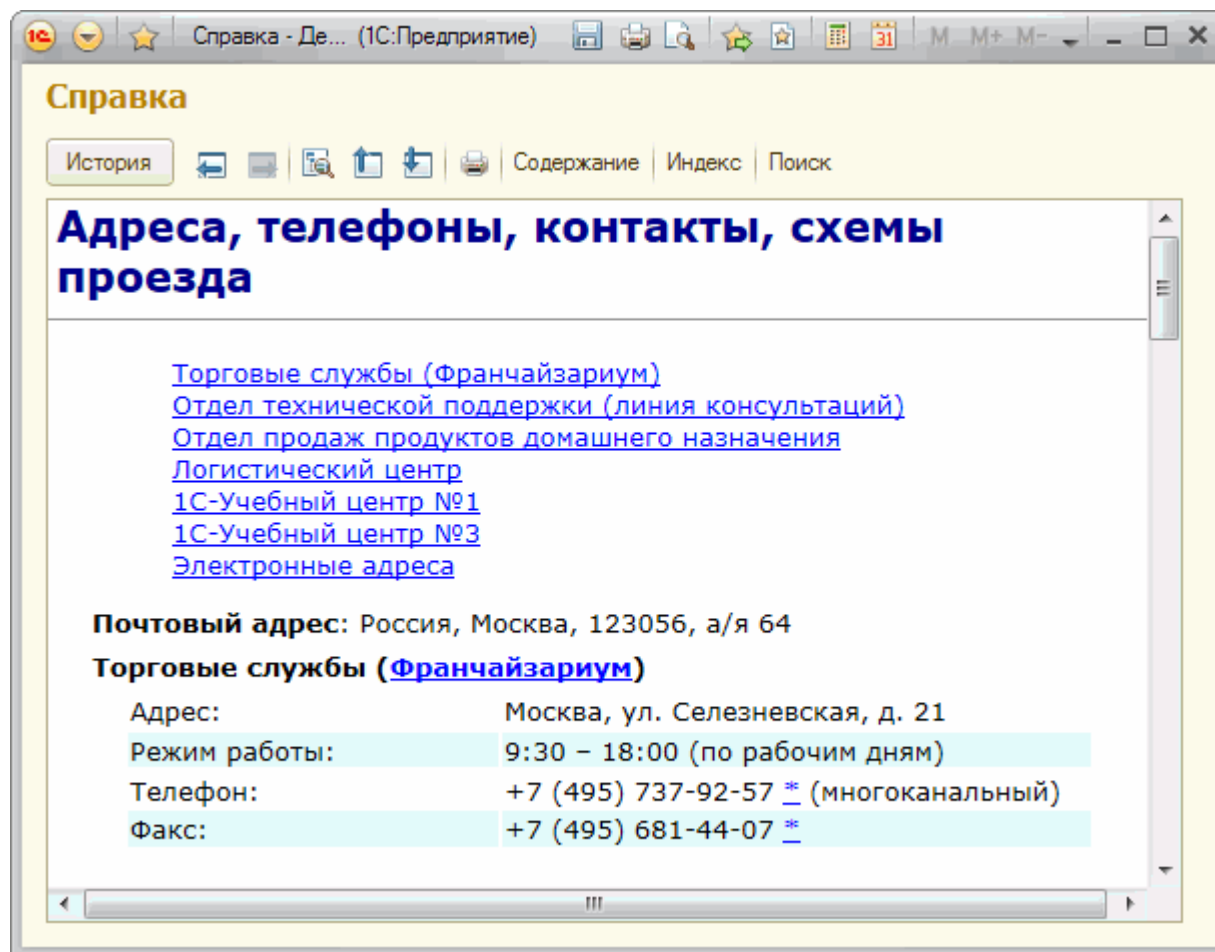


Рис. 214. Окно справочной информации

12.1.1. Просмотр справочной информации

Для поиска нужной главы можно воспользоваться содержанием. Для этого в окне [Справка](#) следует нажать кнопку [Содержание](#).

В левой части окна появится дерево глав содержания.

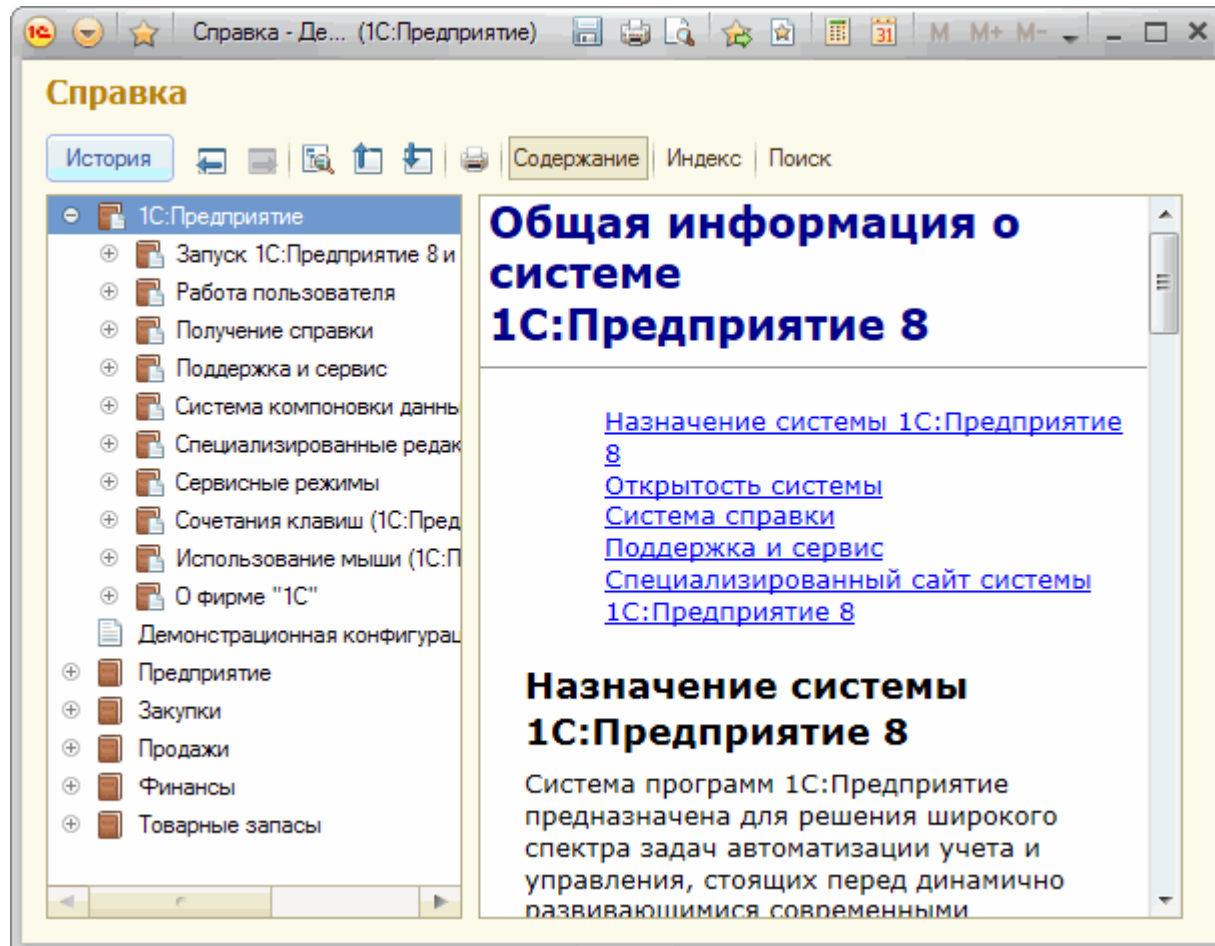


Рис. 215. Содержание справки

Дерево глав строится автоматически и включает в себя системную справку и справку по использованию конфигурации. Также справка может содержать информацию по работе с конкретными разделами программы.

Для удобства просмотра все дерево или отдельные ветви можно развернуть и свернуть.

Пиктограмма + (плюс) в узле ветви указывает, что эту ветвь можно развернуть. Для этого достаточно нажать пиктограмму или дважды нажать мышью наименование главы, расположенное возле пиктограммы. Ветвь откроет для просмотра следующий уровень, а пиктограмма + (плюс) изменится на - (минус). Свернуть ветвь дерева можно щелчком мыши на пиктограмме - (минус).

Чтобы развернуть сразу все ветви дерева, нужно выбрать клавишами или мышью наименование главы корневой ветки дерева и нажать клавишу * (звездочка на цифровой клавиатуре), которая расположена на цифровой клавиатуре.

На первом уровне в дереве располагаются названия разделов.

Последовательно разворачивая ветви дерева глав, можно перейти к описанию конкретного элемента данных.

Для перемещения по главам описания можно использовать дерево глав или кнопки панели управления окна [Справка](#).

Чтобы просмотреть текст интересующей главы в дереве, следует выбрать наименование главы мышью или с помощью клавиш управления курсором и нажать клавишу [Enter](#). В окне просмотра отобразится текст выбранной главы.

Для перемещения по тексту описания можно использовать клавиши управления курсором и клавиши [PgDn](#) и [PgUp](#).

При значительном объеме текста можно использовать линейки прокрутки, расположенные справа и внизу окна, для вывода на экран невидимой части описания.

Кнопки командной панели окна [Справка](#) позволяют быстро перемещаться от одной главы к другой.

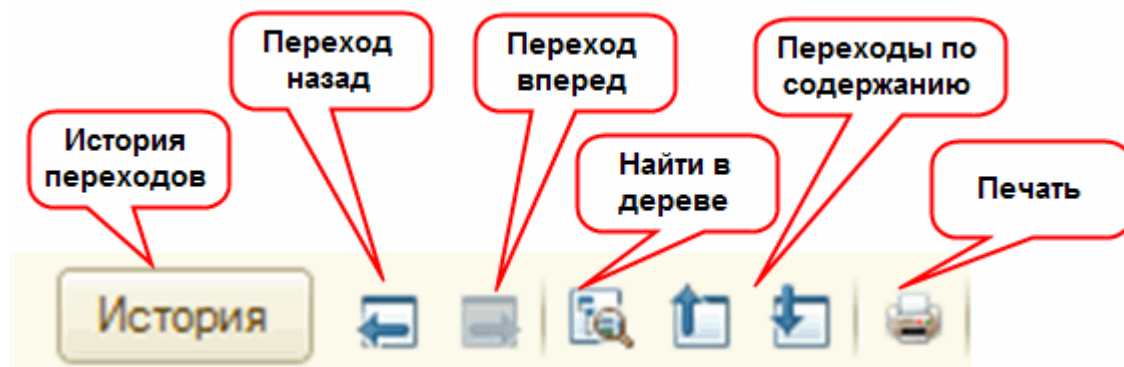


Рис. 216. Командная панель окна справочной информации

Кнопкой [Переход к следующему](#) осуществляется переход к следующей по порядку главе. В дереве глав при необходимости будет открыта соответствующая ветвь, а выделение будет перемещаться по названиям глав, показывая текущее положение в дереве. Для перехода к предыдущей по порядку главе используется кнопка [Переход к предыдущему](#).

После того как глава описания найдена и окно справки открыто, для поиска положения главы в дереве описаний используется кнопка командной панели [Найти текущий элемент в дереве](#).

Если в процессе просмотра выбирались несколько страниц, то с помощью команд [Переход вперед](#), [Переход назад](#), и [История](#) можно вернуться к просмотренным страницам.

Программа сохраняет историю просмотра справочных глав. Для возврата на предыдущую просмотренную главу или для перехода на следующую просмотренную главу в истории используются кнопки истории переходов.

12.1.2. Поиск справочной информации

Для поиска нужной информации в описании существует несколько способов, которые рассмотрены ниже.

Поиск по индексу. Окно [Справка](#) предоставляет поиск справочной информации по индексу. На командной панели нужно выбрать команду [Индекс](#). В левой части окна отобразится индекс разделов справки.

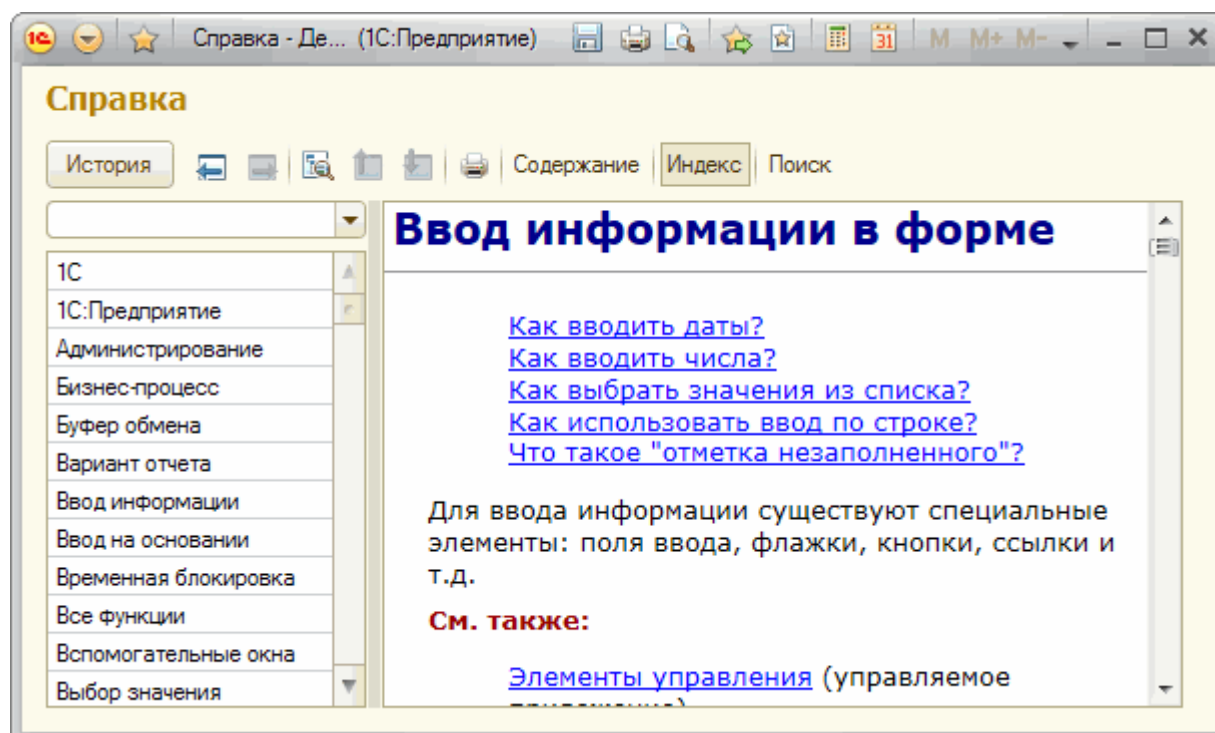
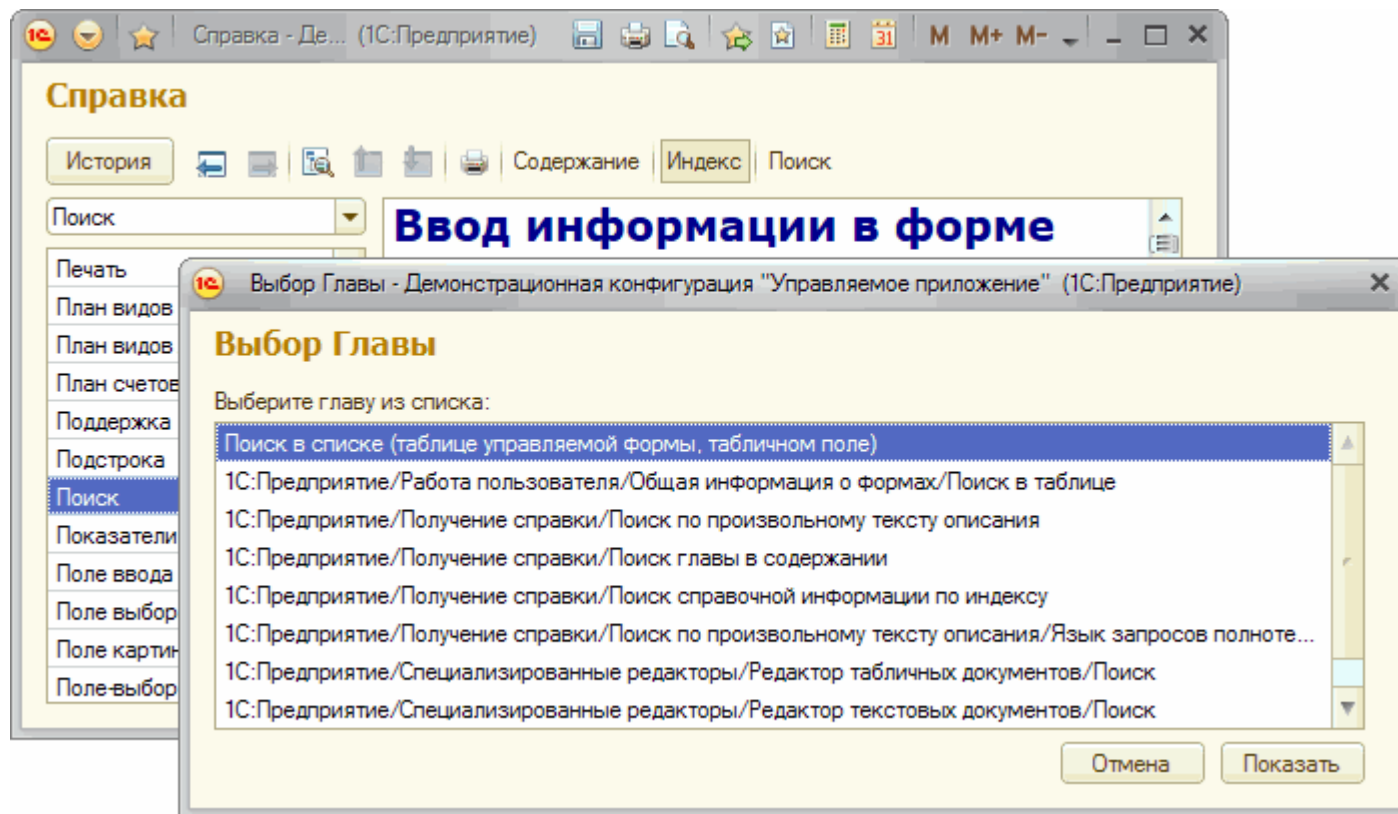


Рис. 217. Вызов индекса справки

В верхней строке находится поле ввода, предназначенное для написания наименования главы. Ниже расположен отсортированный список глав. Если начать набирать название главы, то по мере набора символов будет выделяться глава, первые символы названия которой совпадают с вводимыми.

Если нужная глава найдена, следует нажать клавишу **Enter**. В окне **Справка** будет выведено содержимое справки по данной главе.

Если по данной главе в содержании имеется несколько описаний, то их список выводится в окно **Выбор главы**.

**Рис. 218. Выбор главы справки**

Следует указать нужную главу и нажать кнопку **Показать**.

СОВЕТ. Также для вызова индекса справки используется сочетание клавиш **Alt + Shift + F1**.

Поиск по справке. Для поиска глав, в которых встречается указанная строка, нужно выбрать пункт **Справка – Поиск по справке**. На экран выводится окно, в котором выполняется поиск по произвольному тексту описания. В верхней части содержится поле для ввода строки поиска и поле списка найденных глав описаний.

Для начала поиска следует ввести текст и нажать кнопку **Поиск** (или клавишу **Enter**). Программа выполняет поиск глав, в которых встречается введенный текст. Регистр ввода может быть любым, слова текста учитываются целиком (если не использован оператор *****), с учетом морфологии. Допускается использование поисковых операторов (см. главу [«Формат поисковых выражений»](#), [здесь](#)).

В процессе ввода программа выводит список этих глав. Если введенный текст нигде не встречается, то ниже поля ввода программа выводит об этом сообщение.

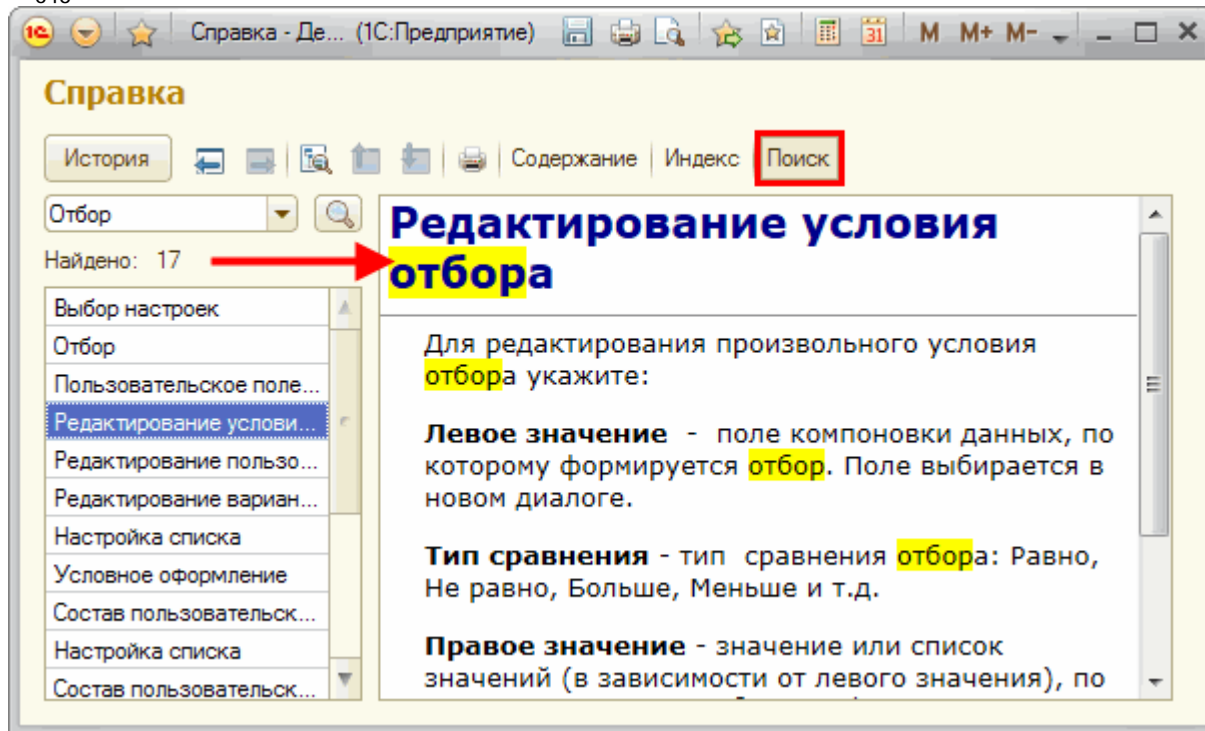


Рис. 219. Поиск по справке

При открытии главы программа покажет описание таким образом, чтобы было видно первое вхождение указанного текста.

Для просмотра главы нужно выбрать ее в списке и нажать клавишу **Enter**. Описание выбранной главы показывается в соседнем поле.

СОВЕТ. Также для вызова поиска по справке можно использовать сочетание клавиш **Alt + F1**.

Список операторов, используемых при поиске по справке, приведен в главе «[Формат поисковых выражений](#)» [здесь](#).

12.1.3. Печать описания

Для вывода любой главы описания на печать необходимо выполнить следующие действия:

- В окне описания выбрать главу для печати и нажать в область с текстом, где отобразится текстовый курсор.
- Выделить нужный текст и запомнить его в буфере обмена.
- Вставить запомненную часть в любой текстовый редактор и распечатать

12.2. Окно «О программе»

Окно **О программе** вызывается выбором пункта **Справка – О программе** в главном меню или нажатием кнопки **i** области системных команд.



Рис. 220. Вызов окна «О программе»

Окно предоставляет общую информацию о системе «1С:Предприятие» и условиях конкретного запуска системы.

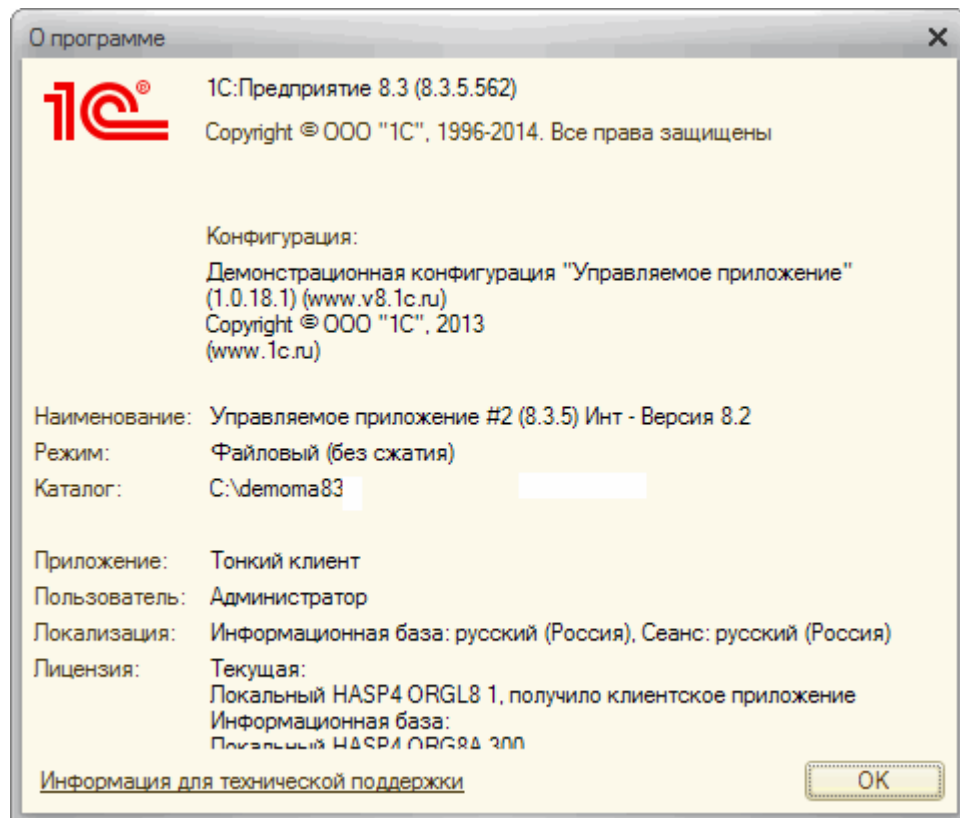


Рис. 221. Окно «О программе»

В окне отображаются следующие данные:

- информация о текущей версии системы «1С:Предприятие»;
- информация о конфигурации, с которой в настоящий момент работает пользователь (наименование и версия конфигурации);
- сведения об информационной базе, с которой в настоящий момент работает пользователь:
 - имя информационной базы, как оно задано в списке информационных баз диалога запуска «1С:Предприятия»;
 - формат базы данных (формат хранения информации) – файловый или клиент-сервер;
 - каталог, в котором располагается информационная база, и сведения о региональных установках информационной базы;
 - приложение (режим запуска: [Толстый клиент](#), [Тонкий клиент](#) и [Веб-клиент](#)). Также указывается скорость соединения и режим сжатия данных ([без сжатия](#), [обычное](#), [усиленное](#));
 - имя пользователя, который работает с программой в данный момент. Если задано полное имя текущего пользователя и оно не совпадает с кратким именем, имя текущего пользователя показывается в формате [Краткое имя](#); [Полное имя](#);
 - данные о локализации информационной базы и сеанса;
 - данные о текущей лицензии клиента и сервера «1С:Предприятия», а также данные обо всех лицензиях, работавших с данной информационной базой и изменявших данную конфигурацию;
- гиперссылка [Информация для технической поддержки](#), при нажатии которой в новом окне отображается информация о программе и журнал ошибок.

Приведенную в окне информацию можно копировать в буфер обмена с помощью команды контекстного меню.

Глава 13. Формат поисковых выражений

Поиск может осуществляться по нескольким словам, с использованием поисковых операторов и поиском по точной фразе.

По умолчанию поиск с учетом синонимов и нечеткий поиск не производятся. Для выполнения поисков этих видов следует использовать соответствующие операторы.

В данной таблице приведены поисковые операторы, которые можно использовать как при поиске по справке, так и при полнотекстовом поиске в данных. Некоторые операторы можно использовать только для одного вида поиска (см. описание конкретных операторов в таблице).

Оператор	Пример	Описание
Пробел И AND &	запись документ запись И документ запись AND документ запись & документ	Будут найдены все разделы, содержащие и «запись», и «документ» с учетом морфологии
ИЛИ OR ,	запись ИЛИ документ запись OR документ запись документ запись , документ	Будут найдены все разделы, содержащие хотя бы одно из слов «запись» или «документ» с учетом морфологии
НЕ NOT ~	закрытие НЕ месяц закрытие NOT месяц закрытие ~ месяц	Будут найдены все разделы, содержащие слово «закрытие», но не содержащие слова «месяц» с учетом морфологии. Использование знака ~ в начале строки не допускается
РЯДОМ/[±]n NEAR/[±]n	Пример 1: право РЯДОМ/3 пользователь Пример 2: право РЯДОМ/+3 пользователь Пример 3: право РЯДОМ/-3 пользователь	<p>Поиск раздела, содержащего указанные слова с учетом морфологии на расстоянии n слов между словами.</p> <p>Знак указывает, в каком направлении от первого слова будет искаться второе слово</p> <p>(«+» – после первого; «-» – до первого слова).</p> <p>Если знак не указан, то будет найден текст, содержащий указанные слова на дистанции n слов друг от друга. Порядок слов не имеет значения.</p> <p>В примере 1 будут найдены разделы, в которых слово «пользователь» находится не более трех слов до или после слова «право».</p> <p>В примере 2 будут найдены разделы, в которых слово «пользователь» находится не более трех слов после слова «право».</p> <p>В примере 3 будут найдены разделы, в которых слово «пользователь»</p>

		находится не более трех слов перед словом «право»
РЯДОМ NEAR	Библиотека РЯДОМ имени РЯДОМ Достоевского	Краткая форма. Запрос в короткой форме найдет разделы, в которых слова встречаются не больше, чем на 8 слов друг от друга в любую сторону
«»	«проведение документа»	Поиск точной фразы с учетом морфологии (эквивалентно «проведение РЯДОМ/+1 документ»)
()	(проведение выписка) & (счета, документа)	Группировка слов (сколько угодно уровней вложенности)
*	доку*	Поиск с использованием группового символа (замена окончания слова). Должно быть введено более одного значащего символа. Запрос «доку*» найдет «документ», «документировать», «документальный» и др. (Если в индексе поиска окажется более 300 слов, начинающихся с «доку», система попросит уточнить запрос.) В поиске по справке знак * может быть в любом месте слова и сколько угодно раз, например «*ОК*нт*». При полнотекстовом поиске поддерживается использование только одного символа «*», и он должен быть в конце слова. То есть запрос «арх*» найдет «архив», «археология»
#	#Система Система#2	Нечеткий поиск слов с заданным количеством отличий от указанного в строке поиска. Запрос «#Система» (эквивалентно запросу «Система#1») найдет «система», «сивтема». Запрос «Система#2» найдет «ситтама», «сеттема». Данный оператор используется при полнотекстовом поиске и не используется при поиске по справке
!	!красный кафель	Поиск с учетом синонимов русского, английского и украинского языков. Оператор «!» ставится перед соответствующим словом. Пример: поиск «!красный кафель», найдет еще и «алый кафель», и «коралловый кафель». Данный оператор используется при полнотекстовом поиске и не используется при поиске по справке

Если не указано никаких операторов (слова набраны через пробел), то программа осуществляет поиск всех слов из запроса с использованием оператора И.

ПРИМЕЧАНИЕ 1. Написание операторов И (AND), ИЛИ (OR), НЕ (NOT), РЯДОМ (NEAR) допускается только в верхнем регистре.

ПРИМЕЧАНИЕ 2. Операторы не используются как унарные (в начале строки поиска). Например, нельзя сделать выбор всех глав, в которых отсутствует указанный текст.

ПРИМЕЧАНИЕ 3. Все символы в поле поиска, кроме символов поисковых операторов, букв и цифр, игнорируются. Для поиска спецсимволов, используемых в тексте, их следует заключать в кавычки. Например: "РЯДОМ".

Глава 14. Работа с веб-клиентом

Данная глава содержит сведения о работе с веб-клиентом «1С:Предприятия».

Для того чтобы иметь доступ к данным информационной базы с помощью веб-клиента, информационная база должна располагаться на специальном сервере.

Веб-клиент работает в среде веб-браузера. Поддерживаются:

- Mozilla Firefox версии 17 или более поздней для Microsoft Windows и Linux;
- Microsoft Internet Explorer версии 8.0 и выше для Microsoft Windows. При работе в Windows XP рекомендуется использовать версию 8.0. При работе в Windows Vista и Windows 7 – версию 10.0.
- Google Chrome версии 4 и выше для Microsoft Windows;
- Safari версии 4.0.5 и выше для Mac OS X (версии 10.5 и выше);
- мобильная версия Safari для устройств iPad в среде iOS версии 3.2, 4.2 и выше. Рекомендуется использовать версию 4.2 и выше.

ВНИМАНИЕ! Для работы с веб-клиентом в операционной системе Windows XP SP 2 требуется наличие системной библиотеки `jscrip.dll` версии 5.6.0.8834 или выше. Библиотека находится в каталоге `system32` операционной системы. Если на вашем компьютере установлена более ранняя версия библиотеки, то возможно значительное замедление работы веб-клиента. Для обновления версии необходимо скачать пакет обновлений, находящийся по адресу <http://support.microsoft.com/kb/942840/en-us>.

14.1. Запуск веб-клиента

Для запуска веб-клиента необходимо в адресной строке браузера ввести адрес информационной базы следующего вида:

<http://host>

Где `host` – адрес сайта, соответствующего информационной базе.

При загрузке информационной базы в адресную строку автоматически добавится конструкция символов вида `/aa_BB`, где `aa` – идентификатор языка интерфейса, `BB` – идентификатор страны.

При этом браузер должен быть настроен соответствующим образом. О настройке браузера см. разделы «[Настройка Microsoft Internet Explorer](#)» [здесь](#), «[Настройка Mozilla Firefox](#)» [здесь](#), «[Настройка Google Chrome](#)» [здесь](#) и «[Настройка Safari](#)» [здесь](#).

Можно запустить веб-клиент и одновременно перейти по внешней ссылке. При вводе внешней ссылки в адресную строку браузера произойдет загрузка приложения и открытие данных ссылки.

Подробнее о механизме ссылок см. раздел «[Ссылки](#)» [здесь](#).

ПРИМЕЧАНИЕ. При работе в браузерах Internet Explorer внешние ссылки рекомендуется открывать в пустом окне браузера.

При первом запуске веб-клиента программа проверит срабатывание блокировщика всплывающих окон веб-браузера. Если требуемые настройки веб-браузера не установлены:

- для всех веб-браузеров, кроме Mozilla Firefox и Google Chrome, будет выведена инструкция по настройке браузера, и запуск веб-клиента будет прекращен;
- для веб-браузера Mozilla Firefox будет выведен диалог настройки, с помощью которого можно установить все настройки, необходимые для работы веб-клиента. Диалог содержит описание необходимых настроек и две кнопки – [Разрешить](#) и [Завершить работу](#). При нажатии кнопки [Разрешить](#) в настройке браузера вносятся необходимые изменения и продолжается загрузка приложения. При нажатии кнопки [Завершить работу](#) работа приложения завершается;
- для веб-браузера Google Chrome отобразится окно предупреждения о блокировании всплывающего окна. При нажатии кнопки [Открыть инструкцию](#) в новой вкладке откроется инструкция по настройке веб-браузера.

Эта проверка применяется при первом запуске, если используется интерфейс [В отдельных окнах](#) и используются модальные окна.

14.1.1. Окно веб-клиента

Командный интерфейс приложения в веб-клиенте выглядит аналогично интерфейсу тонкого клиента.

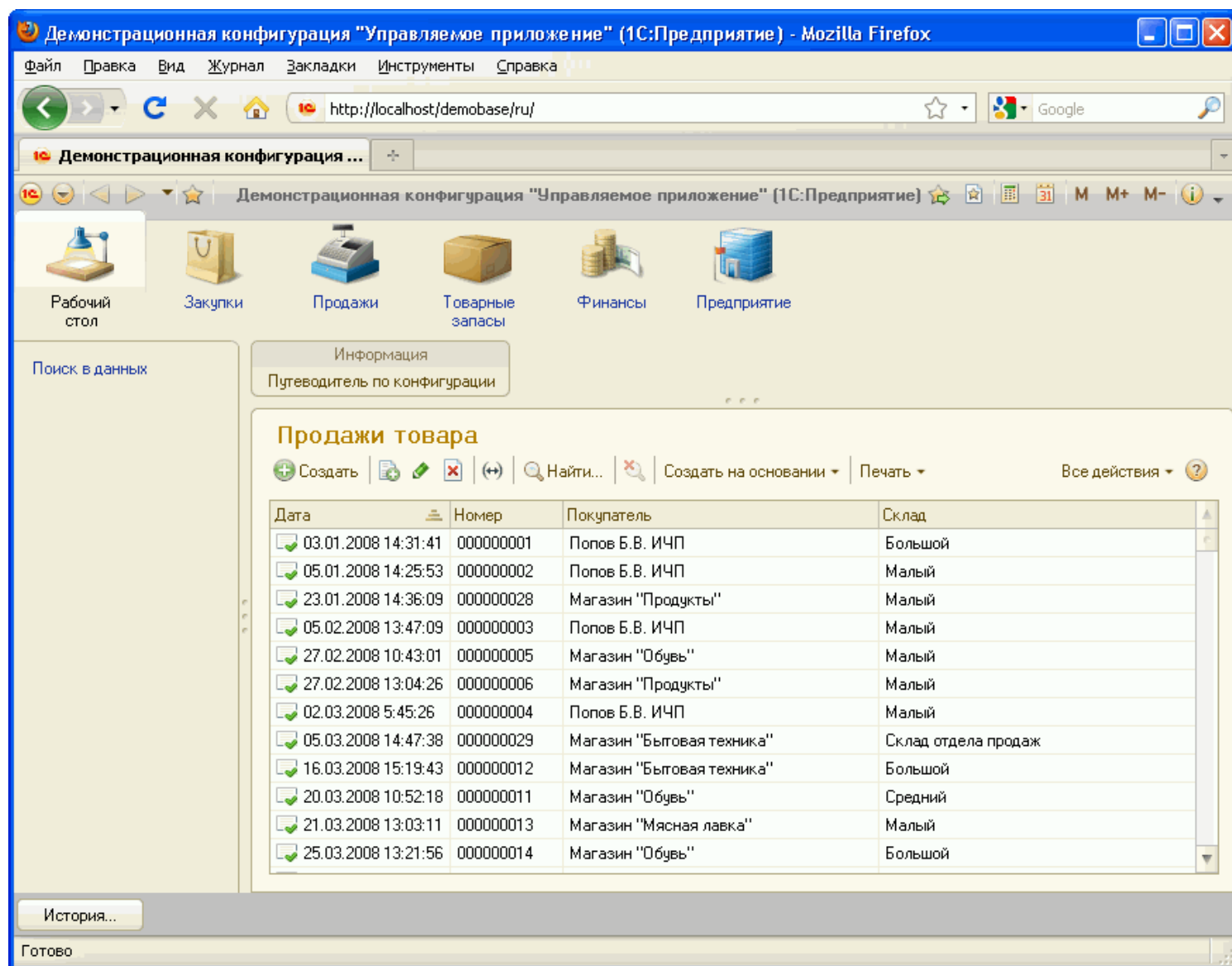


Рис. 222. Основное окно веб-клиента

При переходе по областям приложения (разделам, спискам и т. д.) в основном окне веб-клиента история переходов сохраняется.

Для выхода из приложения и перехода на пустую страницу в главном меню основного окна нужно выбрать пункт [Файл – Выход](#).

При каждом открытии отдельного окна в веб-клиенте выполняется проверка на срабатывание блокировщика всплывающих окон браузера. Если всплывающее окно было заблокировано:

- для всех веб-браузеров, кроме Mozilla Firefox, будет выведена инструкция по настройке браузера;
- для веб-браузера Mozilla Firefox будет выведен диалог настройки, с помощью которого можно установить все настройки, необходимые для работы веб-клиента.

14.1.2. Особенности печати

В форме печати веб-клиента рядом с кнопкой [Печать](#) отображается гиперссылка [Как настроить параметры печати](#), по нажатию которой отображается раздел справки. В этом разделе написано, как настроить печать в браузере, из которого вызывается печать.

14.1.3. Работа с файлами

Работа с файлами в веб-клиенте имеет особенности, которые связаны с особенностями моделей безопасности веб-браузера.

Диалоговые окна, отображаемые при загрузке/сохранении файлов, специфичны для конкретного вида браузера.

Все операции с файловой системой (получение файла с сервера, передача файла на сервер, удаление файла и т. п.) выполняются строго интерактивно.

В зависимости от конфигурации в интерфейсе системы может использоваться команда, включающая расширения для работы с файлами.

О настройке расширения для работы с файлами для браузеров Microsoft Internet Explorer и Mozilla Firefox см [здесь](#) и стр. 253. В браузерах Google Chrome и Safari расширение для работы с файлами не поддерживается.

ПРИМЕЧАНИЕ. Перетаскивание файлов в окно веб-клиента не поддерживается.

14.2. Особенности веб-клиента

Работа в веб-клиенте имеет некоторые особенности, перечисленные ниже.

14.2.1. Сохранение документов

В целях безопасности веб-браузера все операции с файловой системой требуют явного подтверждения пользователя.

Если установлено расширение для работы с файлами, то при сохранении файлов на диск веб-клиент выдаст предупреждение [Получить файл <Путь к файлу> с сервера?](#)

Если расширение для работы с файлами не установлено, то при сохранении файлов будет использоваться стандартный диалог браузера.

14.2.2. Работа с закладками

Если в браузере Internet Explorer 8 открыты закладки, то с помощью сочетания клавиш **Ctrl + Tab** возможно переключение между закладками браузера, а не между окнами приложения. Если закладки отключены, то переключение работает без ограничений для всех сочетаний клавиш (**Ctrl + Tab**, **Ctrl + F6**).

В браузере Mozilla Firefox по умолчанию нельзя активировать окно с помощью сочетаний клавиш. Чтобы использовать данную возможность, требуется дополнительная настройка. Подробнее о настройках браузера см. раздел [«Настройка Mozilla Firefox»](#) [здесь](#).

14.2.3. Работа с окнами

- Модальное окно в веб-браузере Microsoft Internet Explorer блокирует все родительские окна, в веб-браузере Mozilla Firefox блокируется только родительское окно данного окна. При этом остальные окна приложения затемняются, на эти окна можно переключиться, но в них нельзя выполнять никаких действий.
- При открытии модального окна перестает выполняться обновление отображения главного окна в веб-браузере Microsoft Internet Explorer. Поведение остальных окон аналогично поведению в браузере Mozilla Firefox.
- При попытке в веб-клиенте закрытия окна, которое заблокировано другим окном, выдается стандартное предупреждение браузера, включающее следующий текст: **"Это окно можно будет закрыть только после того, как будут закрыты открытые из него окна! Выберите Отмена/Cancel для отказа от закрытия этого окна!"**.
- Окно веб-клиента не может быть меньше некоторого размера, определяемого используемым веб-браузером.
- В некоторых случаях, когда в других клиентах для выбора значения открывается меню или всплывающий список, в веб-клиенте открывается отдельное окно.
- Меню расшифровки отчетов, построенных с помощью системы компоновки данных, отображается в виде отдельного окна веб-браузера.

При работе с несколькими мониторами поведение окон в веб-клиенте имеет следующие особенности:

- поведение основного окна зависит от поведения окна браузера при работе с несколькими мониторами;
- любое вспомогательное, блокирующее и модальное окно открывается в том же мониторе, в котором было последнее активное окно.

14.2.4. Другие особенности

14.2.4.1. Установка режима низкой скорости соединения

Чтобы установить режим низкой скорости соединения, следует выбрать пункт главного меню [Сервис – Параметры](#). В диалоге параметров установить флажок [Устанавливать режим низкой скорости соединения при запуске](#). Режим низкой скорости соединения будет установлен при следующем запуске веб-клиента.

ПРИМЕЧАНИЕ. При запуске информационной базы из адресной строки браузера для установки режима низкой скорости соединения также можно использовать параметр [O=Low](#).

14.2.4.2. Работа с файлами

- С помощью команды главного меню [Файл – Новый](#) можно создавать только текстовые документы.
- Не поддерживается режим сравнения файлов.
- Не поддерживается список последних открытых файлов.
- Не поддерживается перетаскивание файлов в формы веб-клиента.

14.2.4.3. Графическая схема

На веб-клиенте поддерживается открытие, сохранение и печать графической схемы. Редактирование графических схем не поддерживается.

14.2.4.4. Работа с табличным документом

Редактирование табличных документов в веб-клиенте по сравнению с тонким клиентом ограничено. При этом имеется возможность зафиксировать таблицу и отображать группы, используя соответствующие команды меню [Таблица](#).

Если у непустой ячейки табличного документа расположение текста [Авто](#), то ячейка объединяется либо до следующей заполненной ячейки, либо до конца табличного документа по горизонтали.

Также при работе с табличным документом в веб-клиенте имеются следующие особенности:

- поддерживается ограниченный список интерактивных команд, настраивающих отображение табличного документа;
- в ячейке табличного документа текст всегда будет выводиться горизонтально;
- фоновые картинки в табличном документе не отображаются;
- узоры для ячеек табличного документа не отображаются;
- в табличном документе и таблице формы не поддерживается прокрутка с помощью нажатия колеса мыши;
- в табличном документе, если после заполненной ячейки существуют пустые ячейки (до следующей заполненной ячейки или до конца строки табличного документа), ячейки отображаются объединенными.

Печать табличного документа

По умолчанию печать табличного документа в веб-клиенте осуществляется в формате PDF. Чтобы напечатать документ, используя формат HTML, следует в системных настройках (в главном меню пункт [Сервис – Параметры](#)) установить переключатель в группе [Способ печати](#) напротив пункта [HTML](#).

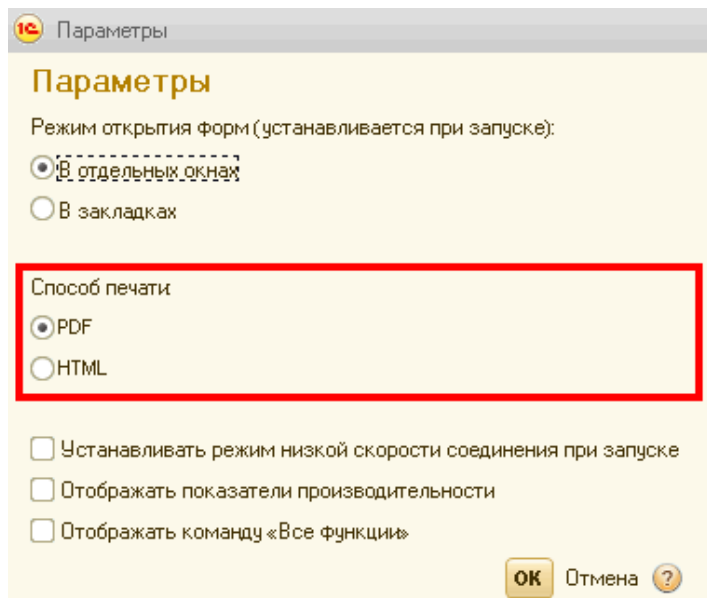


Рис. 223. Способ печати. Настройка по умолчанию

В браузере Google Chrome, если программа Acrobat Reader не установлена, и для документа используется режим "точная печать", то при выводе его на печать, будет отображено окно, содержащее пояснение.

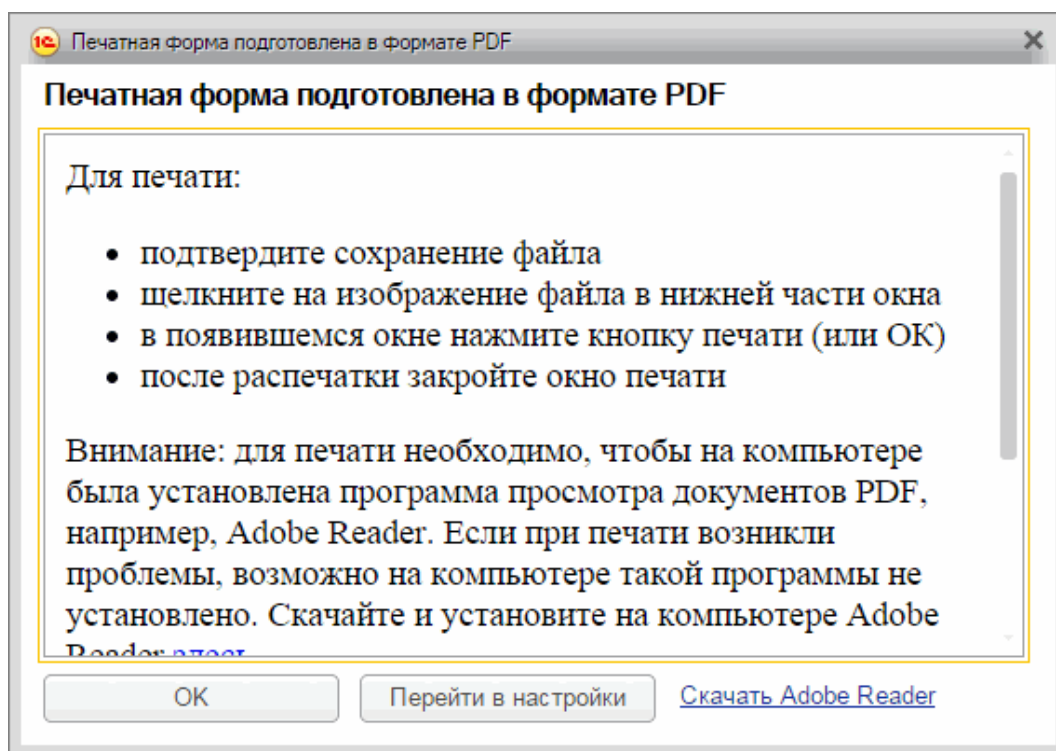


Рис. 224. Окно печати с пояснением

14.2.4.5. Поле картинки

В окне просмотра картинки в веб-клиенте команды изменения масштаба отсутствуют. С помощью команды [Растянуть пропорционально](#) подменю [Размер](#) можно растянуть картинку пропорционально по границам окна.

14.2.4.6. Окно состояния

В браузере Mozilla Firefox состояние обновляется при каждом серверном вызове и по окончании выполнения команды на встроеном языке.

14.2.4.7. Оповещения

На веб-клиенте оповещения отображаются внутри текущего окна.

Если оповещение отображается во вспомогательном окне и вспомогательное окно закрывается, то оповещение отображается в основном окне.

14.2.4.8. Активация окна

В браузере Internet Explorer 8 в некоторых ситуациях при переходе к уже открытому окну будет отображен диалог [Следует нажать ОК для перехода к окну](#). Переход будет выполнен после того, как будет нажата кнопка [ОК](#) в этом диалоге.

14.2.4.9. Прочие особенности

- В панели навигации все группы команд скрываются при нажатии клавиш [Ctrl + Alt + Shift + Num-](#) и раскрываются при нажатии [Ctrl + Alt + Shift + Num+](#).
- Не поддерживается прерывание работы программы с помощью клавиш [Ctrl + Break](#) в тех случаях, когда такое прерывание возможно в других клиентах.
- Не поддерживается настройка отображения диаграммы, дендрограммы, графической и географической схемы.
- Для диаграммы Ганта ограничены возможности настройки.
- Текст, не помещающийся в ячейку таблицы, не дополняется справа многоточием.
- В табличном документе, поле табличного документа и таблице формы не поддерживается прокрутка с помощью нажатия колеса мыши.
- В веб-клиенте поддерживается ограниченный набор шрифтов, доступных в используемых операционных системах. Другие шрифты не могут выбираться в диалоге выбора шрифта в веб-клиенте, а при использовании замещаются на поддерживаемые шрифты.
- Доступ к главному меню выполняется с помощью клавиши [F10](#), доступ с помощью одиночного нажатия клавиши [Alt](#) не поддерживается.
- Не поддерживается установка в веб-браузере масштаба отображения, отличного от 100 %, или использование нестандартного dpi.
- Во время выполнения длительных операций не происходит изменения вида курсора в тех случаях, когда такое изменение происходит в других клиентах.
- В веб-браузере Mozilla Firefox представление команды панели навигации и панели действий, не помещающееся в панели, не дополняется справа многоточием.
- В веб-браузере Microsoft Internet Explorer представление команды панели действий, не помещающееся в панели, не дополняется справа многоточием.
- Режим изменения размеров календаря, открытого из поля ввода, с помощью мыши не отключается, если кнопка мыши будет отпущена за границами окна, в котором открыт календарь. Для отключения режима изменения размера следует повторно нажать и отпустить левую кнопку мыши внутри окна с календарем.
- В веб-браузере Mozilla Firefox и в некоторых случаях при использовании веб-браузера Microsoft Internet Explorer цвет полос прокрутки поля HTML-документа будет соответствовать цвету, заданному в операционной системе.
- Отличается от других клиентов поведение команд отмены и возврата действия в многострочных полях ввода. В веб-клиенте объем отменяемых изменений определяется веб-браузером.
- При нажатии [Shift + Del](#) в многострочном поле ввода удаляется вся строка, без учета автоматического разбиения строки по границе поля ввода.
- Если в тексте подсказки присутствуют разделители строк, то в веб-браузере Mozilla Firefox вместо них отображаются пробелы.
- При работе с многострочными полями ввода и полем текстового документа в меню [Правка](#) доступны команды [Найти](#) – для поиска значений и [Заменить](#) – для замены значений в поле.
- В браузерах Google Chrome и Safari не поддерживается работа внешних компонентов, расширения для работы с файлами и расширения для работы со средствами криптозащиты.
- В браузерах Google Chrome и Safari не поддерживается команда [Вставка](#) через контекстное меню. Этот

пункт в контекстном меню не доступен.

- В браузерах **Google Chrome** и **Safari** в диалоге перехода по ссылке нет кнопки **Вставить из буфера обмена**. При открытии диалога перехода по ссылке значение буфера обмена в поле ввода ссылки не подставляется.
- В браузерах **Google Chrome** и **Safari** не поддерживается автоматическая аутентификация пользователя средствами операционной системы.
- В браузерах **Google Chrome** и **Safari** отсутствует возможность изменения колонтитулов при печати документов.
- В браузере **Safari** язык интерфейса веб-клиента будет соответствовать языку интерфейса веб-браузера (если он не переопределен при помощи параметра запуска).
- В браузере **Safari** размер окон календаря и калькулятора не может быть меньше минимально разрешенного размера окна для этого браузера.
- В браузере **Google Chrome** отсутствует диалог настройки печати и предварительного просмотра.
- В браузере **Google Chrome** не выше 7-й версии при перемещении между окнами с помощью горячих клавиш, при попытке открыть уже открытый документ и в других подобных случаях, после переключения на новое окно отображается сообщение **Выполнен переход к окну. Следует нажать ОК для продолжения**.
- Переключение между окнами веб-клиента по нажатию клавиш **Ctrl + Tab**, **Ctrl + Shift + Tab** не поддерживается для браузера **Chrome** под **Windows**.
- В браузерах **Google Chrome** и **Safari** команды **Вперед/Назад** поля HTML-документа не осуществляют навигацию по содержимому, заданному в виде строки (только по содержимому, заданному **URL**).
- Не поддерживается диалог настройки калькулятора.
- В окне **Все функции** доступен только поиск с помощью диалога поиска, который отображается при нажатии клавиш **Ctrl + F**.

14.2.5. Особенности работы на iPad

- Элемент управления **Форматированный Документ** функционирует в режиме «только для чтения».
- Не реализовано перетаскивание.
- Не выполняется переход к главному окну при открытых вспомогательных окнах.
- Не поддерживается печать.
- У полей ввода не отображается контекстное меню.
- Рекомендуется для завершения работы с веб-клиентом пользоваться кнопкой закрытия в верхнем правом углу главного окна.
- Поддерживается только режим работы с окнами в закладках;
- Поддерживаются стандартные жесты для увеличения и уменьшения масштаба.

14.3. Настройка Microsoft Internet Explorer

Для использования интернет-браузера необходимо выполнить настройку его конфигурации. Для этого в меню **Сервис (Tools)** нужно выбрать пункт **Свойства обозревателя (Internet Options)** и изменить следующие настройки:

- Разрешить выполнение JavaScript скриптов:
 - на вкладке **Безопасность (Security)** для зоны, в которой находится сайт, нажать кнопку **Другой (Custom level)**;
 - в разделе **Сценарии (Scripting)** для параметра **Активные сценарии (Active scripting)** установить **Разрешить (Enable)**.

- Разрешить всплывающие окна:

- на вкладке [Конфиденциальность \(Privacy\)](#) в разделе [Блокирование всплывающих окон \(Pop-up Blocker\)](#) снять флажок [Блокировать всплывающие окна \(Block Pop-ups\)](#).

- Разрешить использование cookies:

- на вкладке [Конфиденциальность \(Privacy\)](#) с помощью регулятора выбрать уровень использования cookies – не выше [Умеренно высокий \(Medium High\)](#).

Для работы с внешними компонентами необходимо выполнить следующую настройку интернет-браузера:

- на вкладке [Безопасность \(Security\)](#) нажать кнопку [Другой \(Custom level\)](#);
- в открывшемся окне установить значение [Разрешить \(Enable\)](#) для следующих параметров:
 - [Запрос разрешения для ActiveX объектов \(Automatic prompting for ActiveX controls\)](#);
 - [Загрузка подписанных ActiveX объектов с разрешения пользователя \(Download signed ActiveX controls\)](#);
 - [Работа с ActiveX объектами и плагинами \(Run ActiveX controls and plug-ins\)](#);
 - [Работа с объектами, помеченными как безопасные \(Script ActiveX controls marked safe for scripting\)](#).

При каждом открытии вспомогательных окон раскладка клавиатуры выбирается из настроек операционной системы. Если она не совпадает с языком ввода данных, то, чтобы исключить переключение языка, рекомендуется в настройках указать нужный язык ввода.

Для настройки языка ввода по умолчанию нужно выполнить следующие действия:

1. Открыть настройки операционной системы [Язык и региональные стандарты \(Regional and Languages Options\)](#).
2. На закладке [Языки \(Languages\)](#) в разделе [Языки и службы текстового ввода \(Text services and input languages\)](#) нажать кнопку [Подробнее... \(Details...\)](#).
3. В открывшемся окне на закладке [Параметры \(Settings\)](#) в разделе [Язык ввода по умолчанию \(Default input language\)](#) выбрать нужный язык.
4. Нажать кнопку [ОК](#).

ПРИМЕЧАНИЕ. Подробная инструкция по настройке браузера также доступна при выборе команды главного меню [Справка – Особенности настройки веб-браузера](#).

14.3.1. Настройка расширения для работы с файлами

Реализация для Internet Explorer требует установки специального ActiveX компонента [1С:Предприятие 8.3. Расширение работы с файлами для Internet Explorer \(x86\)](#) либо его 64-разрядного варианта для 64-разрядного Internet Explorer. Компонент включен в платформу, но его установка должна быть предусмотрена в прикладном решении. Если у пользователя недостаточно прав для установки компонента, то ее может выполнить администратор.

Для работы с компонентом в браузере должны быть предоставлены права на установку и использование подписанных [ActiveX](#) компонентов для зоны, в которой находится веб-сайт.

ПРИМЕЧАНИЕ. Для корректной работы расширения для работы с файлами рекомендуется использовать библиотеку Microsoft Core XML Services (MSXML) версии 4.0 или 6.0.

Список версий модулей MSXML, ограничения и рекомендации по установке расположены по адресу: <http://support.microsoft.com/kb/269238>.

14.4. Настройка Mozilla Firefox

Для использования браузера Mozilla Firefox требуется следующая настройка:

1. В меню **Инструменты (Tools)** выбрать пункт **Настройки (Options)**.

- на панели **Содержимое (Content)** установить следующие настройки:
 - снять флажок **Блокировать всплывающие окна (Block pop-up windows)**;
 - установить флажок **Использовать JavaScript (Enable JavaScript)**.
- На панели **Приватность (Privacy)** установить следующие настройки:
 - установить флажок **Принимать cookies с сайтов (Accept cookies from sites)**;
 - для детальной настройки приема или отклонения cookies нажать кнопку **Исключения (Exceptions)** и следовать пояснениям диалога настройки.
- Нажать **ОК**.

Затем нужно запустить веб-клиент «1С:Предприятия», введя в адресную строку браузера адрес информационной базы. В главном меню веб-клиента выбрать пункт **Сервис – Настройка веб-браузера**. На экране отобразится диалог включения расширения браузера.

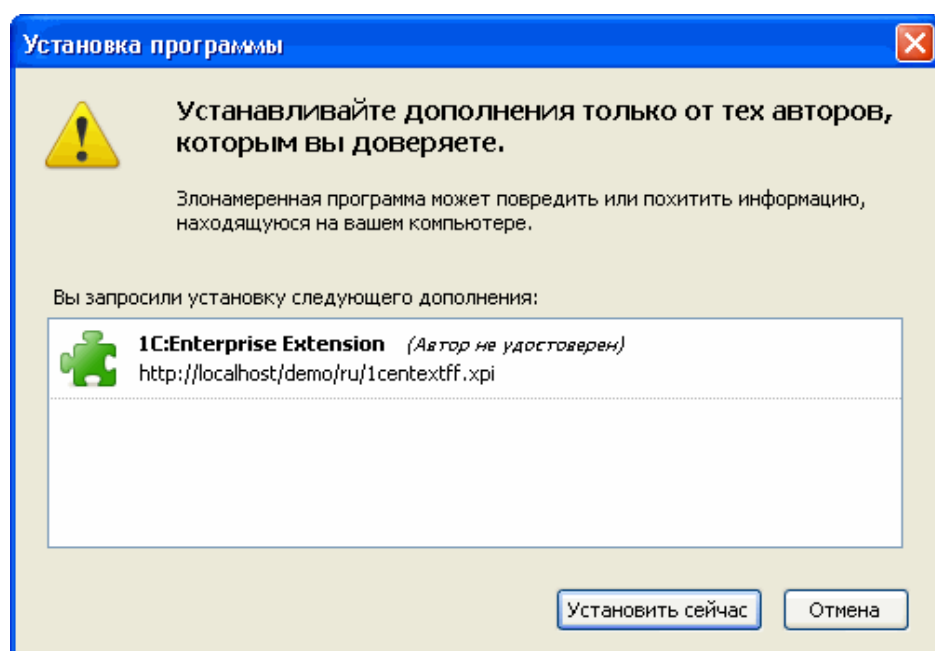


Рис. 225. Диалог установки расширения браузера Mozilla Firefox

Для работы с буфером обмена операционной системы с расширением для работы с файлами, внешними компонентами и криптографией следует нажать кнопку **Включить расширение (рекомендуется)**. При этом будут установлены соответствующие настройки.

При выборе ссылки **Работать без расширения** будет отображена информация, необходимая для ручной настройки браузера.

При нажатии кнопки **Завершить работу** текущий сеанс «1С:Предприятия» будет завершен.

ПРИМЕЧАНИЕ. Дополнительная информация о настройке аутентификации средствами ОС в браузере Mozilla Firefox доступна по адресу (на английском языке): https://developer.mozilla.org/en/Integrated_Authentication.

14.4.1. Языковые настройки Mozilla Firefox

Если язык интерфейса платформы не совпадает с языком интерфейса браузера, то для правильной работы веб-клиента требуется выполнить следующую настройку (расположение настроек приведено для версии 3.5):

1. В главном меню выбрать **Tools (Инструменты) – Options (Настройки)**.
2. В появившемся окне на закладке **Contents (Содержимое)** в группе **Fonts & Colors (Шрифты и цвета)** нажать кнопку **Advanced (Дополнительно)**.

3. В параметре **Default Character Encoding** (Кодировка по умолчанию) выбрать **Unicode** (Юникод (UTF-8)).

ПРИМЕЧАНИЕ. В ситуации, когда веб-браузер Mozilla Firefox запомнил решение пользователя по предоставлению привилегий, и требуется изменить это решение, необходимо отредактировать файл **prefs.js**, который хранится в папке его профиля веб-браузера Mozilla Firefox.

Описание, как найти папку профиля, расположено по адресу: <http://support.mozilla.org/ru/kb/profil-gde-firefox-branit-vashi-zakladki-paroli-i>.

Далее нужно:

1. Завершить работу Mozilla Firefox.
2. Открыть **prefs.js** любым текстовым редактором.
3. Найти в тексте строку **UniversalXPConnect**. Будет найден фрагмент текста следующего вида:

```
user_pref("capability.principal.codebase.p<XX>.granted", "UniversalXPConnect");
```

```
user_pref("capability.principal.codebase.p<XX>.id", "<URL>");
```

```
user_pref("capability.principal.codebase.p.<XX>.subjectName", "").
```

Либо:

```
user_pref("capability.principal.codebase.p<XX>.denied", "UniversalXPConnect");
```

```
user_pref("capability.principal.codebase.p<XX>.id", "<URL>");
```

```
user_pref("capability.principal.codebase.p<XX>.subjectName", "").
```

Где **<XX>** – любое число, а **<URL>** – это адрес нужного веб-сайта. Если **<URL>** другой, то нужно продолжить поиск «UniversalXPConnect» по тексту.

4. Удалить найденные строки из текста.

14.5. Настройка Google Chrome

Для использования веб-браузера необходимо выполнить настройку свойств обозревателя.

Для этого нужно открыть меню **Инструменты (Tools)**, выбрать пункт **Параметры (Options)**, на закладке **Расширенные (Under the Hood)** нажать кнопку **Настройки содержания (Content settings)** и выполнить настройки:

- разрешить использование cookies: на закладке **Файлы cookie (Cookies)** установить флажок **Разрешить сохранять локальные данные (рекомендуется) (Allow local data to be set (recommended))** или настроить исключения;
- разрешить выполнение JavaScript скриптов: на закладке **JavaScript** установить флажок **Разрешить всем сайтам использовать JavaScript (рекомендуется) (Allow all sites to run JavaScript (recommended))** или настроить исключения;
- разрешить всплывающие окна: на закладке **Всплывающие окна (Pop-ups)** установить флажок **Разрешить всплывающие окна для всех сайтов (Allow all sites to show pop-ups)** или настроить исключения.

Рекомендуется выполнить настройку для сохранения файлов. Для этого:

- открыть меню **Инструменты (Tools)**, выбрать пункт **Параметры (Options)**;
- на закладке **Расширенные (Under the Hood)** в разделе **Загрузки (Downloads)** установить флажок **Запрашивать место для сохранения каждого файла перед загрузкой (Ask where to save each file before download)**.

ПРИМЕЧАНИЕ. Подробная инструкция по настройке браузера также доступна при выборе команды главного меню [Справка – Особенности настройки веб-браузера](#).

14.6. Настройка Safari

Для использования веб-браузера необходимо выполнить настройку свойств обозревателя. Для этого нужно открыть меню Настройки Safari (Safari Settings), выбрать пункт [Настройки \(Preferences\)](#) и выполнить настройки:

- разрешить выполнение JavaScript скриптов: на закладке [Безопасность \(Security\)](#) установить флажок [Подключить JavaScript \(Enable JavaScript\)](#);
- разрешить использование cookies: на закладке [Безопасность \(Security\)](#) в группе [Принимать cookies \(Accept cookies\)](#) установить флажок [Только от посещаемых сайтов \(Only from sites I visit\)](#);
- разрешить всплывающие окна: на закладке [Безопасность \(Security\)](#) снять флажок [Блокировать всплывающие окна \(Block pop-up windows\)](#) .

Рекомендуется выполнить настройку для сохранения файлов. Для этого:

- в меню Настройки Safari (Safari Settings) выбрать пункт [Настройки \(Preferences\)](#);
- на закладке [Основные \(General\)](#) установить флажок [Всегда выполнять запрос перед загрузкой \(Always prompt before downloading\)](#).

ПРИМЕЧАНИЕ. Настройка вопроса о местоположении сохраняемого файла недоступна в веб-браузере Safari для Mac OS.

Глава 15. Режим низкой скорости соединения

Режим низкой скорости соединения предназначен для использования системы по медленным каналам связи (например, через GPRS-модем). Скорость работы и размер трафика оптимизируются за счет уменьшения количества обращений к серверу и отказа от передачи несущественной информации.

Скорость соединения устанавливается в диалогах добавления и редактирования информационной базы (подробнее см. книгу «1С:Предприятие 8.3. Руководство администратора»), в параметрах информационной базы при запуске клиента (подробнее см. раздел [«Установка скорости соединения»](#) [здесь](#)).

Внешне установка режима низкой скорости соединения проявляется в виде отсутствия картинок разделов на панели разделов.

При начале работы показываются команды рабочего стола и пустая рабочая область, на которой содержится гиперссылка [Показать рабочий стол](#). Если необходима работа с формами рабочего стола, нужно открыть его по этой ссылке.

При работе с данными в режиме низкой скорости следует учитывать, что оповещение сервера о закрытии формы, которая заблокировала данные, посылается через 20 секунд или одновременно с запросом об открытии другой формы.

В этом режиме при выборе из списков в поле ввода и при вводе по строке данные будут считываться из базы данных не каждый раз, а по возможности будут использоваться ранее полученные данные. Если в базе появились новые данные и их нужно выбрать, а они еще не отображаются в списке, то нужно повторить выбор в поле ввода (повторно нажать кнопку выбора без перехода в другое поле) или обновить список (командой [Обновить](#) или клавишей **F5** в форме списка выбора). Также в поле ввода при вводе с клавиатуры всплывающий список автоматически не открывается. Чтобы открыть его, следует нажать клавишу [Стрелка вниз](#).

Глава 16. Режим обычного приложения

Данная глава содержит информацию по работе с конфигурациями, разработанными специально для режима обычного приложения или созданными на платформе версии 8.1.

16.1. Интерфейс

Основное окно программы в режиме «обычного приложения» значительно отличается от режима управляемого приложения и в общем случае может иметь вид, как показано на рисунке ниже.

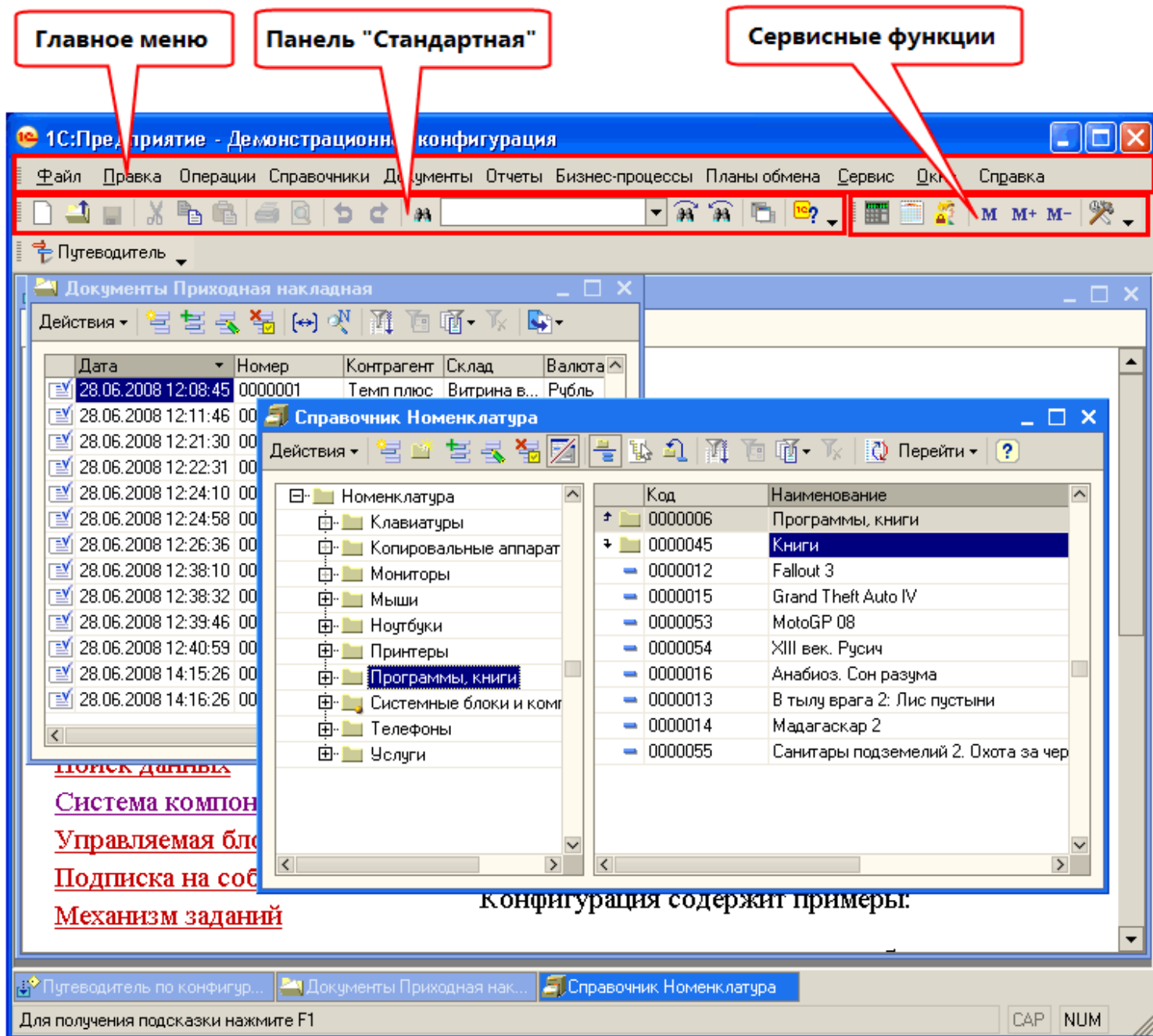


Рис. 226. Основное окно «обычного приложения»

Главное меню содержит команды работы с файлами, редактирования, вызова форм объектов конфигурации и т. д.

На панелях инструментов располагаются команды вызова стандартных и сервисных функций. Для перехода к спискам и объектам информационной базы используется меню **Операции** и кнопки панелей инструментов, если определено на этапе конфигурирования.

16.2. Меню «Операции»

Меню **Операции** используется для перехода к спискам и объектам информационной базы, а также для

вызова сервисных функций. При выборе типа объекта в меню [Операции](#) на экране отобразится список существующих объектов данного типа. В этом списке следует выбрать наименование нужного объекта и затем нажать кнопку [OK](#). Форма выбранного объекта откроется в новом окне.

16.2.1. Настройка списка

16.2.1.1. Отбор и сортировка списка

Настройка просмотра списка, отображаемого в таблице (списки объектов, журналы и списки документов и т. д.), производится в диалоге [Отбор и сортировка](#), который вызывается выбором пункта [Установить отбор и сортировку списка](#).

Настройка отбора

В некоторых списках предусматривается возможность отбора данных по значениям реквизитов. Это позволяет просматривать только те данные, которые имеют определенное значение в реквизите, выбранном для отбора. Например, можно вывести всех контрагентов, определенных как частное лицо.

Отбор данных производится по определенному набору реквизитов.

Отбор данных характеризуется **видом отбора**, **видом сравнения** и **значением отбора**.

Видами отбора называются критерии, которые определяются на этапе конфигурирования. Пользователь, выполняя отбор, может применять только те виды отборов, которые определены на этапе конфигурирования. Видом отбора может быть, например, реквизит [Вид контрагента](#).

Вид сравнения указывает, как будет применен отбор. Состав возможных видов сравнения определяется типом данных.

Можно отбирать данные, содержащие даты больше заданной, находящиеся в интервале дат, входящие в список или, наоборот, не входящие в список и т. д.

Значением отбора может являться любое из имеющихся значений этого реквизита, например, [Частное лицо](#). Вид и значение отбора составляют условие отбора. Условие отбора в определенный момент времени может быть только одно.

Заметим, что возможность отбора предусматривается для большинства списков, причем по одному или нескольким реквизитам. Если для списка отборы не предусмотрены, то в меню и в панели инструментов не будут содержаться элементы интерфейса для управления отборами.

При установке отбора в окне формы списка выводятся только элементы, удовлетворяющие выбранному условию отбора.

Отбор данных в списке может быть установлен несколькими способами.

Ручная установка отбора. Пользователь может сам задавать условие отбора, т. е. выбирать и вид отбора, и значение отбора.

Для ручной установки отбора следует выбрать пункт [Установить отбор и сортировку списка](#).

Данный пункт меню и кнопка могут быть недоступны, если пользователю не предоставлена возможность устанавливать отбор самостоятельно.

При входе в режим установки отбора открывается диалог, в котором необходимо выбрать вид и конкретное значение отбора. Состав элементов отбора определяется при конфигурировании задачи для каждого вида списка. Состав элементов отбора может быть установлен или изменен программно перед открытием формы.

Например, для журнала документов настройка списка может выглядеть следующим образом:

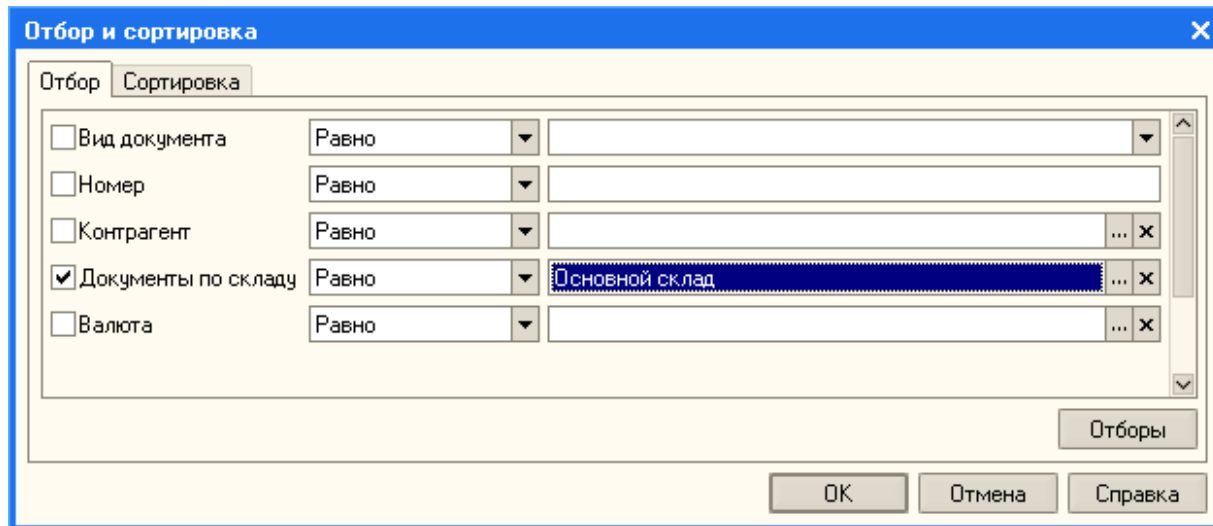


Рис. 227. Настройка списка журнала документов

Для того чтобы сформировать условие отбора, следует выбрать реквизиты, указать их значение и вид сравнения. Например, реквизит – [Документы по складу](#), значение – [Основной склад](#), а вид сравнения – [Не равно](#).

Справа от списка видов отбора в диалоге расположено поле для непосредственного ввода значения отбора. Формат ввода зависит от типа значения. Например, для отбора по числовому реквизиту нужно будет ввести числовое значение, а для отбора по организации – выбрать организацию из списка организаций. После ввода значения для установки отбора нажать кнопку [OK](#).

Для некоторых видов отбора непосредственный ввод значения невозможен из-за того, что в них включаются значения различных типов.

Если выполняется установка отбора для иерархических данных, то состав списка зависит от режима просмотра. При установленном иерархическом просмотре условия отбора не применяются к группам, т. е. показываются все группы справочника. Если осуществляется неиерархический просмотр, то показываются только элементы, удовлетворяющие условиям отбора.

Для иерархических данных возможна установка вида сравнения по иерархии. Если выбран вид сравнения [В иерархии](#), то будет осуществляться проверка того, что сравниваемое значение входит в группу. Если выбран вид сравнения [Не в иерархии](#), то будет осуществляться проверка того, что сравниваемое значение не входит в группу.

При изменении отбора в списке для всех регистров, подчиненных регистратору, порядок может заменяться оптимальным для установленного отбора в том случае, когда отбор был установлен системой при открытии списка. Например, при установке отбора по регистратору устанавливается порядок по номеру строки, а при отмене отбора по регистратору – по дате.

Отбор по текущему значению

При анализе данных, выведенных в табличное поле, часто требуется выполнить отбор по конкретному текущему значению ячейки. Такой отбор можно выполнить с использованием диалога [Отбор и сортировка](#). Но для упрощения процедуры отбора можно воспользоваться специальной командой [Отбор по текущему значению](#) или соответствующей кнопкой командной панели.

Команда доступна, если для данной колонки таблицы можно произвести отбор. Визуально такие колонки можно определить по доступности кнопки [Отбор по текущему значению](#) командной панели.

Чтобы выполнить отбор по значению, нужно выбрать ячейку, содержащую это значение, и команду [Отбор по текущему значению](#). При выполнении команды производится отбор, и кнопка будет отображаться в нажатом виде.

Выполнение отбора по значению эквивалентно выполнению отбора по данной колонке с указанием единственного значения и условия отбора [Равно](#).

Для отмены отбора по значению нужно выбрать ячейку, содержащую это значение, и повторно выбрать команду [Отбор по текущему значению](#).

Допускается последовательное выполнение отборов по значению (после отбора по значению в одной колонке выполняется отбор по значению другой колонки и т. д.).

Если по колонке выполнен отбор, отличный от отбора по значению, то кнопка [Отбор по текущему значению](#) не будет показана как нажатая, а выполнение отбора по значению приведет к изменению условий отбора, выполненного прежде.

Отключение отбора

Чтобы отключить отбор, нужно выбрать пункт [Отключить отбор](#). В этом случае будут показываться все записи, имеющиеся в списке.

Настройка порядка представления

На закладке [Сортировка](#) указывается порядок, в котором будут выводиться отобранные объекты.

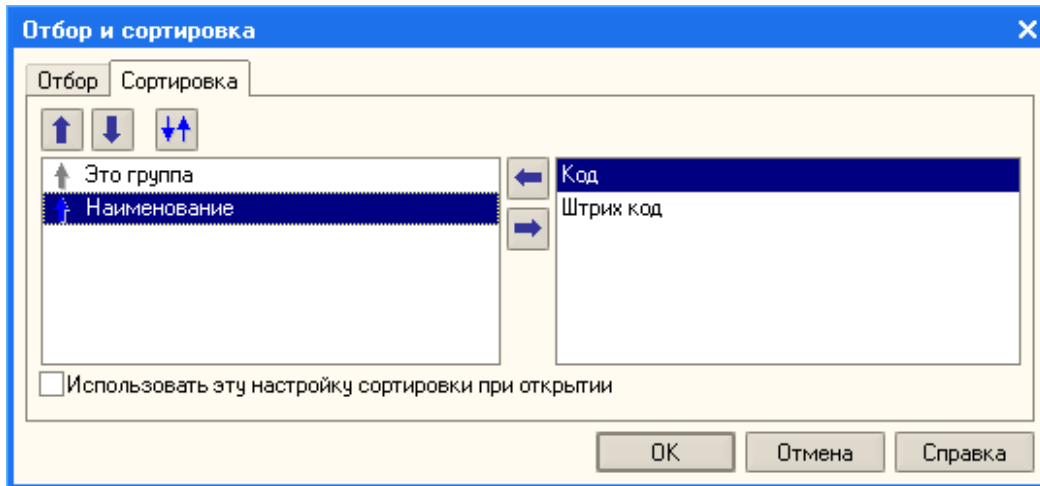


Рис. 228. Настройка сортировки списка

На закладке расположены два списка. В левом формируется набор реквизитов для задания порядка вывода данных, очередность действия реквизитов (приоритет) и направление сортировки по каждому реквизиту. Реквизиты выбираются из списка, расположенного в правой части. С помощью кнопок формируется список, очередность действия реквизитов и направление сортировки.

Если указано несколько реквизитов, то данные сортируются сначала по первому реквизиту с учетом направления сортировки, затем данные с одинаковым значением этого реквизита сортируются по второму реквизиту также с учетом направления сортировки и т. д.

Чтобы выбранная сортировка устанавливалась при открытии формы, следует установить флажок [Использовать эту настройку сортировки при открытии](#).

16.2.1.2. Настройка списка

Настройка порядка следования колонок в списке, их видимости, а также поведения списка при открытии и управления обновлением данными производится в диалоге [Настройка списка](#). Диалог открывается командой [Действия – Настройка списка](#).

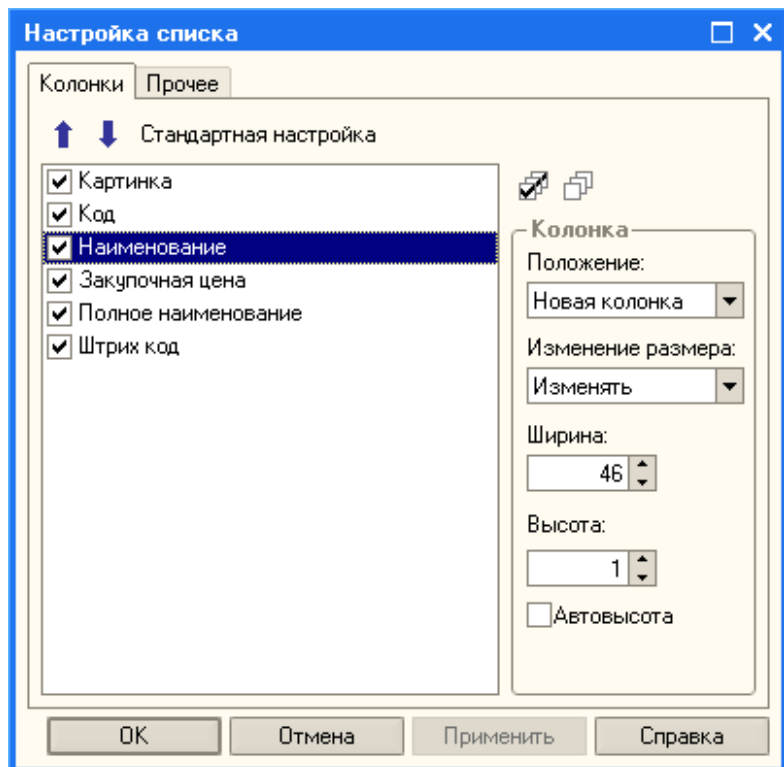


Рис. 229. Диалог настройки отображения колонок списка

На закладке [Колонки](#) производится настройка показа колонок. С помощью кнопок [Переместить вверх](#) и [Переместить вниз](#) устанавливается порядок следования колонок в списке.

Кнопка [Применить](#) позволяет сразу просмотреть результат выбранных настроек.

Для восстановления стандартного порядка следует нажать кнопку [Стандартная настройка](#).

ВАЖНО! Если при конфигурировании или средствами языка для колонки запрещена ее настройка или изменение позиции (перетаскиванием или в диалоге), то настройка таких колонок невозможна. Также не допускается вставка колонок между двумя колонками, которым запрещено изменение положения.

На закладке [Прочее](#) производится настройка восстановления позиции списка при его открытии, выбор позиции в списке при открытии, а также настройка автоматического обновления списка.

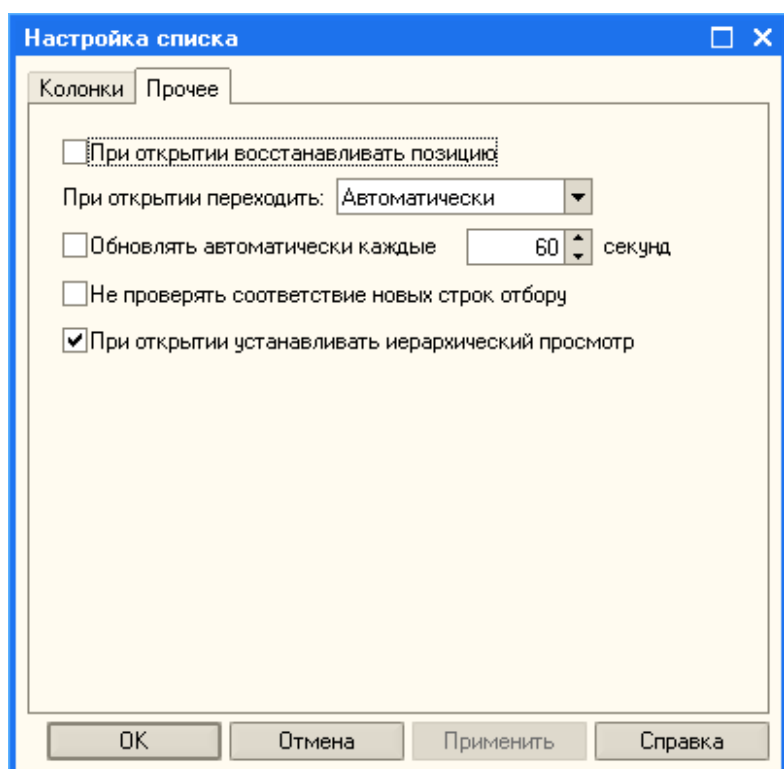


Рис. 230. Настройка дополнительных параметров списка

Если флажок [При открытии восстанавливать позицию](#) установлен, то активной будет строка, которая была активна при закрытии списка.

Выбор в реквизите [При открытии переходить значения](#) [В начало списка](#) или [В конец списка](#) означает, что при открытии списка активной становится первая или последняя строка списка, в зависимости от указанного в настройке условия перехода.

Данные, показываемые в списке, могут меняться не только при их интерактивном вводе пользователем, но и при создании других объектов, а также при работе в многопользовательском режиме. Часто необходимо, чтобы информация, размещенная в списке, была актуальной без дополнительных действий пользователя.

Если флажок [Обновлять автоматически каждые](#) установлен, то система будет обновлять список через интервал, указанный в поле ввода, расположенном справа от флажка. Основное назначение данного режима настройки заключается в обеспечении автоматической актуализации данных в списке с заданной периодичностью. Устанавливать режим рекомендуется только для отдельных форм списков, с которыми одновременно работают несколько пользователей, результаты работы которых должны быть видны сразу. На практике такие ситуации встречаются достаточно редко. Поэтому настройка автоматического обновления, как правило, не нужна.

Если флажок [Не проверять соответствие новых строк отбору](#) снят, то при окончании ввода новой строки, не соответствующей установленному отбору, об этом будет выдано сообщение.

Флажок [При открытии устанавливать иерархический просмотр](#) присутствует только при настройках иерархических списков. Он предназначен для управления иерархическим просмотром в форме. Его состояние соответствует состоянию кнопки командной панели [Иерархический просмотр](#) списка.

16.2.2. Отчеты

16.2.2.1. Конструктор настроек отчета

Конструктор настроек отчета имеет несколько относительно простых вариантов настройки отчета, которые предлагается выбрать. Он удобен для тех, кому нужно настроить отчет, не вникая во все «тонкости», так как позволяет выбрать только основные настройки для каждого вида элементов отчета.

Вызов этого конструктора может привести к потере настроек, установленных вручную, поэтому на экране появится вопрос: [Настройки, введенные вручную, будут утеряны. Продолжить?](#) При ответе [Да](#) запустится конструктор настроек компоновки данных.

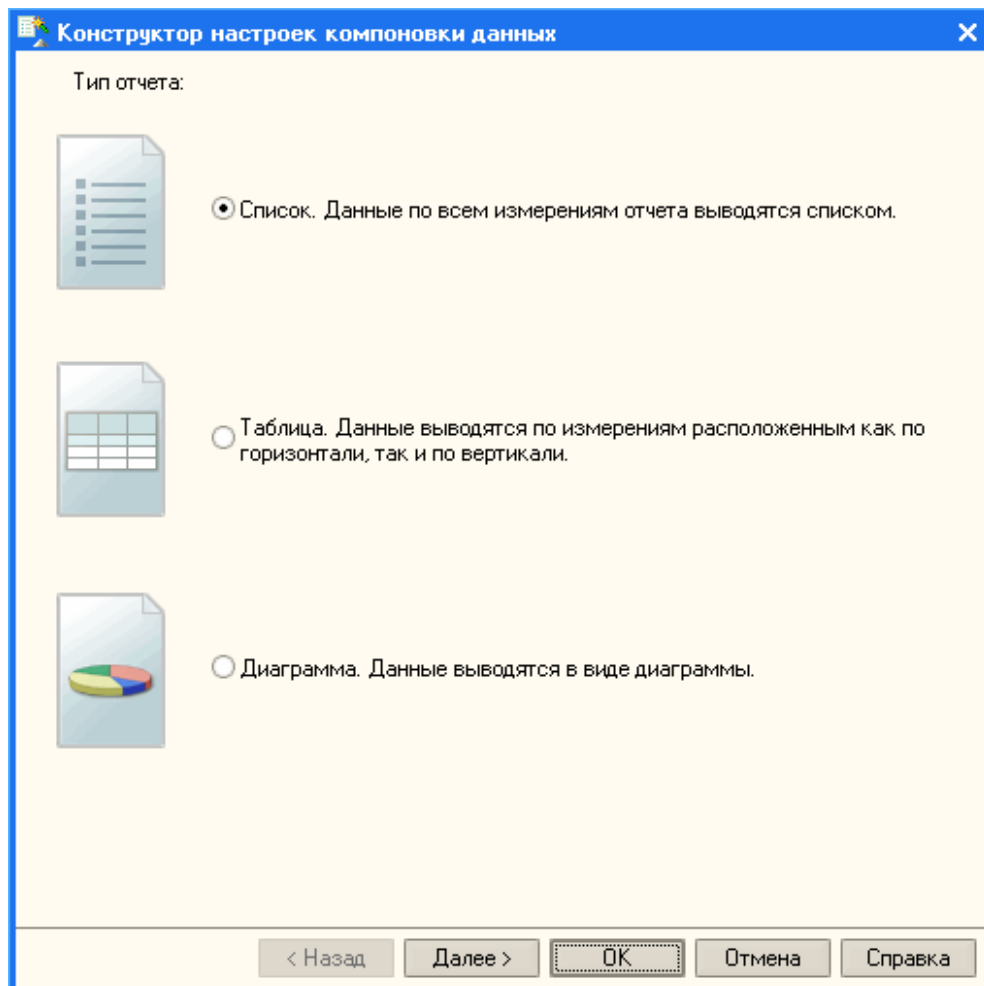


Рис. 231. Конструктор настроек компоновки данных

Нужно выбрать необходимый вид расположения данных и нажать кнопку **Далее**. Если не нужно настраивать остальные параметры – кнопку **OK**.

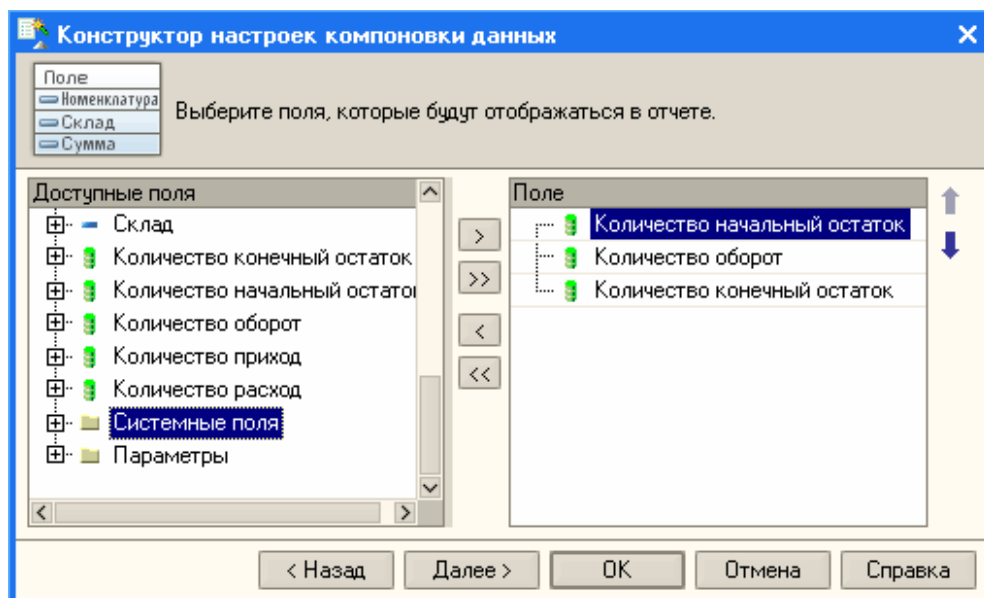


Рис. 232. Выбор полей в отчете

В открывшемся окне выбора полей нужно выбрать поля, которые должны отображаться в отчете. Для этого можно использовать как кнопки, так и перетаскивание. Выбранные поля можно сортировать при помощи стрелок.

Для перехода на следующее окно настроек следует нажать кнопку **Далее**.

Для типа отчета **Таблица** можно указать группировки в строках, колонках и таблицах. Для типа отчета **Диаграмма** – в сериях, точках и таблицах.

В открытом окне следует выбрать поля и тип группировки, в качестве которого могут выступать:

- **Без иерархии** – только данные элементов;
- **Иерархия** – данные как по группам, так и по элементам;
- **Только иерархия** – вывод данных по группам.

После нажатия кнопки **Далее >** конструктор настроек перейдет в окно выбора поля упорядочивания.

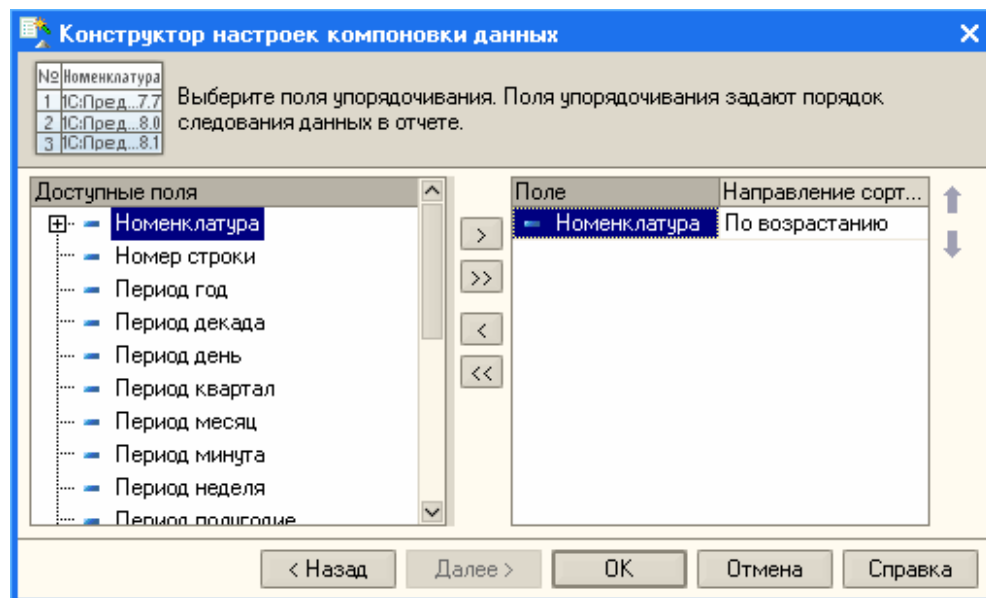


Рис. 233. Выбор полей упорядочивания

Поля упорядочивания выбираются аналогично полям группировок. Для каждого поля устанавливается направление сортировки.

Для типов отчета **Список** и **Таблица** на этом конфигурирование параметров в конструкторе настроек закончено. Для типа отчета **Диаграмма** после нажатия кнопки **Далее >** выбирается тип диаграммы.

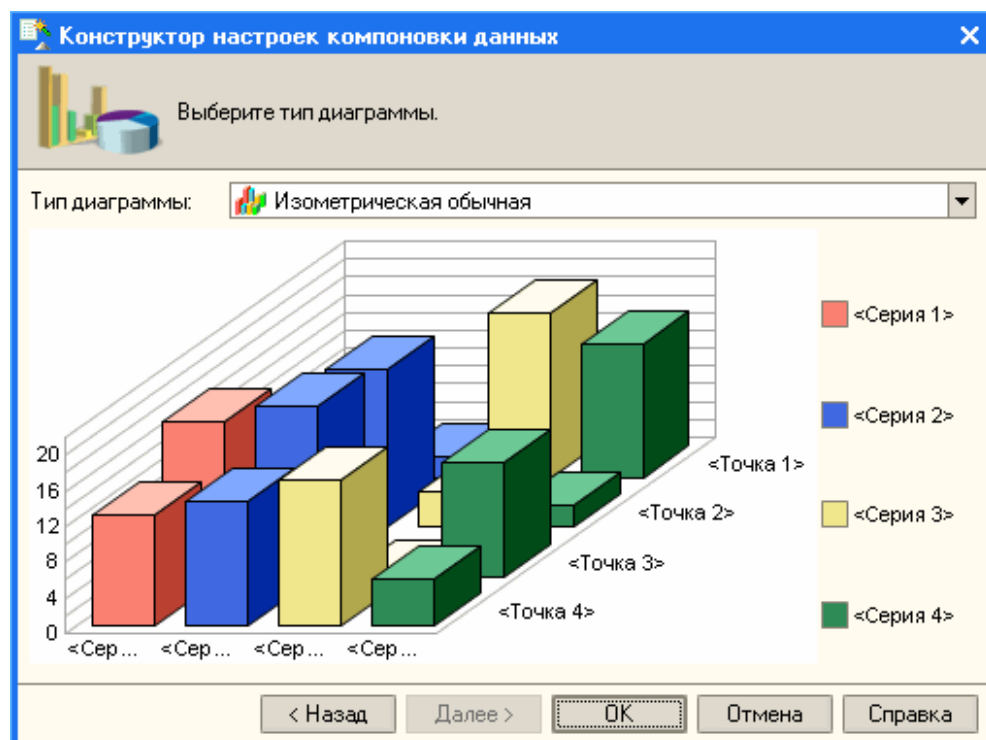


Рис. 234. Выбор диаграммы в конструкторе настроек отчета

По окончании работы с конструктором настроек компоновки данных сохраненные настройки станут доступными для последующих изменений в окне настроек отчета.

16.2.3. Настройка отчета

Отчет содержит настройки, которые могут быть изменены пользователем. В понятие «настройки» в данном случае входят:

- структура отчета,
- состав полей, выводимых в отчет,
- состав полей, входящих в группировки,
- отборы,
- сортировки,
- условные оформления,
- другие настройки.

Настройки открываются для изменения в отдельном окне по кнопке [Настройки](#) отчета (если она предусмотрена на этапе конфигурирования).

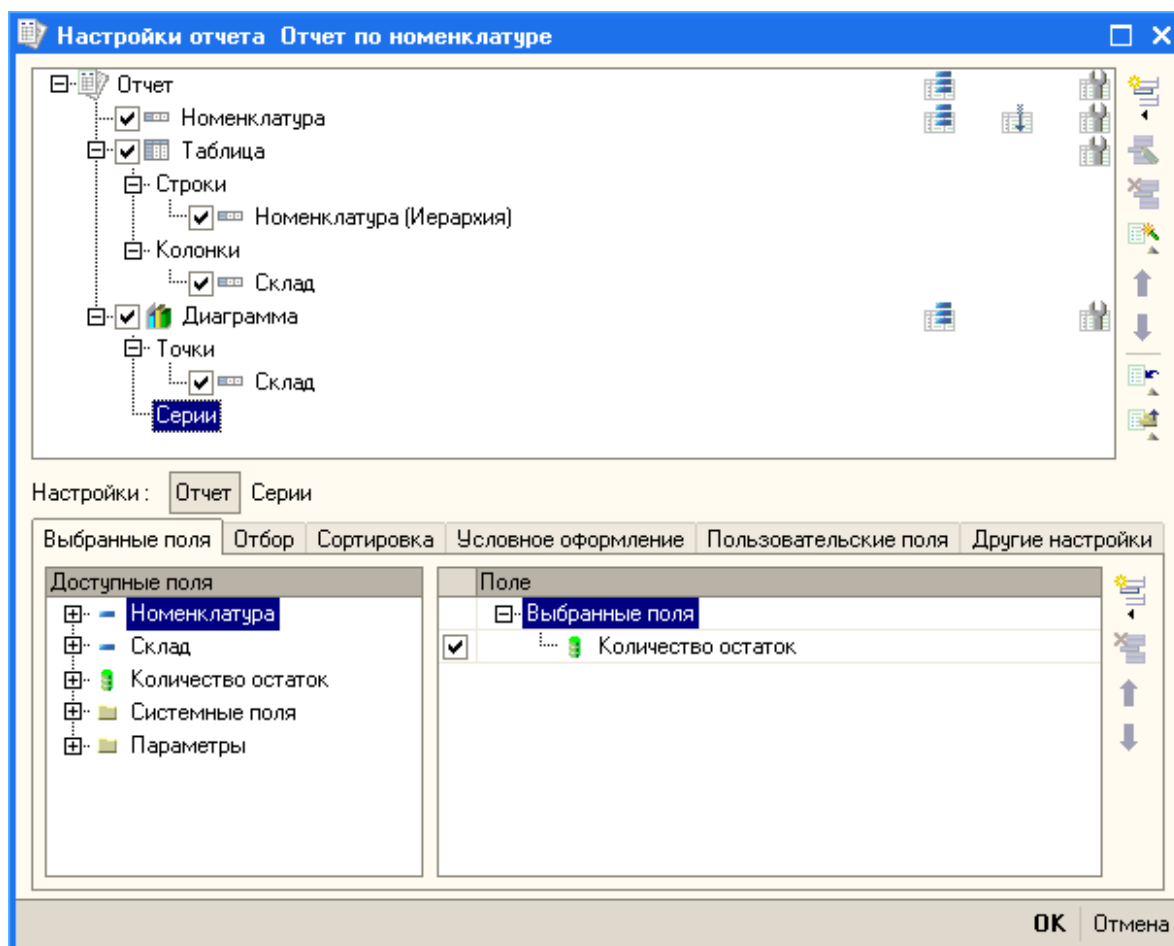


Рис. 235. Форма настроек отчета «Отчет по номенклатуре»

Форма редактирования настроек состоит из табличного поля структуры отчета, командной панели выбора настраиваемого элемента и набора настроек, распределенных по закладкам. Настроить можно как отчет целиком, так и каждый его элемент по отдельности, в т. ч. вложенный. Переключение между режимами настройки осуществляется по кнопкам [Отчет](#) и [<Элемент структуры>](#) (заголовок второй кнопки меняется в зависимости от выбранного элемента структуры). С помощью кнопок на командной панели формы настройки отчета можно сохранить в файл, или загрузить из файла XML-документа, или же вызвать конструктор настроек отчета.

Работа с настройками отчета в целом не отличается от работы с настройками варианта отчета.

16.3. Сервисные возможности

16.3.1. Установка параметров

Для установки различных параметров работы системы «1С:Предприятие» используется пункт [Сервис – Параметры](#).

Параметры предназначены для настройки работы различных режимов программы для конкретного пользователя. Это значит, что при работе нескольких пользователей с одной информационной базой значения параметров устанавливаются для каждого пользователя независимо. Большинство параметров запоминаются при окончании работы с программой и автоматически устанавливаются с теми же значениями при следующем запуске системы.

Все параметры, которые можно установить, объединены в несколько групп, переключение между которыми осуществляется щелчком мыши на той или иной закладке диалога [Параметры](#).

После изменения значений параметров, для того чтобы они были установлены в системе, следует нажать кнопку [ОК](#). При этом новые значения параметров будут установлены и окно параметров будет закрыто.

Нажатие кнопки [Применить](#) также устанавливает новые значения параметров, но при этом окно диалога установки параметров остается открытым.

Для выхода из режима установки параметров без установки выполненных изменений следует нажать кнопку [Отмена](#).

16.3.1.1. Общие параметры

Установка рабочей даты. Рабочая дата используется для определения даты в тех случаях, когда конкретная дата не указана. Например, рабочая дата автоматически проставляется в качестве даты документа при вводе нового документа.

Если флажок [Использовать текущую дату компьютера](#) установлен, то дата выбирается из системной даты компьютера. Если флажок снят, то можно установить новую рабочую дату.

16.3.1.2. Настройка текстового редактора

Управляющие элементы закладки [Тексты](#) предназначены для управления встроенным текстовым редактором системы «1С:Предприятие».

- **Перетаскивание текста** – В редакторе текстов для перемещения и копирования блоков текста можно использовать режим [drag & drop](#) («перенеси и оставь»).
- **Разрешить помещать курсор после концов строк** – Если установлено, то можно помещать курсор после концов строк. В противном случае положение курсора ограничивается положением символа [Перевод строки](#).
- **Отображать пробелы и табуляцию** – Если установлено, то символы пробелов и табуляции будут показываться в тексте. Этот режим устанавливается для проверки форматирования текста. Если установлено свойство [Отображать пробелы и табуляцию](#), то для ввода символов, которыми будет осуществляться показ пробелов и табуляции, становятся доступными свойства [Пробел](#) и [Табуляция](#).
- **Пробел** – Указывается символ показа пробелов.
- **Табуляция** – Указывается символ показа табуляции.
- **Заменять табуляцию при вводе** – Если установлено, то при вводе текста символ табуляции заменяется на установленное в свойстве [Ширина табуляции](#) число пробелов.
- **Ширина табуляции** – Устанавливает соответствие символа табуляции числу символов при вводе текста.
- **Автоотступ** – Устанавливает или отменяет автоматический отступ при вводе символа [Перевод строки](#) (при нажатии клавиши [Enter](#)).
- **Шрифт** – Выбор шрифта, используемого при вводе текста. Для ввода текста модулей рекомендуется

выбирать моноширинные шрифты (например, Courier).

16.3.1.3. Настройка показа справки

На закладке [Справка](#) размещен переключатель, с помощью которого настраивается показ справки.

Если выбрано [Выводить в одном окне](#), то справочная информация показывается в одном окне.

Если выбрано [Выводить в разных окнах](#), то справочная информация показывается в разных окнах.

16.3.1.4. Системные настройки

На закладке [Системные](#) размещены элементы управления подключения к отладочному режиму. Устанавливать переключатели следует только в том случае, если выполняется отладка работы конфигурации.

Если переключатель [Отладка в текущем сеансе разрешена](#) установлен, то устанавливается отладочный режим, и данный предмет отладки включается в список доступных предметов отладки (режим [Конфигуратор](#)). Следует иметь в виду, что после применения настроек снять флажок нельзя.

Если требуется установить возможность отладки только для следующего запуска, то следует установить переключатель [Устанавливать режим разрешения отладки при запуске](#).

16.3.2. Панели инструментов

Панель инструментов представляет собой панель с набором кнопок. Нажатие мышью на любую из кнопок вызывает немедленное выполнение связанной с этой кнопкой команды. Как правило, каждая кнопка снабжена рисунком, образно отражающим характер команды, связанной с этой кнопкой.

Система «1С:Предприятие» в режиме обычного приложения использует панели инструментов двух типов: жестко связанные с окном и плавающие.

Панели, жестко привязанные к какому-либо окну, обычно содержат кнопки, которые предназначены для работы с информацией, расположенной только в этом окне. Они занимают в окне фиксированное положение (обычно под заголовком окна) и не могут изменяться пользователем.

Плавающие панели инструментов содержат, как правило, кнопки для вызова команд общего применения, не привязанных к информации конкретного окна. Количество панелей инструментов, одновременно находящихся в главном окне системы «1С:Предприятие», может меняться в зависимости от режима работы программы. Кроме того, можно определять собственные панели инструментов, komponуя их из заранее заданного набора кнопок.

Плавающие панели инструментов можно располагать в любом удобном месте: как «прижав» к какой-либо стороне окна программы, так и «подвесив» в любом месте экрана. При перемещении панели инструментов ее будущее положение показывается инверсным прямоугольником.

16.3.3. Виды панелей инструментов

В системе «1С:Предприятие» в режиме обычного приложения могут существовать следующие виды плавающих панелей инструментов:

- системные панели инструментов;
- панели, созданные в процессе конфигурирования;
- панели, созданные пользователем.

Системные панели инструментов. Эти панели инструментов «зашиты» в программу и не могут быть удалены ни средствами системы «1С:Предприятие», ни средствами конфигуратора. Можно изменить состав находящихся на них кнопок и даже удалить все кнопки, однако программа «помнит» конфигурацию таких панелей и по команде пользователя может ее восстановить.

Панели инструментов, созданные в процессе настройки конфигурации. Такие панели инструментов создаются при конфигурировании и содержат команды для работы с конкретными элементами данных:

документами, журналами, отчетами и т. п. Как и системные панели, они могут быть изменены и восстановлены. Удалить такие панели можно только при работе в режиме [Конфигуратор](#).

Панели инструментов, создаваемые пользователем. В процессе работы с системой «1С:Предприятие» пользователь может создать свои собственные панели инструментов. Такие панели создаются из заранее заданного набора кнопок, и их количество неограниченно.

16.3.4. Управление панелями инструментов

Управление панелями инструментов ведется при помощи пункта [Сервис – Настройка](#) главного меню программы.

Диалог [Настройка](#) содержит две закладки.

16.3.4.1. Закладка «Панели инструментов»

С помощью управляющих элементов данной закладки выполняется вся работа по созданию, редактированию и удалению панелей инструментов.

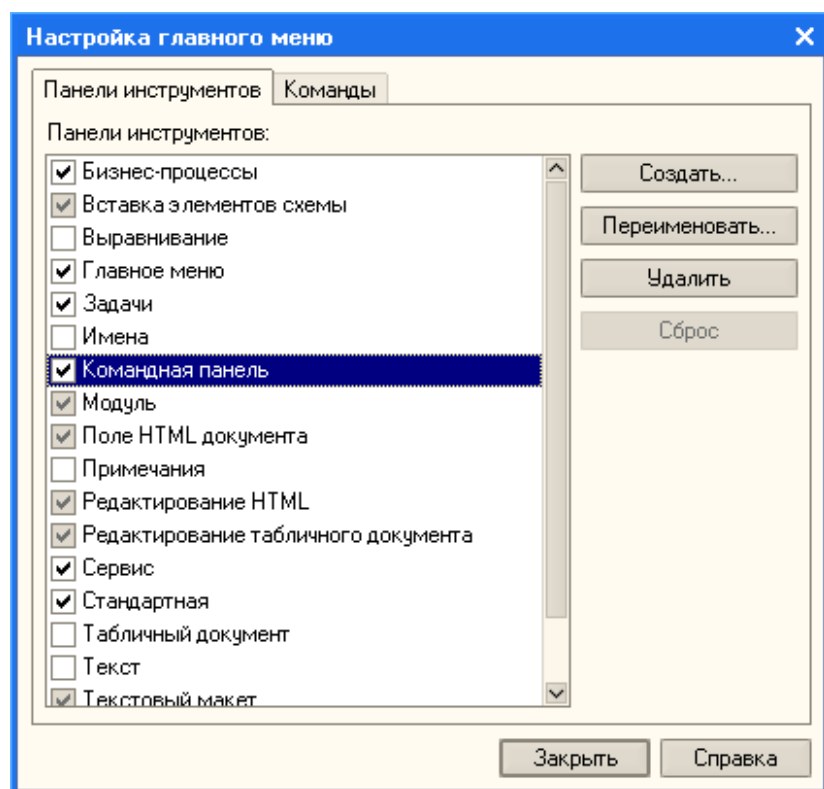


Рис. 236. Диалог настройки панелей инструментов

В списке панелей инструментов флажком отмечены панели, находящиеся в настоящий момент на экране. Флажок можно устанавливать и снимать и таким образом вызывать выбранную панель инструментов на экран и убирать ее.

При первом открытии этого диалога в списке панелей инструментов будут присутствовать только системные панели и панели, созданные в процессе настройки конфигурации. При создании пользователем новых панелей инструментов в список будут добавлены их названия.

16.3.4.2. Создание панели инструментов

Для создания панели инструментов необходимо перейти на закладку [Панели инструментов](#) и нажать кнопку [Создать...](#) В появившемся запросе необходимо ввести имя новой панели инструментов. После нажатия кнопки [ОК](#) на экране появится новая панель инструментов – пока еще без кнопок.

16.3.4.3. Переименование панели инструментов

Переименовывать можно только панели инструментов, созданные пользователем.

Для переименования панели необходимо выбрать ее имя в списке, нажать кнопку [Переименовать](#) и указать

новое имя. После выбора имени другой панели в списке или закрытия окна диалога кнопкой [Заккрыть](#) панели инструментов будет присвоено новое имя.

16.3.4.4. Удаление панели инструментов

Удалить можно только панели инструментов, созданные пользователем.

Для удаления панели необходимо выбрать ее имя в списке и нажать кнопку [Удалить](#). Панель инструментов удаляется без дополнительного предупреждения.

16.3.5. Временная блокировка

Режим временной блокировки предназначен для предотвращения несанкционированного использования системы «1С:Предприятие» в отсутствие пользователя.

Часто конкретная конфигурация использует имя, с которым пользователь входил в систему, для идентификации авторства документов. Зачастую на пользователя возлагается ответственность за выполненные действия. Поэтому важно, чтобы во время сеанса никто не мог бы выполнить какие-либо действия «под чужим именем». В этом случае может быть использован режим временной блокировки.

Режим временной блокировки имеет смысл только тогда, когда для пользователя задан пароль.

Для входа в режим временной блокировки нужно выбрать пункт [Сервис – Временная блокировка](#). Окно приложения системы «1С:Предприятие» минимизируется.

Чтобы продолжить работу, следует щелкнуть по ярлыку приложения. На экран будет выдан диалог для ввода пароля.

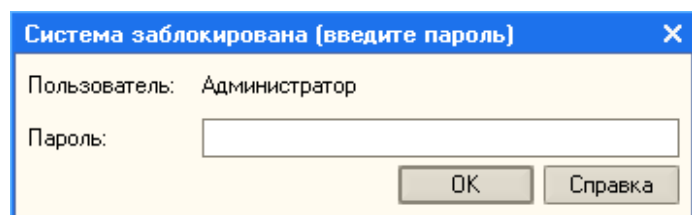


Рис. 237. Диалог для ввода пароля при временной блокировке

Должен быть введен точно такой же пароль, как и при запуске системы. После правильного ввода пароля система становится доступной для работы.

16.4. Установка параметров пользователя

В режиме [1С:Предприятие](#) можно сменить пароль и выбрать язык интерфейса. Для этого нужно выбрать пункт [Сервис – Параметры пользователя](#).

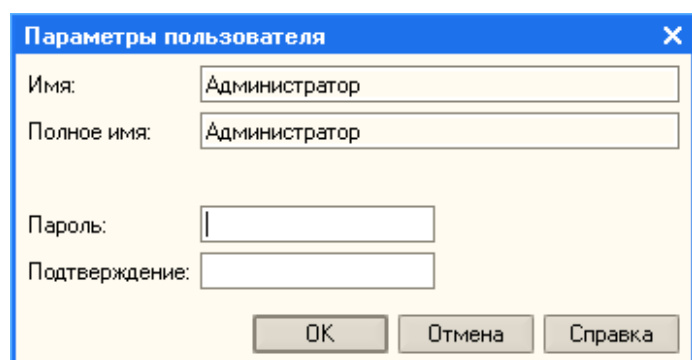


Рис. 238. Настройка параметров пользователя

В поле ввода [Пароль](#) ввести пароль. Пароль представляет собой произвольную строку, состоящую из букв и цифр. Пароль не должен содержать пробелов и специальных символов, размер пароля не должен превышать 255 символов.

При вводе пароль отображается звездочками, поэтому будьте внимательны.

В поле **Подтверждение** нужно указать введенный пароль еще раз для исключения ошибки ввода. Если повторно введенный пароль отличается от первоначального, то при нажатии кнопки **ОК** на экран выводится предупреждение: **Пароль и подтверждение пароля не совпадают**, и пароль не будет установлен.

Если вы передумали устанавливать пароль, следует нажать кнопку **Отмена**.

ВНИМАНИЕ! Присвоенный пароль нельзя посмотреть, поэтому будьте внимательны при установке пароля.

16.5. Сохранение журнала регистрации

В режиме обычного приложения пользователь может сохранить данные журнала регистрации в файл специализированного формата.

Для сохранения журнала регистрации нужно открыть его и выбрать пункт **Файл – Сохранить как**. На экран выводится диалог выбора каталога и файла, в который будет произведена выгрузка, а также тип файла (по умолчанию указывается тип журнала регистрации ***.lgf**). Выгрузка также возможна в формате XML.

16.6. Управление окнами

16.6.1. Панель окон

В нижней части рабочей области программы показывается строка **Панель окон**.

В панели окон размещены ярлыки открытых окон, показываемых в обычном режиме размещения (см. раздел «**Состояние (режим размещения) окон**» [здесь](#)). Для активизации нужного окна следует щелкнуть мышью по его ярлыку.

Панель окон можно спрятать. Для этого в панели окон, главном меню, панели инструментов или панели состояния нужно открыть контекстное меню и снять флажок в строке **Панель окон**. Для показа панели установить флажок.

Чтобы панель окон не занимала постоянно часть рабочей области программы, можно установить режим **Автоматически прятать**. В этом режиме панель окон скрыта. Чтобы она появилась, следует подвести указатель мыши к месту размещения панели.

16.6.1.1. Диалог настройки окон

Диалог предназначен для работы со списком всех открытых окон. С его помощью производится переход к выбранному открытому окну, сохранение сделанных изменений, управление расположением окон, закрытие одного, группы или всех открытых окон.

Для вызова диалога окон нужно выбрать пункт **Окна – Окна...** В диалоге выводится список окон, открытых к настоящему моменту. В список включаются только те окна, которые имеют статус состояния **Обычное**.

Если содержимое открытого окна модифицировано, то такое окно в строке списка помечается символом * (звездочка), располагающимся слева от наименования окна.

Все действия выполняются с одним или несколькими окнами, указанными в списке. Выбор или отмена выбора нескольких строк осуществляется стандартным образом.

Действия, которые выполняются с одним или группой открытых окон, описаны в следующей таблице.

Кнопка	Действие	Число выбранных окон (условие)
Активизировать	Активизирует указанное окно	Одно
Сохранить	Производится сохранение произведенных изменений для некоторых видов документов (например, текстовых и табличных документов), для которых это предусмотрено	Любое
Закрыть окна	Закрываются все выбранные окна	Любое
Расположить	Производится управление расположением окон на экране.	Больше одного

	<p>Горизонтально – располагает выбранные окна слева направо.</p> <p>Вертикально – располагает выбранные окна сверху вниз.</p> <p>Подряд – располагает выбранные окна подряд</p>	
Соединить	Производится соединение двух окон	Установлено разрешение соединения
Разъединить	Производится разъединение соединенных окон	
ОК	Закрывает диалог Окна	Любое
Сортировать	Список окон сортируется по алфавиту	Любое

16.6.2. Служебные окна

В процессе работы с системой «1С:Предприятие» для выдачи пользователю различной вспомогательной информации используются **служебные окна**. К таким окнам относятся:

- табло,
- окно сообщений.

В настоящем разделе изложены общие приемы работы со служебными окнами, а также приведены особенности каждого из перечисленных окон.

16.6.2.1. Табло

Иногда при работе с системой «1С:Предприятие» необходимо контролировать значение того или иного параметра. Для этого можно использовать **табло** – специальное окно, в котором отображаются результаты вычисления введенных в него формул. Вызов табло на экран выполняется выбором пункта меню [Сервис – Табло](#).

Как правило, табло используется администратором системы, т. е. пользователем, знающим встроенный язык системы «1С:Предприятие» и представляющим, какие именно функции можно использовать при расчетах в табло.

Табло представляет собой табличное поле, в каждой строке которого пользователь может ввести формулу и после нажатия клавиши [Enter](#) получить его значение.

Формулы могут включать арифметические выражения, выражения с использованием функций и процедур встроенного языка системы «1С:Предприятия», а также функции и процедуры модуля приложения и общих модулей.

Каждая формула вводится в первую колонку таблицы и должна находиться на отдельной строке. Результат вычисления формулы выдается во второй колонке справа от формулы. Если формула введена неправильно, то вместо результата появится фраза: [Ошибка в выражении!](#)

Данные в табло можно разместить на закладках. Показ закладок производится установкой или снятием флажка в пункте [Закладки](#) контекстного меню.

Результат расчета может быть скопирован в буфер обмена (пункт [Копировать результат](#) контекстного меню второй колонки).

Если в процессе работы исходные данные, используемые в формулах, изменились, то для получения актуальных результатов расчетов необходимо выполнить обновление. Для этого в контекстном меню табло нужно выбрать пункт [Пересчитать](#) или [Пересчитать все](#).

Для очистки содержимого строки табло необходимо выбрать строку и нажать клавишу [Del](#).

16.6.2.2. Окно сообщений

Для выдачи пользователю различной информации, помимо запросов и диалогов, система «1С:Предприятие»

использует специальное **окно сообщений**. Это окно система открывает автоматически, когда в этом возникает необходимость.

В левой части окна напротив каждого сообщения выводится пиктограмма, характеризующая важность данного сообщения.

В любой момент окно сообщений может быть закрыто, но при этом выведенная в нем информация пропадет.

16.7. Состояние (режим размещения) окон

Окна могут размещаться на экране в разных режимах:

- обычное расположение, когда окно располагается только внутри рабочей области программы;
- свободное расположение, когда окно может свободно перемещаться по всему экрану (не только внутри рабочей области программы);
- прикрепленное, когда окно может «прикрепиться» к границам рабочей области программы;
- прячущееся, когда окно может «спрятаться», если с ним в данный момент не работают.

Выбор состояния доступен для окон текстовых и табличных документов, HTML-документов, а также для тех окон, особое состояние которых определено в конфигурации.

Выбор состояния размещения выполняется командами контекстного меню заголовка окна.

Окна в режиме **Свободное** всегда располагаются поверх окон, находящихся в других режимах. Если несколько окон находятся в режиме **Свободное**, то активное окно всегда будет поверх других окон.

Для окон в режиме размещения **Свободное** доступна установка соединения окон. Если для пары окон, находящихся в одинаковом состоянии, установлено свойство **Соединяемое** (в контекстном меню заголовка окна), то данные окна могут соединяться. Чтобы соединить окна, находящиеся в состоянии **Обычное** с установленным свойством **Свободное**, требуется при их перемещении нажать клавишу **Shift**. Для соединенных окон также можно устанавливать способность соединяться с другими окнами.

Если свойство **Соединяемое** не установлено, то выбор режимов **Прикрепленное** и **Прячущееся** недоступен.

Для режимов **Прикрепленное** и **Прячущееся** свойство **Соединяемое** всегда установлено и не может быть изменено.

Особого описания требует режим состояния окна **Прикрепленное**. В этом режиме окно может быть в следующих положениях:

- прикреплено к одной из сторон окна;
- прикреплено к любой стороне другого окна, для которого установлен режим состояния **Прикрепленное**;
- расположено поверх другого прикрепляемого окна (совмещенные окна).

Если для окна выбран режим **Прикрепленное**, то окно прикрепляется к одной из границ рабочей области или к другому окну, находящемуся в режиме **Прикрепленное**.

Для изменения размера прикрепленного окна нужно подвести указатель мыши к свободной (обращенной в сторону рабочей области) границе. Он примет вид раздвоенной стрелки (горизонтальной или вертикальной). Потянуть границу мышью.

Если для окна выбран режим **Свободное**, то в процессе перетаскивания окна при приближении к границе рабочей области программы или другому окну, находящемуся в режиме **Прикрепленное**, контур окна может измениться скачкообразно. Если в этот момент отпустить кнопку мыши, то окно прикрепится и перейдет в состояние **Прикрепленное**.

Для прикрепленных окон изменить состояние можно также с помощью перетаскивания окна мышью. Для перетаскивания окна следует в области заголовка окна нажать левую кнопку мыши и, не отпуская ее, перетащить окно в другое место. Состояние окна меняется на **Свободное**.

Для прикрепляемых окон важной особенностью являются различные способы прикрепления нескольких окон к одной стороне окна или другого окна. Окна можно расположить слоями, когда каждое окно будет занимать всю сторону рабочей области окна или другого окна. Можно расположить окна последовательно, когда каждое окно будет расположено в одном слое вдоль границы другого окна. Если прикрепляемых окон более двух, то можно часть из них расположить слоями, а часть – последовательно.

Для расположения одного прикрепляемого окна (совмещение) поверх другого нужно выполнить следующие действия:

- захватить мышью заголовок окна;
- переместить окно так, чтобы его заголовок оказался над заголовком другого прикрепляемого окна. При этом контур перетаскиваемого окна должен иметь снизу контур закладки;
- отпустить кнопку мыши.

Результатом этих действий будет прикрепляемое окно, внизу которого будут закладки. Текст закладок совпадает с текстом заголовков окон. На закладках располагаются совмещенные окна.

Для показа нужного окна следует щелкнуть нужную закладку. Команда [Заккрыть](#) в этом случае закрывает только активную закладку.

Чтобы разделить совмещенные окна, достаточно захватить мышью закладку нужного окна и перетащить ее в сторону (контролируйте перенос контуром перемещаемого окна).

Если при переносе указатель мыши остается в области закладок, то таким способом можно изменить порядок следования закладок.

Если на экране есть совмещенные окна, то при совмещении с ними другого окна можно перетаскивать его не на заголовок, а в область закладок. При этом можно сразу выбрать порядок следования закладок.

Если выбран режим [Прячущееся](#), то сбоку рабочей области программы появляется дополнительная строка для ярлыков окон, в которую добавляется ярлык текущего окна, а само окно прикрепляется к этой стороне рабочей области. Расположение этой строки определяется расположением окна в режиме [Прикрепленное](#). В этом режиме окно остается на экране, пока активно. Если активизировать любое другое окно, предыдущее окно прячется. Чтобы показать это окно, необходимо подвести указатель мыши к ярлыку данного окна (щелкать кнопкой мыши не нужно). При выводе указателя мыши за окно прячущееся окно автоматически убирается с экрана.

Приложение 1. Язык выражений системы компоновки данных

Язык выражений системы компоновки данных предназначен для записи выражений, используемых в различных частях системы, например, в настройках компоновки данных, для описания выражений пользовательских полей.

Литералы

В выражении могут присутствовать литералы. Возможны литералы следующих типов:

- Строка,
- Число,
- Дата,
- Булево.

Строка

Строковый литерал записывается в символах «"».

"Строковый литерал"

[Копировать в буфер обмена](#)

При необходимости использования внутри строкового литерала символа «"» следует использовать два таких символа.

"Литерал "" в кавычках""

[Копировать в буфер обмена](#)

Число

Число записывается без пробелов, в десятичном формате. Дробная часть отделяется при помощи символа «.».

10.5

[Копировать в буфер обмена](#)

200

[Копировать в буфер обмена](#)

Дата

Литерал типа *Дата* записывается при помощи ключевого литерала *ДАТАВРЕМЯ* (*DATETIME*). После данного ключевого слова в скобках, через запятую, перечисляются год, месяц, день, часы, минуты, секунды. Указание времени необязательно.

ДАТАВРЕМЯ(1975, 1, 06) – Шестое января 1975 года

[Копировать в буфер обмена](#)

ДАТАВРЕМЯ(2006, 12, 2, 23, 56, 57) – Второе декабря 2006 года, 23 часа 56 минут 57 секунд

[Копировать в буфер обмена](#)

Булево

Булевы значения могут быть записаны при помощи литералов *Истина* (*True*), *Ложь* (*False*).

Значение

Для указания литералов других типов (системных перечислений, предопределенных данных) используется ключевое слово *ЗНАЧЕНИЕ*, после которого в скобках идет указание имени литерала.

ЗНАЧЕНИЕ(ВидСчета.Активный)

[Копировать в буфер обмена](#)

Поля

В выражениях могут использоваться поля наборов данных. Поле идентифицируется путем к данным. Части пути к данным отделяются друг от друга символом «.».. Имя поля не является чувствительным к регистру.

Номенклатура.Артикул

[Копировать в буфер обмена](#)

Продажи.СуммаОборот

[Копировать в буфер обмена](#)

Параметры

Выражения могут использовать параметры. Для использования в выражении параметра достаточно написать его имя, которому будет предшествовать символ *&*.

&Контрагент

[Копировать в буфер обмена](#)

&ДатаНачала

[Копировать в буфер обмена](#)

1.1. Операции над числами

Унарный –

Данная операция предназначена для изменения знака числа на обратный.

–Продажи.Количество

[Копировать в буфер обмена](#)

Унарный +

Данная операция не выполняет над числом никаких действий.

`+Продажи.Количество`

[Копировать в буфер обмена](#)

Бинарный –

Данная операция предназначена для вычисления разности двух чисел.

`ОстаткиОбороты.НачальныйОстаток - ОстаткиОбороты.КонечныйОстаток`

[Копировать в буфер обмена](#)

`ОстаткиОбороты.НачальныйОстаток - 100`

[Копировать в буфер обмена](#)

`400 - 357`

[Копировать в буфер обмена](#)

Бинарный +

Данная операция предназначена для вычисления суммы двух чисел.

`ОстаткиОбороты.НачальныйОстаток + ОстаткиОбороты.Оборот`

[Копировать в буфер обмена](#)

`ОстаткиОбороты.НачальныйОстаток + 100`

[Копировать в буфер обмена](#)

`400 + 357`

[Копировать в буфер обмена](#)

Произведение

Данная операция предназначена для вычисления произведения двух чисел.

`Номенклатура.Цена * 1.2`

[Копировать в буфер обмена](#)

`2 * 3.14`

[Копировать в буфер обмена](#)

Деление

Данная операция предназначена для получения результата деления одного операнда на другой.

`Номенклатура.Цена / 1.2`

[Копировать в буфер обмена](#)

`2 / 3.14`

[Копировать в буфер обмена](#)

Остаток от деления

Данная операция предназначена для получения остатка от деления одного операнда на другой.

`Номенклатура.Цена % 1.2`

[Копировать в буфер обмена](#)

`2 % 3.14`

[Копировать в буфер обмена](#)

1.2. Операции над строками

Конкатенация (Бинарный +)

Данная операция предназначена для конкатенации двух строк.

`Номенклатура.Артикул + " : "+" Номенклатура.Наименование`

[Копировать в буфер обмена](#)

ПОДОБНО (LIKE)

Данная операция проверяет соответствие строки переданному шаблону.

Значением оператора **ПОДОБНО** является **Истина**, если значение выражения удовлетворяет шаблону, и **Ложь** в противном случае.

Следующие символы в строке шаблона имеют смысл, отличный от просто очередного символа строки:

- **%** – процент: последовательность, содержащая ноль и более произвольных символов;
- **_** – подчеркивание: один произвольный символ;
- **[...]** – один или несколько символов в квадратных скобках: один символ, любой из перечисленных внутри квадратных скобок. В перечислении могут встречаться диапазоны, например `a-z`, означающие произвольный символ, входящий в диапазон, включая концы диапазона;
- **[^...]** – в квадратных скобках значок отрицания, за которым следует один или несколько символов: любой символ, кроме тех, которые перечислены следом за значком отрицания.

Любой другой символ означает сам себя и не несет никакой дополнительной нагрузки. Если в качестве самого себя необходимо записать один из перечисленных символов, то ему должен предшествовать спецсимвол, указанный после ключевого слова **СПЕЦСИМВОЛ** (**ESCAPE**).

Например, приведенный ниже шаблон означает подстроку, состоящую из последовательности символов: буквы **A**; буквы **B**; буквы **V**; одной цифры; одной из букв **a**, **b**, **v** или **г**; символа подчеркивания; буквы **a**; буквы **b**; буквы **v**. Причем эта последовательность может располагаться начиная с произвольной позиции в строке.

`"%АВВ[0-9][абвг]_абв%" СПЕЦСИМВОЛ "\ "`

[Копировать в буфер обмена](#)

1.3. Операции сравнения

РАВНО

Данная операция предназначена для сравнения двух операндов на равенство.

`Продажи.Контрагент = Продажи.НоменклатураОсновнойПоставщик`

[Копировать в буфер обмена](#)

НЕ РАВНО

Данная операция предназначена для сравнения двух операндов на неравенство.

`Продажи.Контрагент <> Продажи.НоменклатураОсновнойПоставщик`

[Копировать в буфер обмена](#)

МЕНЬШЕ

Данная операция предназначена для проверки того, что первый операнд меньше второго.

`ПродажиТекущие.Сумма < ПродажиПрошлые.Сумма`

[Копировать в буфер обмена](#)

БОЛЬШЕ

Данная операция предназначена для проверки того, что первый операнд больше второго.

`ПродажиТекущие.Сумма > ПродажиПрошлые.Сумма`

[Копировать в буфер обмена](#)

МЕНЬШЕ ИЛИ РАВНО

Данная операция предназначена для проверки того, что первый операнд меньше либо равен второму.

`ПродажиТекущие.Сумма <= ПродажиПрошлые.Сумма`

[Копировать в буфер обмена](#)

БОЛЬШЕ ИЛИ РАВНО

Данная операция предназначена для проверки того, что первый операнд больше либо равен второму.

`ПродажиТекущие.Сумма >= ПродажиПрошлые.Сумма`

[Копировать в буфер обмена](#)

Операция «В» (IN)

Данная операция осуществляет проверку наличия значения в переданном списке значений. Результатом операции будет *Истина*, если значение найдено, или *Ложь* – в противном случае.

`Номенклатура В (&Товар1, &Товар2)`

[Копировать в буфер обмена](#)

Операция проверки наличия значения в наборе данных

Операция осуществляет проверку наличия значения в указанном наборе данных. Набор данных для проверки должен содержать одно поле.

`Продажи.Контрагент В Контрагенты`

[Копировать в буфер обмена](#)

Операция проверки значения на NULL ЕСТЬ NULL (IS NULL)

Данная операция возвращает значение *Истина*, если оно является значением *NULL*.

`Продажи.Контрагент ЕСТЬ NULL`

[Копировать в буфер обмена](#)

Операция проверки значения на неравенство NULL ЕСТЬ НЕ NULL (IS NOT NULL)

Данная операция возвращает значение *Истина*, если оно не является значением *NULL*.

`Продажи.Контрагент ЕСТЬ НЕ NULL`

[Копировать в буфер обмена](#)

1.4. Логические операции

Логические операции принимают в качестве операндов выражения, имеющие тип *Булево*.

Операция «НЕ» (NOT)

Операция *НЕ* возвращает значение *Истина*, если ее операнд имеет значение *Ложь*, и *Ложь*, если ее операнд имеет значение *Истина*.

`НЕ Документ.Грузополучатель = Документ.Грузоотправитель`

[Копировать в буфер обмена](#)

Операция «И» (AND)

Операция *И* возвращает значение *Истина*, если оба операнда имеют значение *Истина*, и *Ложь*, если один из операндов имеет значение *Ложь*.

`Документ.Грузополучатель = Документ.Грузоотправитель И Документ.Грузополучатель = &Контрагент`

[Копировать в буфер обмена](#)

Операция «ИЛИ» (OR)

Операция *ИЛИ* возвращает значение *Истина*, если один из операндов имеет значение *Истина*, и *Ложь*, если оба операнда имеют значение *Ложь*.

[Документ.Грузополучатель](#) = [Документ.Грузоотправитель](#) ИЛИ [Документ.Грузополучатель](#) = &[Контрагент](#)

1.5. Агрегатные функции

Агрегатные функции осуществляют некоторое действие над набором данных.

СУММА (SUM)

Агрегатная функция **СУММА** рассчитывает сумму значений выражений, переданных ей в качестве аргумента для всех детальных записей. В качестве параметра может быть использован результат функции **Массив**.

[СУММА\(Продажи.СуммаОборот\)](#)

[Копировать в буфер обмена](#)

КОЛИЧЕСТВО (COUNT)

Функция **КОЛИЧЕСТВО** рассчитывает количество значений, отличных от значения **NULL**. В качестве параметра может быть использован результат функции **Массив**. Например:

[КОЛИЧЕСТВО\(Продажи.Контрагент\)](#)

[Копировать в буфер обмена](#)

КОЛИЧЕСТВО (РАЗЛИЧНЫЕ) (COUNT (DISTINCT))

Эта функция рассчитывает количество различных значений. В качестве параметра может быть использован результат функции **Массив**.

[КОЛИЧЕСТВО\(РАЗЛИЧНЫЕ Продажи.Контрагент\)](#)

[Копировать в буфер обмена](#)

МАКСИМУМ (MAX)

Функция получает максимальное значение. В качестве параметра может быть использован результат функции **Массив**.

[МАКСИМУМ\(Остатки.Количество\)](#)

[Копировать в буфер обмена](#)

МИНИМУМ (MIN)

Функция получает минимальное значение. В качестве параметра может быть использован результат функции **Массив**.

[МИНИМУМ\(Остатки.Количество\)](#)

[Копировать в буфер обмена](#)

СРЕДНЕЕ (AVG)

Функция получает среднее значение для значений, отличных от **NULL**. В качестве параметра может быть использован результат функции **Массив**.

[СРЕДНЕЕ\(Остатки.Количество\)](#)

[Копировать в буфер обмена](#)

МАССИВ (ARRAY)

Функция возвращает массив, содержащий для каждой детальной записи значение параметра. Если выражение, указанное в параметре, содержит функцию **Массив**, то считается, что данное выражение является агрегатным. В качестве параметра для функции указывается выражение произвольного типа.

В качестве параметра может быть указано выражение, возвращающее таблицу значений. При этом результатом работы функции **Массив** будет массив, содержащий значения первой колонки таблицы значений, переданной в качестве параметра.

[МАССИВ\(КоличествоПредложений\)](#)

[Копировать в буфер обмена](#)

ТАБЛИЦАЗНАЧЕНИЙ (VALUETABLE)

Функция возвращает таблицу значений, содержащую столько колонок, сколько параметров у функции. Детальные записи получаются из наборов данных, которые нужны для получения всех полей, участвующих в выражениях параметров функции. Если выражение содержит функцию **ТаблицаЗначений**, то считается, что данное выражение является агрегатным.

У функции может быть один или несколько параметров произвольного типа. После каждого параметра может располагаться необязательное ключевое слово **КАК** и имя, которое будет назначено колонке таблицы значений.

[ТаблицаЗначений\(Различные Номенклатура, ХарактеристикаНоменклатуры КАК Характеристика\)](#)

[Копировать в буфер обмена](#)

ГРУППОВАЯОБРАБОТКА (GROUPPROCESSING)

Функция возвращает объект **ДанныеГрупповойОбработкиКомпоновкиДанных** (`DataCompositionGroupProcessingData`), который имеет следующие свойства:

- **Данные (Data)**. Тип – таблица значений. В это свойство функция помещает таблицу значений, содержащую результаты вычисления выражения, указанного в первом параметре функции для каждой групповой записи группировки. Если группировка иерархическая, то каждый уровень иерархии обрабатывается функцией отдельно, при этом значения для иерархических записей также помещаются в данные.
- **ТекущийЭлемент (CurrentItem)** – строка таблицы значений, являющихся текущими. При вызове для общего итога **ТекущийЭлемент** содержит значение **Неопределено**. В данное свойство функция помещает строку таблицы значений, для которой в настоящий момент вычисляется функция.
- **ВременныеДанныеОбработки (ProcessingTempData)** – структура, в которую могут быть помещены данные промежуточных результатов. Рекомендуется в различных функциях давать свойствам уникальные имена, т. к. возможна ситуация, когда один и тот же объект **ДанныеГрупповойОбработкиКомпоновкиДанных** будет передан в несколько различных функций.

Функция имеет следующие параметры:

- **Выражения** – строка, в которой через запятую перечислены выражения, которые нужно вычислить. После каждого выражения возможно наличие необязательного ключевого слова **КАК** и имени колонки результирующей таблицы значений.
- **ВыраженияИерархии** – выражения, которые нужно вычислить для иерархических записей. Аналогично параметру **Выражения** с тем отличием, что параметр

Выражения используется для неиерархических записей, а параметр **ВыраженияИерархии** используется для иерархических записей. Если параметр не указан, то для вычисления значений для иерархических записей используются выражения, указанные в параметре **Выражение**.

● **ИмяГруппировки** – имя группировки, в которой нужно вычислять группировку обработки. Тип **Строка**. Если не указано, то вычисление происходит в текущей группировке. Если вычисление идет в таблице и параметр содержит пустую строку или не указан, то значение вычисляется для группировки-строки. Компоновщик макета при генерации макета компоновки данных заменяет данное имя именем группировки в результирующем макете. Если группировка недоступна, то функция будет заменена на значение **NULL**.

[Копировать в буфер обмена](#)

ГрупповаяОбработка ("Сумма (СуммаОборот) ")

СВЕРНУТЬ (GROUPBY)

Функция предназначена для удаления дубликатов из массива. Возвращает массив или таблицу значений, в которой отсутствуют дублирующиеся элементы.

Параметры:

- **Массив** или таблица значений;
- номера или имена колонок таблицы значений, среди которых нужно искать дубликаты. По умолчанию все колонки (через запятую).

ПОЛУЧИТЬЧАСТЬ(GETPART)

Функция возвращает таблицу значений, содержащую определенные колонки из исходной таблицы значений.

Параметры:

- таблица значений, из которой нужно получить колонки;
- номера или имена колонок, которые нужно получить, разделенные запятой.

Возвращаемое значение: таблица значений, включающая только колонки, которые указаны в параметре **Номера/имена колонок**.

УПОРЯДОЧИТЬ(ORDER)

Предназначена для упорядочивания элементов массива.

Параметры:

- **Массив** или **ТаблицаЗначений**;
- номер или имя колонки таблицы значений, по которой нужно упорядочить. Для массива номер можно не указывать. Направление упорядочивания **Необходимость автоупорядочивания**. **Убыв/Возр + Автоупорядочивание**.

Возвращаемое значение: массив или таблица значений с упорядоченными элементами.

СОЕДИНИТЬСТРОКИ (JOINSTRINGS)

Агрегатная функция, позволяющая объединять строки в одну строку.

Параметры:

- **Значения, которые нужно объединить в одну строку**. Если является массивом, то в строку будут объединяться элементы массива. Если является таблицей значений, то в строку будут объединяться все колонки и строки таблицы.
- **Разделитель элементов**. Строка, содержащая текст, который нужно использовать в качестве разделителя между элементами массива и строками таблицы значений. По умолчанию символ перевода строк.
- **Разделители колонок**. Строка, содержащая текст, который нужно использовать в качестве разделителя между колонками таблицы значений. По умолчанию «;».

Каждый (Every)

Если хоть одна запись имеет значение **Ложь**, то функция возвращает **Ложь**. Иначе – **Истина**.

Синтаксис:

Каждый (Выражение)

Параметр:

Выражение – тип **Булево**.

Пример:

[Копировать в буфер обмена](#)

Каждый ()

Любой (Any)

Если хоть одна запись имеет значение **Истина**, то результат – **Истина**. Иначе – **Ложь**.

Синтаксис:

Любой (Выражение)

Параметр:

Выражение – тип **Булево**.

Пример:

[Копировать в буфер обмена](#)

Любой ()

Стандартное Отклонение Генеральной Совокупности (Stddev_Pop)

Вычисляет стандартное отклонение совокупности. Используется формула: $\text{SQRT}(\text{ДисперсияГенеральнойСовокупности}(X))$

Синтаксис: *СтандартноеОтклонениеГенеральнойСовокупности(Выражение)*

Параметр:

Выражение – тип *Число*.

Тип возвращаемого значения – *Число*.

Пример:

X	Y
1	7
2	1
3	2
4	5
5	7
6	34
7	32
8	43
9	87

[Копировать в буфер обмена](#)

ВЫБРАТЬ ДисперсияВыборки(Y, X) ИЗ Таблица

Результат:

805.694444

[Копировать в буфер обмена](#)

Стандартное Отклонение Выборки (Stddev_Samp)

Функция вычисляет совокупное типовое стандартное отклонение. Используется формула: $\text{SQRT}(\text{ДисперсияВыборки}(X))$

Синтаксис:

СтандартноеОтклонениеВыборки(Выражение)

Параметр:

Выражение – тип *Число*.

Тип возвращаемого значения – *Число*.

Пример:

X	Y
1	7
2	1
3	2
4	5
5	7
6	34
7	32
8	43
9	87

[Копировать в буфер обмена](#)

ВЫБРАТЬ СтандартноеОтклонениеВыборки(Y) ИЗ Таблица

Результат:

28.3847573

[Копировать в буфер обмена](#)

Дисперсия Выборки (Var_Samp)

Функция вычисляет типовое различие ряда чисел без учета значений **NULL** в этом наборе. Используется формула: $(\text{Сумма}(X^2) - \text{Сумма}(X)^2 / \text{Количество}(X)) / (\text{Количество}(X) - 1)$.

Если $\text{Количество}(X) = 1$, то возвращается значение **NULL**.

Синтаксис:

ДисперсияВыборки(Выражение)

Параметр:

Выражение – тип *Число*.

688

Пример:

X	Y
1	7
2	1
3	2
4	5
5	7
6	34
7	32
8	43
9	87

[Копировать в буфер обмена](#)

ВЫБРАТЬ ДисперсияВыборки(Y, X) ИЗ Таблица

Результат:

[Копировать в буфер обмена](#)

805.694444

ДисперсияГенеральнойСовокупности (Var_Pop)

Функция вычисляет различие совокупности ряда чисел без учета значений **NULL** в этом наборе. Используется формула: $(\text{Сумма}(X^2) - \text{Сумма}(X)^2 / \text{Количество}(X)) / \text{Количество}(X)$.

Синтаксис:

ДисперсияГенеральнойСовокупности(Выражение)

Параметр:

Выражение – тип Число.

Пример:

X	Y
1	7
2	1
3	2
4	5
5	7
6	34
7	32
8	43
9	87

[Копировать в буфер обмена](#)

ВЫБРАТЬ ДисперсияГенеральнойСовокупности(Y, X) ИЗ Таблица

Результат:

[Копировать в буфер обмена](#)

716.17284

КовариацияГенеральнойСовокупности (Covar_Pop)

Вычисляет ковариацию ряда числовых пар. Используется формула: $(\text{Сумма}(Y * X) - \text{Сумма}(X) * \text{Сумма}(Y) / n) / n$, где n число пар (Y, X), в которых ни Y ни X не являются **NULL**.

Синтаксис:

КовариацияГенеральнойСовокупности(Y, X)

Параметры:

- Y – тип Число;
- X – тип Число.

Пример:

X	Y
1	7
2	1
3	2
4	5
5	7
6	34

7	32
8	43
9	87

[Копировать в буфер обмена](#)**ВЫБРАТЬ** КовариацияГенеральнойСовокупности(Y, X) ИЗ Таблица

Результат:

59.4444444

[Копировать в буфер обмена](#)

КовариацияВыборки (Covar_Samp)

Вычисляет типовое различие ряда чисел без учета значений `NULL` в этом наборе. Используется формула: $(\text{Сумма}(Y * X) - \text{Сумма}(Y) * \text{Сумма}(X) / n) / (n-1)$, где `n` число пар `(Y, X)` в которых ни `Y`, ни `X` не являются `NULL`.

Синтаксис:

КовариацияВыборки(Y, X)

Параметры:

- `Y` – тип Число;
- `X` – тип Число.

Пример:

X	Y
1	7
2	1
3	2
4	5
5	7
6	34
7	32
8	43
9	87

[Копировать в буфер обмена](#)**ВЫБРАТЬ** КовариацияВыборки(Y, X) ИЗ Таблица

Результат:

66.875

[Копировать в буфер обмена](#)

Корреляция (Corr)

Функция вычисляет коэффициент корреляции ряда числовых пар. Используется формула:

$\text{КовариацияГенеральнойСовокупности}(Y, X) / (\text{СтандартноеОтклонениеГенеральнойСовокупности}(Y) * \text{СтандартноеОтклонениеГенеральнойСовокупности}(X))$.
 Не учитываются пары, в которых `Y` или `X` – `NULL`.

Синтаксис:

Корреляция(Y, X)

Параметры:

- `Y` – тип Число;
- `X` – тип Число.

Пример:

X	Y
1	7
2	1
3	2
4	5
5	7
6	34
7	32
8	43
9	87

[Копировать в буфер обмена](#)**ВЫБРАТЬ** Корреляция(X, Y) ИЗ Таблица

Результат:

0.860296149

РегрессияНаклон (Regr_Slope)

Функция вычисляет наклон линии. Используется формула: $\frac{\text{КовариацияГенеральнойСовокупности}(Y, X)}{\text{ДисперсияГенеральнойСовокупности}(X)}$. Вычисляется без учета пар, содержащих `NULL`.

Синтаксис:

РегрессияНаклон(`Y`, `X`)

Параметры:

- `Y` – тип Число;
- `X` – тип Число.

Пример:

X	Y
1	7
2	1
3	2
4	5
5	7
6	34
7	32
8	43
9	87

[Копировать в буфер обмена](#)

[ВЫБРАТЬ](#) [РегрессияНаклон\(Y, X\)](#) ИЗ [Таблица](#)

Результат:

8.91666667

[Копировать в буфер обмена](#)

РегрессияОтрезок (Regr_Intercept)

Функция вычисляет Y-точку пересечения линии регресса. Используется формула: $\text{Среднее}(Y) - \text{РегрессияНаклон}(Y, X) * \text{Среднее}(X)$. При вычислении не учитываются пары, содержащие `NULL`.

Синтаксис:

РегрессияОтрезок(`Y`, `X`)

Параметры:

- `Y` – тип Число;
- `X` – тип Число.

Пример:

X	Y
1	7
2	1
3	2
4	5
5	7
6	34
7	32
8	43
9	87

[Копировать в буфер обмена](#)

[ВЫБРАТЬ](#) [РегрессияОтрезок\(Y, X\)](#) ИЗ [Таблица](#)

Результат:

-20.361111

[Копировать в буфер обмена](#)

РегрессияКоличество (Regr_Count)

Вычисляет количество пар, не содержащих `NULL`.

Синтаксис:

РегрессияКоличество(`Y`, `X`)

Параметры:

- Y – тип Число;
- X – тип Число.

Пример:

X	Y
1	7
2	1
3	2
4	5
5	7
6	34
7	32
8	43
9	87

[Копировать в буфер обмена](#)

[ВЫБРАТЬ РегрессияКоличество\(Y, X\) ИЗ Таблица](#)

Результат:

[Копировать в буфер обмена](#)

9

РегрессияR2 (Regr_R2)

Вычисляет коэффициент детерминации. Вычисляется без учета пар, содержащих NULL.

Синтаксис:

`РегрессияR2(Y, X)`

Параметры:

- Y – тип Число;
- X – тип Число.

Возвращаемое значение:

NULL – если ДисперсияГенеральнойСовокупности(X)=0; 1 – если ДисперсияГенеральнойСовокупности(Y)=0 и ДисперсияГенеральнойСовокупности(X) <> 0; POW(Корреляция(Y, X), 2) – если ДисперсияГенеральнойСовокупности(Y)>0 и ДисперсияГенеральнойСовокупности(X) <> 0.

Пример:

X	Y
1	7
2	1
3	2
4	5
5	7
6	34
7	32
8	43
9	87

[Копировать в буфер обмена](#)

[ВЫБРАТЬ РегрессияR2\(Y, X\) ИЗ Таблица](#)

Результат:

[Копировать в буфер обмена](#)

0.740109464

РегрессияСреднееX (Regr_AvgX)

Вычисляет среднее число Y после исключения X и Y пар, где или X, или Y являются пустыми. Среднее (X) вычисляется без учета пар, содержащих NULL.

Синтаксис:

`РегрессияСреднееX(Y, X)`

Параметры:

- Y – тип Число;
- X – тип Число.

Пример:

X	Y
1	7
2	1
3	2
4	5
5	7
6	34
7	32
8	43
9	87

692

X	Y
1	7
2	1
3	2
4	5
5	7
6	34
7	32
8	43
9	87

[Копировать в буфер обмена](#)

[ВЫБРАТЬ](#) РегрессияСреднееX(Y, X) ИЗ Таблица

Результат:

5

[Копировать в буфер обмена](#)

РегрессияСреднееY (Regr_AvgY)

Функция вычисляет среднее число **X** после исключения **X** и **Y** пар, где или **X**, или **Y** являются пустыми. Среднее (**Y**) вычисляется без учета пар, содержащих **NULL**.

Синтаксис:

РегрессияСреднееY (Y, X)

Параметры:

- **Y** – тип Число;
- **X** – тип Число.

Пример:

X	Y
1	7
2	1
3	2
4	5
5	7
6	34
7	32
8	43
9	87

[Копировать в буфер обмена](#)

[ВЫБРАТЬ](#) РегрессияСреднееY(Y, X) ИЗ Таблица

Результат:

24.2222222

[Копировать в буфер обмена](#)

РегрессияSXX (Regr_SXX)

Функция возвращает сумму квадратов независимых выражений, используемых в линейной модели регресса. Функция может использоваться, чтобы оценить статистическую обоснованность модели регресса.

Значение вычисляется по формуле: РегрессияКоличество(Y, X) * ДисперсияГенеральнойСовокупности(X). При этом пары, содержащие **NULL**, не учитываются.

Синтаксис:

РегрессияSXX (Y, X)

Параметры:

- **Y** – тип Число;
- **X** – тип Число.

Пример:

X	Y
1	7
2	1
3	2
4	5
5	7

6	34
7	32
8	43
9	87

[ВЫБРАТЬ РегрессияSXX\(Y, X\) ИЗ Таблица](#)

[Копировать в буфер обмена](#)

Результат:

60

[Копировать в буфер обмена](#)

РегрессияSY (Regr_SY)

Функция вычисляет значение по формуле: $\text{РегрессияКоличество}(Y, X) * \text{ДисперсияГенеральнойСовокупности}(Y)$.

Значение вычисляется без учета пар, содержащих NULL.

Синтаксис:

$\text{РегрессияSY}(Y, X)$

Параметры:

- Y – тип Число;
- X – тип Число.

Пример:

X	Y
1	7
2	1
3	2
4	5
5	7
6	34
7	32
8	43
9	87

[ВЫБРАТЬ РегрессияSY\(Y, X\) ИЗ Таблица](#)

[Копировать в буфер обмена](#)

Результат:

6445.55556

[Копировать в буфер обмена](#)

РегрессияSXY (Regr_SXY)

Функция вычисляет значение по формуле: $\text{РегрессияКоличество}(Y, X) * \text{КовариацияГенеральнойСовокупности}(Y, X)$. Вычисляется без учета пар, содержащих NULL.

Синтаксис:

$\text{РегрессияSXY}(Y, X)$

Параметры:

- Y – тип Число;
- X – тип Число.

Пример:

X	Y
1	7
2	1
3	2
4	5
5	7
6	34
7	32
8	43
9	87

[ВЫБРАТЬ РегрессияSXY\(Y, X\) ИЗ Таблица](#)

[Копировать в буфер обмена](#)

535

МестоВПорядке (Rank)

Синтаксис:

МестоВПорядке(Порядок, ПорядокИерархии, ИмяГруппировки)

Параметры:

- **Порядок** – тип *Строка*. Содержит выражения, в последовательности которых нужно расположить групповые записи, разделенные через запятую. Направление упорядочивания управляется при помощи слов **Возр**, **Убыв**. После поля также можно указать строку **Автоупорядочивание**, что обозначает, что при упорядочивании ссылок нужно использовать поля упорядочивания, определенные для объекта, на который ссылка. Если последовательность не указана, то значение рассчитывается в последовательности группировки;
- **ПорядокИерархии** – тип *Строка*. Содержит выражения упорядочивания для иерархических записей;
- **ИмяГруппировки** – тип *Строка*. Имя группировки, в которой нужно вычислять группировку обработки. Если не указано, то вычисление происходит в текущей группировке. Если вычисление идет в таблице и параметр содержит пустую строку или не указан, то значение вычисляется для группировки-строки. Компоновщик макета при генерации макета компоновки данных заменяет данное имя на имя группировки в результирующем макете. Если группировка не доступна, то функция будет заменена на значение **NULL**.

Пример:

```
МестоВПорядке (" [Количество Оборот] ")
```

[Копировать в буфер обмена](#)

КлассификацияABC (ClassificationABC)

Функция возвращает номер класса, начиная с 1, который соответствует классу А.

Синтаксис:

КлассификацияABC(Значение, КоличествоГрупп, ПроцентыДляГрупп, ИмяГруппировки)

Параметры:

- **Значение** – тип *Строка*. Значение, по которому нужно рассчитывать классификацию. Строка, в которой указано выражение;
- **КоличествоГрупп** – тип *Число*. Задаёт количество групп, на которое нужно разбить;
- **ПроцентыДляГрупп** – тип *Число*. Количество групп, на которое нужно разбить минус 1. Указывается через запятую;
- **ИмяГруппировки** – тип *Строка*. Имя группировки, в которой нужно вычислять группировку обработки. Если не указано, то вычисление происходит в текущей группировке. Если вычисление выполняется в таблице и параметр содержит пустую строку или не указан, то значение вычисляется для группировки-строки. Данное имя будет заменено на имя группировки в результирующем макете. Если группировка не доступна, то функция будет заменена на значение **NULL**.

Пример:

```
КлассификацияABC("Сумма([Количество Оборот] )", 3)
```

[Копировать в буфер обмена](#)

1.6. Другие операции

Операция «ВЫБОР»

Операция **ВЫБОР** предназначена для осуществления выбора одного из нескольких значений при выполнении некоторых условий.

```
ВЫБОР Когда Сумма > 1000 Тогда Сумма Иначе 0 Конец
```

[Копировать в буфер обмена](#)

Правила сравнения двух значений

Если типы сравниваемых значений отличаются друг от друга, то отношения между значениями определяются на основании приоритета типов:

- **NULL** (самый низший),
- **Булево**,
- **Число**,
- **Дата**,
- **Строка**,
- **ссылочные типы**.

Отношения между различными ссылочными типами определяются на основе ссылочных номеров таблиц, соответствующих тому или иному типу.

Если типы данных совпадают, то производится сравнение значений по следующим правилам:

- у типа **Булево** значение **ИСТИНА** больше значения **ЛОЖЬ**;
- у типа **Число** обычные правила сравнения для чисел;
- у типа **Дата** более ранние даты меньше более поздних;
- у типа **Строка** строки сравниваются в соответствии с установленными национальными особенностями базы данных;
- **ссылочные типы** сравниваются на основе своих значений (номера записи и т. п.).

Работа со значением NULL

Любая операция, в которой значение одного из операндов `NULL`, будет давать результат `NULL`.

Есть исключения:

- операция **И** будет возвращать `NULL` только в случае, если ни один из операндов не имеет значения `Ложь`;
- операция **ИЛИ** будет возвращать `NULL` только в случае, если ни один из операндов не имеет значение `Истина`.

Приоритеты операций

Операции имеют следующие приоритеты (первая строка имеет низший приоритет):

- **ИЛИ**;
- **И**;
- **НЕ**;
- **В, ЕСТЬ NULL, ЕСТЬ НЕ NULL**;
- **=, <>, <=, <, >=, >**;
- **Бинарный +, Бинарный -**;
- ***, /, %**;
- **Унарный +, Унарный -**.

1.7. Функции

ВЫЧИСЛИТЬ (EVAL)

Функция **ВЫЧИСЛИТЬ** предназначена для вычисления выражения в контексте некоторой группировки. Функция используется для совместимости с предыдущими версиями платформы. Вместо неё рекомендуется использовать функцию **ВычислитьВыражение**.

Функция имеет следующие параметры:

- **Выражение** – строка, содержащая вычисляемое выражение.
- **Группировка** – строка, содержащая имя группировки, в контексте которой необходимо вычислить выражение. Если в качестве имени группировки используется пустая строка, вычисление будет выполнено в контексте текущей группировки. Если в качестве имени группировки будет использована строка **ОбщийИтог**, вычисление будет выполнено в контексте общего итога. В остальных случаях вычисление будет выполняться в контексте родительской группировки с таким именем.
- **Тип расчета** – строка, содержащая тип расчета. Если данный параметр имеет значение **ОбщийИтог**, выражение будет вычисляться для всех записей группировки. Если значение параметра – **Группировка**, значения будут вычисляться для текущей групповой записи группировки.

[Копировать в буфер обмена](#)

```
СУММА(Продажи.СуммаОборот) / ВЫЧИСЛИТЬ("Сумма(Продажи.СуммаОборот)", "ОбщийИтог")
```

В данном примере в результате получится отношение суммы по полю `Продажи.СуммаОборот` записи группировки к сумме того же поля во всей компоновке.

ВЫЧИСЛИТЬВЫРАЖЕНИЕ (EVALEXPRESSION)

Функция возвращает выражение для записи указанной группировки. Имеет следующие параметры:

- **Вычисляемое выражение**. Строка, содержащая выражение, которое требуется вычислить.
- **Группировка**. Строка, содержащая имя группировки, в которой требуется вычислить выражение. Если в качестве имени группировки используется пустая строка, то вычисление будет выполнено для текущей группировки. Если указана строка **ОбщийИтог**, то вычисление будет выполнено для общего итога. Если в параметре указано имя группировки-строки или указана пустая строка, то при вычислении в таблице вычисление будет выполнено для группировки, полученной на пересечении требуемой области вычисления (см. описание третьего параметра) и текущей строки группировки колонки. Если в параметре указано имя группировки-колонки, то вычисление выполняется для группировки, полученной на пересечении области вычисления группировки-колонки и текущей записи группировки-строки.
- **Область вычисления**. Строка, содержащая область вычисления. Если в качестве области вычисления используется пустая строка, то вычисление будет выполнено для текущей записи группировки, указанной в первом параметре. Если параметр содержит строку **ОбщийИтог**, то вычисление будет выполнено для общего итога по группировке, указанной во втором параметре.

Если значение параметра **Группировка**, значения будут вычисляться для текущей групповой записи группировки.

Если параметр имеет значение **ГруппировкаНеРесурса**, то при вычислении функции для групповой записи по ресурсам выражение будет вычислено для первой групповой записи исходной группировки.

При вычислении функции **ВычислитьВыражение** со значением **ГруппировкаНеРесурса** для групповых записей, не являющихся группировками по ресурсам, функция вычисляется так же, как бы она вычислялась со значением параметра **Группировка**.

Компоновщик макета компоновки данных при генерации макета компоновки данных, при выводе в макет поля-ресурса, по которому выполняется группировка, выдает в макет выражение, вычисляемое при помощи функции **ВычислитьВыражение** с указанным параметром **ГруппировкаНеРесурса**. Для остальных ресурсов в группировке по ресурсу выдаются обычные выражения ресурсов.

Если параметр содержит строку **Иерархия**, то выражение будет вычислено для родительской иерархической записи, если таковая имеется, и для всей группировки, если родительской иерархической записи нет. Компоновщик макета при генерации выражения для поля **% в группе иерархии** генерирует выражение, содержащее отношение выражения ресурса к функции **ВычислитьВыражение** для выражения ресурса, вычисляемого для текущей группировки с типом вычисления **Иерархия**.

• **Начало**. Указывает, с какой записи нужно начинать фрагмент, в котором рассчитывать агрегатные функции выражения и из какой записи получать значения полей вне агрегатных функций. Строка, содержащая одно из значений:

- **Первая (First)**. Необходимо получать первую запись группировки. После слова в скобках можно указывать выражение, результат которого будет использоваться как смещение от начала группировки. Получаемое значение должно быть целым числом больше нуля. Например, **Первая(3)** – получение третьей записи от начала группировки. Если первая запись выходит за пределы группировки, то считается, что записей нет. Например, если записей 3, а требуется получить **Первая(4)**, то считается, что записей нет.

- **Последняя (Last)**. Необходимо получить последнюю запись группировки. После слова в скобках можно указывать выражение, результат которого будет использоваться как смещение от конца группировки. Получаемое значение должно быть целым числом больше нуля. Например, [Последняя\(3\)](#) – получение третьей записи от конца группировки.

Если последняя запись выходит за пределы группировки, то считается, что записей нет. Например, если записей 3, а требуется получить [Последняя\(4\)](#), то считается, что записей нет.

- **Предыдущая (Previous)**. Необходимо получить предыдущую запись группировки. После слова в скобках можно указывать выражение, результат которого будет использоваться как смещение назад от текущей записи группировки. Например, [Предыдущая\(2\)](#) – получение предыдущей от предыдущей записи.

Если предыдущая запись выходит за пределы группировки (например, для второй записи группировки требуется получить [Предыдущая\(3\)](#)), то получается первая запись группировки.

При получении предыдущей записи для итога по группировке получается первая запись.

- **Следующая (Next)**. Необходимо получить следующую запись группировки. После слова в скобках можно указывать выражение, результат которого будет использоваться как смещение вперед от текущей записи группировки. Например, [Следующая\(2\)](#) – получение следующей от следующей записи.

Если следующая запись выходит за пределы группировки, то считается, что записей нет. Например, если записей 3 и для третьей записи получают [Следующая](#), то считается, что записей нет.

При получении следующей записи для итога по группировке считается, что записи нет.

- **Текущая (Current)**. Необходимо получить текущую запись.

При получении для итога по группировке получается первая запись.

- **ОграничивающееЗначение (BoundaryValue)**. Необходимость получить запись по указанному значению. После слова [ОграничивающееЗначение](#) в скобках нужно указать выражение, со значения которого нужно начинать фрагмент, первого поля упорядочивания.

В качестве записи будет получаться первая запись, значение поля упорядочивания у которой больше или равно указанному значению. Например, если в качестве поля упорядочивания используется поле [Период](#), оно имеет значения 01.01.2010, 01.02.2010, 01.03.2010 и требуется получить [ОграничивающееЗначение\(ДатаВремя\(2010, 1, 15\)\)](#), то будет получена запись с датой 01.02.2010.

- **Конец**. Указывает, до какой записи нужно продолжать фрагмент, в котором рассчитывать агрегатное выражение. Строка, содержащая одно из значений:

- **Первая (First)**. Необходимо получить первую запись группировки. После слова в скобках можно указывать выражение, результат которого будет использоваться как смещение от начала группировки. Получаемое значение должно быть целым числом больше нуля. Например, [Первая\(3\)](#) – получение третьей записи от начала группировки.

Если первая запись выходит за пределы группировки, то считается, что записей нет. Например, если записей 3, а требуется получить [Первая\(4\)](#), то считается, что записей нет.

- **Последняя (Last)**. Необходимо получить последнюю запись группировки. После слова в скобках можно указывать выражение, результат которого будет использоваться как смещение от конца группировки. Получаемое значение должно быть целым числом больше нуля. Например, [Последняя\(3\)](#) – получение третьей записи от конца группировки.

Если последняя запись выходит за пределы группировки, то считается, что записей нет. Например, если записей 3, а требуется получить [Последняя\(4\)](#), то считается, что записей нет.

- **Предыдущая (Previous)**. Необходимо получить предыдущую запись группировки. После слова в скобках можно указывать выражение, результат которого будет использоваться как смещение назад от текущей записи группировки. Например, [Предыдущая\(2\)](#) – получение предыдущей от предыдущей записи.

Если предыдущая запись выходит за пределы группировки (например, для второй записи группировки требуется получить [Предыдущая\(3\)](#)), то считается, что записей нет.

При получении предыдущей записи для итога по группировке считается последняя запись.

- **Следующая (Next)**. Необходимо получить следующую запись группировки. После слова в скобках можно указывать выражение, результат которого будет использоваться как смещение вперед от текущей записи группировки. Например, [Следующая\(2\)](#) – получение следующей от следующей записи.

Если следующая запись выходит за пределы группировки, получается последняя запись. Например, если записей 3 и для третьей записи получают [Следующая](#), то получается третья запись.

При получении следующей записи для итога по группировке считается, что записи нет.

- **Текущая (Current)**. Необходимо получить текущую запись.

При получении для итога по группировке получается первая запись.

- **ОграничивающееЗначение (BoundaryValue)**. Необходимость получить запись по указанному значению. После слова [ОграничивающееЗначение](#) в скобках следует указать выражение, со значения которого нужно начинать фрагмент, первого поля упорядочивания.

В качестве записи будет получаться последняя запись, значение поля упорядочивания у которой меньше или равно указанному значению. Например, если в качестве поля упорядочивания используется поле [Период](#), оно имеет значения 01.01.2010, 01.02.2010, 01.03.2010, требуется получить [ОграничивающееЗначение\(ДатаВремя\(2010, 1, 15\)\)](#), то будет получена запись с датой 01.01.2010.

- **Сортировка**. Строка, в которой перечисляются выражения, разделенные запятыми, в направлении которых нужно упорядочивать последовательность. Если не указана, то упорядочивание выполняется так же, как и у группировки, для которой вычисляется выражение. После каждого выражения можно указать ключевое слово [Возр](#) для упорядочивания по возрастанию, [Убыв](#) для упорядочивания по убыванию, [Автоупорядочивание](#) для упорядочивания ссылочных полей по полям, по которым нужно упорядочивать объект, на который выполняется ссылка. Слово [Автоупорядочивание](#) может использоваться как со словом [Возр](#), так и со словом [Убыв](#).

- **ИерархическаяСортировка**. Строка, в которой перечисляются выражения, разделенные запятыми, в направлении которых нужно упорядочивать последовательность. Применяется для упорядочивания иерархических записей. Если не указано, то упорядочивание будет выполнено в соответствии с упорядочиванием, указанным в параметре [Сортировка](#).

- **ОбработкаОдинаковыхЗначенийПорядка**. Строка, содержащая одно из значений:

- **Вместе (Together)** обозначает, что предыдущая и следующие записи определяются на основании значений выражений упорядочивания.

- **Отдельно (Separately)** обозначает, что для определения предыдущей и следующей записей используется последовательность упорядоченных записей.

Например, если полученная последовательность упорядочена по дате:

2. 02 января 2001 Петров С. 20

[Копировать в буфер обмена](#)

3. 02 января 2001 Сидоров Р. 30

[Копировать в буфер обмена](#)

4. 03 января 2001 Петров С. 40

При использовании в обработке одинаковых значений порядка **Отдельно** предыдущей к записи 3 будет запись 2, а при использовании **Вместе** – запись 1. А фрагментом для текущей записи к записи 2 для **Отдельно** будет запись 2, а для **Вместе** – записи 2 и 3. Таким образом, сумма по текущей записи для **Отдельно** составит 20, а для **Вместе** – 50.

При указанном **Вместе** в параметрах **Начало** и **Конец** нельзя указывать смещение для позиций **Первая**, **Последняя**, **Предыдущая**, **Следующая**.

По умолчанию **Отдельно**.

Пример использования параметров

Если требуется рассчитать сумму с накоплением, то можно использовать следующее выражение:

[Копировать в буфер обмена](#)

```
ВычислитьВыражение("Сумма(СуммаОборот)", , , "Первая", "Текущая")
```

Если требуется получить значение группировки в предыдущей строке, то можно использовать следующее выражение:

[Копировать в буфер обмена](#)

```
ВычислитьВыражение("Курс", , , "Предыдущая")
```

Функция **ВычислитьВыражение** учитывает отбор группировок, но не учитывает иерархические отборы.

Функция **ВычислитьВыражение** не может применяться к группировке в групповом отборе этой группировки. Например, в отборе группировки **Номенклатура** нельзя использовать выражение **ВычислитьВыражение("Сумма(СуммаОборот)", , , "ОбщийИтог") > 1000**. Но такое выражение можно использовать в иерархическом отборе.

Если конечная запись предшествует начальной, то считается, что записи для расчета детальных данных и расчета агрегатных функций отсутствуют.

При расчете интервальных выражений для общего итога (второй параметр функции **ВычислитьВыражение** имеет значение **"ОбщийИтог"**) считается, что записи для расчета детальных данных и расчета агрегатных функций отсутствуют.

УРОВЕНЬ (LEVEL)

Функция предназначена для получения текущего уровня записи.

[Копировать в буфер обмена](#)

```
УРОВЕНЬ()
```

УРОВЕНЬВГРУППИРОВКЕ (LEVELINGROUP)

Функция предназначена для получения уровня записи относительно корня группировки.

[Копировать в буфер обмена](#)

```
УРОВЕНЬВГРУППИРОВКЕ()
```

ЗНАЧЕНИЕЗАПОЛНЕНО (VALUEISFILLED)

Возвращает **Истину**, если значение отлично от значения данного типа по умолчанию, отлично от значения **NULL**, отлично от пустой ссылки, отлично от значения **Неопределено**. Для логических значений осуществляется проверка на значение **NULL**. Для строк осуществляется проверка на отсутствие пробельных символов.

НОМЕРПОПОРЯДКУ (SERIALNUMBER)

Получить следующий порядковый номер.

[Копировать в буфер обмена](#)

```
НОМЕРПОПОРЯДКУ()
```

НОМЕРПОПОРЯДКУВГРУППИРОВКЕ (GROUPSERIALNUMBER)

Возвращает следующий порядковый номер в текущей группировке.

[Копировать в буфер обмена](#)

```
НОМЕРПОПОРЯДКУВГРУППИРОВКЕ()
```

ФОРМАТ (FORMAT)

Получить отформатированную строку переданного значения. Форматная строка задается в соответствии с форматной строкой «1С:Предприятия».

Параметры:

- **Значение**,
- **Форматная строка**.

[Копировать в буфер обмена](#)

```
ФОРМАТ(РасходныеНакладные.СуммаДок, "ЧДЦ=2")
```

НАЧАЛОПЕРИОДА (BEGINPERIOD)

Функция предназначена для выделения определенной даты из заданной даты.

Параметры:

- **Выражение** типа **Дата**;
- **Тип периода** – строка, содержащая одно из значений:

698

- Минута,
- Час,
- День,
- Неделя,
- Месяц,
- Квартал,
- Год,
- Декада,
- Полугодие.

[Копировать в буфер обмена](#)

`НАЧАЛОПЕРИОДА(ДАТАВРЕМЯ(2009, 10, 12, 10, 15, 34), "Месяц")`

Результат:

[Копировать в буфер обмена](#)

01.10.2009 0:00:00

КОНЕЦПЕРИОДА (ENDOFPERIOD)

Функция предназначена для выделения определенной даты из заданной даты.

Параметры:

- Выражение типа `Дата`;
- Тип периода – строка, содержащая одно из значений:
 - Минута,
 - Час,
 - День,
 - Неделя,
 - Месяц,
 - Квартал,
 - Год,
 - Декада,
 - Полугодие.

[Копировать в буфер обмена](#)

`КОНЕЦПЕРИОДА(ДАТАВРЕМЯ(2009, 10, 12, 10, 15, 34), "Неделя")`

Результат:

[Копировать в буфер обмена](#)

13.10.2009 23:59:59.

ДОБАВИТЬКДАТЕ (DATEADD)

Функция предназначена для прибавления к дате некоторой величины. Параметры:

- Выражение типа `Дата`;
- Тип увеличения – строка, содержащая одно из значений:
 - Секунда,
 - Минута,
 - Час,
 - День,
 - Неделя,
 - Месяц,
 - Квартал,
 - Год,
 - Декада,
 - Полугодие.
- Величина – на сколько необходимо увеличить дату. Тип `Число`. Дробная часть игнорируется.

[Копировать в буфер обмена](#)

`ДОБАВИТЬКДАТЕ(ДАТАВРЕМЯ(2009, 10, 12, 10, 15, 34), "Месяц", 1)`

Результат:

12.11.2009 10:15:34

[Копировать в буфер обмена](#)

РАЗНОСТЬДАТ (DATEDIFF)

Функция предназначена для получения разницы между двумя датами. Параметры:

- Выражение типа *Дата*;
- Выражение типа *Дата*;
- Тип разности – одно из значений:
 - Секунда,
 - Минута,
 - Час,
 - День,
 - Месяц,
 - Квартал,
 - Год.

```
РАЗНОСТЬДАТ(ДАТАВРЕМЯ(2009, 10, 12, 10, 15, 34),
```

[Копировать в буфер обмена](#)

```
ДАТАВРЕМЯ(2009, 10, 14, 9, 18, 06),
```

[Копировать в буфер обмена](#)

```
"День")
```

[Копировать в буфер обмена](#)

Результат:

2

[Копировать в буфер обмена](#)

ТЕКУЩАЯДАТА (CURRENTDATE)

Возвращает системную дату. При компоновке макета компоновки во всех выражениях, которые присутствуют в компоновке, функция **ТЕКУЩАЯДАТА** заменяется значением текущей даты.

```
ТЕКУЩАЯДАТА()
```

[Копировать в буфер обмена](#)

ПОДСТРОКА (SUBSTRING)

Данная функция предназначена для выделения подстроки из строки. Параметры:

- выражение, имеющее строковый тип;
- позиция символа, с которого начинается выделяемая из строки подстрока;
- длина выделяемой подстроки.

```
ПОДСТРОКА(Контрагенты.Адрес, 1, 4)
```

[Копировать в буфер обмена](#)

ДЛИНАСТРОКИ (STRINGLENGTH)

Функция предназначена для определения длины строки. Параметр – выражение строкового типа.

```
ДЛИНАСТРОКИ(Контрагенты.Адрес)
```

[Копировать в буфер обмена](#)

ГОД (YEAR)

Данная функция предназначена для выделения года из значения типа *Дата*. Единственный параметр – это выражение, имеющее тип *Дата*.

```
ГОД(РасхНакл.Дата)
```

[Копировать в буфер обмена](#)

КВАРТАЛ (QUARTER)

Данная функция предназначена для выделения номера квартала из значения типа *Дата*. Номер квартала в норме находится в диапазоне от 1 до 4. Единственный параметр функции – это выражение, имеющее тип *Дата*.

```
КВАРТАЛ(РасхНакл.Дата)
```

[Копировать в буфер обмена](#)

МЕСЯЦ (MONTH)

Данная функция предназначена для выделения номера месяца из значения типа *Дата*. Номер месяца в норме находится в диапазоне от 1 до 12. Единственный параметр функции – это выражение, имеющее тип *Дата*.

```
МЕСЯЦ(РасхНакл.Дата)
```

[Копировать в буфер обмена](#)

ДЕНЬГОДА (DAYOFYEAR)

Данная функция предназначена для получения дня года из значения типа `Дата`. День года в норме находится в диапазоне от 1 до 365 (366). Единственный параметр функции – это выражение, имеющее тип `Дата`.

[Копировать в буфер обмена](#)

`ДЕНЬГОДА(РасхНакл.Дата)`

ДЕНЬ (DAY)

Данная функция предназначена для получения дня месяца из значения типа `Дата`. День месяца в норме находится в диапазоне от 1 до 31. Единственный параметр функции – это выражение, имеющее тип `Дата`.

[Копировать в буфер обмена](#)

`ДЕНЬ(РасхНакл.Дата)`

НЕДЕЛЯ (WEEK)

Данная функция предназначена для получения номера недели года из значения типа `Дата`. Недели года нумеруются, начиная с 1. Единственный параметр функции – это выражение, имеющее тип `Дата`.

[Копировать в буфер обмена](#)

`НЕДЕЛЯ(РасхНакл.Дата)`

ДЕНЬНЕДЕЛИ (WEEKDAY)

Данная функция предназначена для получения дня недели из значения типа `Дата`. День недели в норме находится в диапазоне от 1 (понедельник) до 7 (воскресенье). Единственный параметр функции – это выражение, имеющее тип `Дата`.

[Копировать в буфер обмена](#)

`ДЕНЬНЕДЕЛИ(РасхНакл.Дата)`

ЧАС (HOUR)

Данная функция предназначена для получения часа суток из значения типа `Дата`. Час суток находится в диапазоне от 0 до 23. Единственный параметр функции – это выражение, имеющее тип `Дата`.

[Копировать в буфер обмена](#)

`ЧАС(РасхНакл.Дата)`

МИНУТА (MINUTE)

Данная функция предназначена для получения минуты часа из значения типа `Дата`. Минута часа находится в диапазоне от 0 до 59. Единственный параметр функции – это выражение, имеющее тип `Дата`.

[Копировать в буфер обмена](#)

`МИНУТА(РасхНакл.Дата)`

СЕКUNДА (SECOND)

Данная функция предназначена для получения секунды минуты из значения типа `Дата`. Секунда минуты находится в диапазоне от 0 до 59. Единственный параметр функции – это выражение, имеющее тип `Дата`.

[Копировать в буфер обмена](#)

`СЕКUNДА(РасхНакл.Дата)`

ВЫРАЗИТЬ (CAST)

Данная функция предназначена для выделения типа из выражения, которое может содержать составной тип. Если выражение будет содержать тип, отличный от требуемого, будет возвращено значение `NULL`.

Параметры:

- Преобразуемое выражение;
- **Тип** – строка, содержащая строку типа. Например, `Число`, `Строка` и т. п. Кроме примитивных типов данная строка может содержать имя таблицы. В таком случае будет осуществлена попытка выразить к ссылке на указанную таблицу.

[Копировать в буфер обмена](#)

`ВЫРАЗИТЬ(Данные.Реквизит1, "Число(10,3)")`

ЕСТЬNULL (ISNULL)

Данная функция возвращает значение второго параметра, если значение первого параметра – `NULL`. В противном случае будет возвращено значение первого параметра.

[Копировать в буфер обмена](#)

`ЕСТЬNULL(Сумма(Продажи.СуммаОборот), 0)`

Cos

Вычисляет косинус.

Синтаксис:

`Cos(Выражение)`.

Параметр:

Выражение – тип `Число`. Задается в радианах.

ACos

Вычисляет арккосинус в радианах.

Синтаксис:

ASos (Выражение)

Параметр:

Выражение – тип **Число**. Значение косинуса (в диапазоне -1 ... 1), по которому определяется угол.

Sin

Вычисляет синус.

Синтаксис:

Sin(Выражение)

Параметр:

Выражение – тип **Число**. Задается в радианах.

ASin

Вычисляет арксинус в радианах.

Синтаксис:

ASin(Выражение)

Параметр:

Выражение – тип **Число**. Значение синуса (в диапазоне -1 ... 1), по которому определяется угол.

Tan

Вычисляет тангенс.

Синтаксис:

Tan(Выражение)

Параметр:

Выражение – тип **Число**. Значение синуса, по которому определяется угол.

ATan

Вычисляет арктангенс в радианах.

Синтаксис:

ATan(Выражение)

Параметр:

Выражение – тип **Число**. Значение тангенса, по которому определяется угол.

Exp

Возведение числа e в степень.

Синтаксис:

Exp(Выражение)

Параметр:

Выражение – тип **Число**. Значение степени.

Log

Вычисляет натуральный логарифм.

Синтаксис:

Log(Выражение)

Параметр:

Выражение – тип **Число**. Исходное число, больше 0.

Log10

Вычисляет логарифм X по основанию 10.

Синтаксис:

Log10(Выражение)

Параметр:

Выражение – тип **Число**. Исходное число, больше 0.

Pow

Возведение в степень.

Синтаксис:

`Pow(Основание, Показатель)`

Параметры:

- **Основание** – тип `Число`. Основание операции возведения в степень.
- **Показатель** – тип `Число`. Показатель степени.

Sqrt

Вычисляет квадратный корень.

Синтаксис:

`Sqrt(Выражение)`

Параметр:

Выражение – тип `Число`. Неотрицательное число.

Окр (Round)

Округляет исходное число до требуемой разрядности. Режим округления стандартный (1,5 как 2).

Синтаксис:

`Окр(Выражение, Разрядность)`

Параметры:

Выражение – тип `Число`. Исходное число. **Разрядность** – тип `Число`. Число знаков дробной части, до которых производится округление.

Цел (Int)

Отсекает дробную часть числа.

Синтаксис:

`Цел(Выражение)`

Параметр:

Выражение – тип `Число`. Дробное число.

Приложение 2. Текстовый редактор

Текстовый редактор системы «1С:Предприятие» предоставляет пользователю все основные функции, необходимые при редактировании текстов. При работе с редактором доступны операции с блоками текста, функции поиска и замены. Так как работа с любым текстовым редактором осуществляется примерно одинаковым образом, в данном приложении будет дано краткое описание возможностей редактора текстов системы «1С:Предприятие».

Процесс редактирования текстового документа включает следующие этапы:

- создание нового документа или открытие одного из существующих документов;
- ввод и редактирование текста;
- сохранение отредактированного текста;
- печать, если это необходимо.

Ниже будут подробно рассмотрены все эти этапы.

2.1. Создание и открытие текстового документа

Для создания нового текстового документа используется пункт меню **Файл – Новый**. В появившемся запросе следует выбрать строку **Текстовый документ** и нажать кнопку **ОК**. Будет открыто окно текстового редактора, содержащее пустой документ.

Чтобы открыть существующий текстовый документ, используется пункт меню **Файл – Открыть**. На экран будет выдан стандартный диалог открытия файла.

В этом диалоге следует выбрать файл документа. В поле **Тип файла** можно выбрать формат открываемого текстового документа.

ВНИМАНИЕ! Если открыт документ с атрибутом **Только для чтения**, то такой документ не редактируется. Если открыть палитру свойств (пункт **Правка – Свойства**), то в строке **Статус** свойств документа будет указано **Текст открыт для чтения**. Если атрибут документа **Только для чтения** не установлен, то в строке **Статус** свойств документа указывается **Текст открыт для записи**.

Можно создать или открыть одновременно несколько документов в различных окнах.

2.2. Выбор расширения текстового документа

В пункте главного меню **Текст** расположены команды для указания расширения текстового документа.

Возможные значения: **Текст**, **Текстовый макет**, **HTML**. В зависимости от выбранного расширения будет автоматически выбираться режим текстового редактора для данного поля.

По умолчанию (при создании нового документа) документу устанавливается расширение **Текст**.

Функциональные возможности редактирования текстов модулей описаны в книге «1С:Предприятие 8.3. Руководство разработчика».

2.3. Ввод и редактирование текста

Ввод текста осуществляется путем набора нужных символов на клавиатуре. Набираемые символы будут вводиться в месте нахождения курсора – тонкой вертикальной полоски. В процессе ввода и редактирования текста текущее положение курсора – номер строки и номер позиции в строке – выдается в панели состояния программы.

Размер редактируемого документа (число символов и строк) и режим редактирования можно посмотреть, если выбрать пункт **Правка – Свойства**.

2.3.1. Перемещение курсора

Перемещение курсора по редактируемому тексту осуществляется стандартным образом.

2.3.2. Переход к строке текста

Для перехода к конкретной строке документа необходимо в контекстном меню выбрать пункт [Переход к строке](#), а затем в выданном на экран запросе ввести номер строки и нажать кнопку [Перейти](#).

2.3.3. Использование закладок

В процессе редактирования любые строки текста можно отметить, а затем быстро перемещаться между ними. Такие отметки называются **закладками**.

Для установки закладки необходимо установить курсор в строку текста, которую необходимо пометить, и использовать пункт меню [Текст – Установить/снять закладку](#). Установленная закладка изображается кружком голубого цвета в крайней левой колонке окна редактора текстов.

Закладку можно убрать повторным выбором пункта [Текст – Установить/снять закладку](#), когда курсор находится в отмеченной строке.

Для перемещения между отмеченными строками текста используются пункты меню [Текст – Следующая закладка](#), [Текст – Предыдущая закладка](#). Переход по закладкам закольцован: при достижении конца или начала (в зависимости от направления поиска) следующая команда поиска перемещает курсор на первую или последнюю закладку.

Команда [Текст – Убрать все закладки](#) убирает все закладки из редактируемого текста. Закладки автоматически убираются, когда редактируемый документ закрывается.

Установленные закладки при закрытии документа не сохраняются.

2.3.4. Выделение блока текста

Выделение блока текста осуществляется стандартным образом.

Чтобы выделить сразу весь текст документа, нужно выбрать пункт [Правка – Выделить все](#).

Чтобы выделить только одно слово, используется сочетание клавиш [Ctrl + W](#).

2.3.5. Операции с выделенным блоком

Выделенный блок текста может быть перемещен, скопирован или удален стандартным для Microsoft Windows образом.

Для форматирования структурного вида (отступов) текста удобно пользоваться сдвигом блока. Нужно выделить одну или более строк текста (если выделяется одна строка, то обязательно следует выделить конец строки; если несколько, то можно выделять строки частично), выбрать пункт [Текст – Блок – Увеличить отступ](#) или [Текст – Блок – Уменьшить отступ](#). При увеличении отступа редактор сдвигает в указанном направлении выбранные строки на величину, равную размеру табуляции. При уменьшении отступа из начала каждой строки удаляется необходимое количество пробелов и символов табуляции.

2.3.6. Удаление текста

Удаление текста осуществляется стандартным образом.

2.3.7. Вставка конца страницы

Для вставки конца страницы следует выбрать пункт [Текст – Конец страницы](#). Переход к новой странице будет обозначен знаком [§](#). При печати в этом месте текста будет начата новая страница.

2.3.8. Отмена сделанных изменений

Если в процессе редактирования допущена ошибка, текстовый редактор позволяет автоматически вернуть первоначальное состояние редактируемого текста. Для отмены действия необходимо выбрать пункт [Правка – Отменить](#).

После этого результаты последнего действия будут отменены. Выполнять операцию отмены сделанных изменений можно несколько раз подряд.

Чтобы восстановить отмененные действия, следует выбрать пункт [Правка – Вернуть](#).

2.3.9. Поиск и замена

Текстовый редактор системы «1С:Предприятие» позволяет найти в текстовом документе любую последовательность символов и заменить ее другой.

Поиск. Для осуществления поиска необходимо выбрать пункт [Правка – Найти](#). На экран будет выдан диалог для установки параметров поиска.

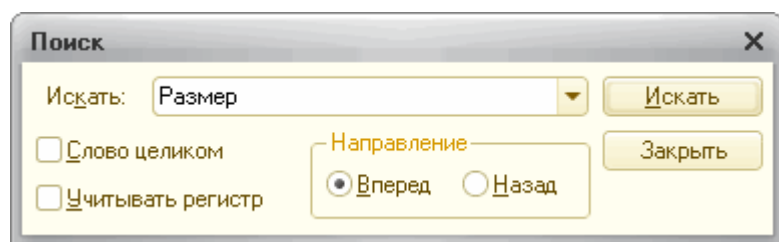


Рис. 239. Диалог установки параметров поиска

В поле [Искать](#) этого диалога следует ввести образец для поиска.

Чтобы различать при поиске прописные и строчные буквы, необходимо включить опцию [Учитывать регистр](#). При установленной опции [Слово целиком](#) будут найдены только целые слова, а не части слов.

Группа [Направление](#) позволяет указать направление поиска: [Вперед](#) – вправо и вниз от текущей позиции курсора; [Назад](#) – влево и вверх от текущей позиции курсора.

После установки необходимых параметров для выполнения поиска следует нажать кнопку [Искать](#). Если указанная последовательность символов будет найдена, она будет выделена.

Независимо от ранее заданного общего направления можно повторить поиск, задав ему направление [Вперед](#), если выбрать пункт [Правка – Найти следующий](#) или нажать клавишу **F3**.

Независимо от ранее заданного общего направления можно повторить поиск, задав ему направление [Назад](#), если выбрать пункт [Правка – Поиск – Найти предыдущий](#) или нажать клавиши **Shift + F3**.

ВНИМАНИЕ! Когда поиск достигает конца документа, он продолжается с начала, и наоборот, при достижении начала документа поиск продолжается с конца.

Замена. Для выполнения замены символьной строки необходимо выбрать пункт [Правка – Заменить](#). На экран будет выдан диалог для установки параметров замены.

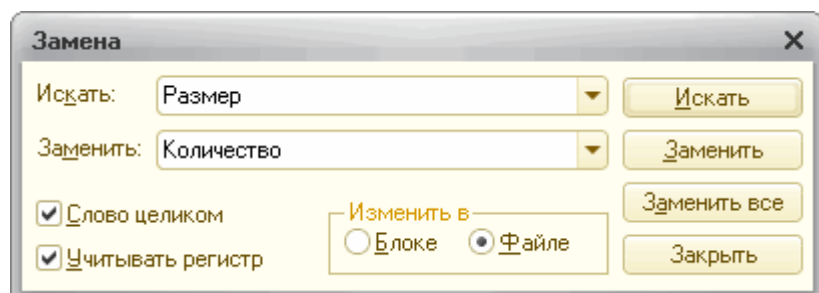


Рис. 240. Диалог установки параметров замены

В поле [Искать](#) этого диалога следует ввести образец для поиска. В поле [Заменить](#) необходимо указать последовательность символов, на которую будет заменяться образец поиска, указанный в поле [Искать](#).

Чтобы различать при поиске прописные и строчные буквы, нужно включить опцию [Учитывать регистр](#). При установленной опции [Слово целиком](#) будут найдены только целые слова, а не части слов.

Группа [Изменить в](#) позволяет указать диапазон поиска – в выделенном тексте (блоке) или во всем документе.

После установки необходимых параметров для начала процесса поиска и замены следует нажать кнопку [Искать](#).

Если искомый образец будет найден, он будет выделен в тексте. Для замены найденной последовательности символов следует нажать кнопку [Заменить](#). Будет произведена ее замена, и поиск будет продолжен.

Чтобы заменить сразу все вхождения искомой строки, нужно нажать кнопку [Заменить все](#).

ВНИМАНИЕ! Когда замена достигает конца документа, она продолжается с начала, и наоборот, при достижении начала документа замена продолжается с конца.

2.3.10. Сохранение текстового документа

Для сохранения текстового документа используется пункт [Файл – Сохранить](#). Если документ сохраняется в первый раз, на экран будет выдан стандартный диалог для сохранения файла, в котором необходимо указать имя нового документа и, если необходимо, каталог, в который он будет помещен.

Так же, как и при открытии документа, при сохранении в поле [Тип файла](#) можно выбрать формат сохраняемого файла.

Для сохранения существующего текстового документа под другим именем используется пункт [Файл – Сохранить как](#). На экран будет выдан стандартный диалог для сохранения файла, в котором необходимо указать новое имя документа и, если необходимо, каталог, в который он будет помещен. При этом, если в текстовом документе установлена одна из четырех стандартных кодировок ([UTF-8](#), [UTF-16](#), [ANSI](#), [OEM](#)), она будет использоваться в качестве кодировки по умолчанию в диалоге сохранения файла. Если документ имеет другую кодировку, то будет использована кодировка [UTF-8](#). После сохранения документа под новым именем файл со старым именем остается на диске.

2.3.11. Печать текстового документа

Настройка параметров страницы. Режим настройки параметров страницы позволяет выбрать печатающее устройство и установить размер и ориентацию бумаги, на которой будет производиться печать документа. Для вызова этого режима следует выбрать пункт [Файл – Параметры страницы](#). На экран будет выдан диалог [Параметры страницы](#).

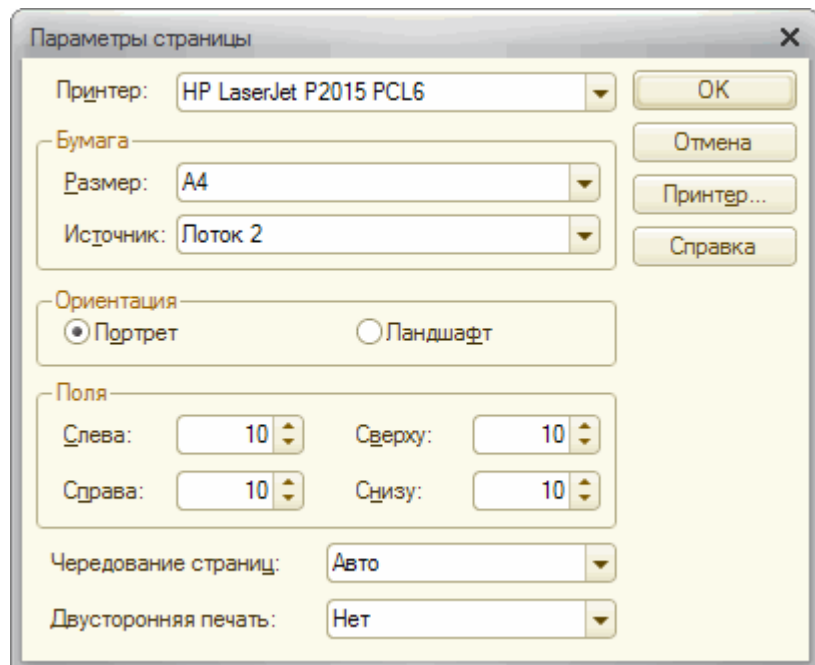


Рис. 241. Диалог настройки параметров страницы

Поле **Принтер** позволяет выбрать для печати один из принтеров, подключенных к вашему компьютеру. Нажатием кнопки **Стрелка Вниз** можно открыть список таких принтеров и выбрать требуемый принтер, щелкнув мышью по его названию в списке. Аналогичным образом выбирается размер бумаги и лоток принтера, откуда будет подаваться бумага при печати.

Группа **Поля** позволяет задать расстояние между краем листа и печатаемым текстом. Текст, который не помещается в отведенных границах по ширине, будет вынесен на соседнюю страницу.

Чередование страниц. Поле **Чередование страниц** позволяет установить при печати чередование четных и нечетных страниц:

- **Не использовать** – поля печатаются как обычно;
- **Зеркально слева** – нечетные страницы печатаются как обычно, а четные печатаются с переменной местами левого и правого поля границ;
- **Зеркально сверху** – нечетные страницы печатаются как обычно, а четные страницы печатаются с переменной местами нижнего и верхнего полей границ;
- **Авто** – если в поле **Двусторонняя печать** указано **Нет**, поля выводятся как обычно; если **Переворот влево**, то используется вывод со значением **Зеркально слева**; если **Переворот вверх**, то используется вывод со значением **Зеркально сверху**.

Двусторонняя печать. Поле позволяет настроить двустороннюю печать страниц:

- **Нет** – двусторонняя печать не используется;
- **Переворот влево** – двусторонняя печать, при которой страницу нужно переворачивать влево;
- **Переворот вверх** – двусторонняя печать, при которой страницу нужно переворачивать вверх.

После установки всех необходимых параметров для их сохранения следует нажать кнопку **ОК**. Нажав кнопку **Отмена**, можно отказаться от изменения параметров страницы.

Просмотр документа перед печатью. Вы можете перед печатью просмотреть документ в том виде, в каком он будет напечатан. Для этого следует выбрать пункт **Файл – Предварительный просмотр**.

Печать документа. Распечатать документ можно с помощью стандартного диалога для настройки параметров печати или без диалога, используя стандартные настройки принтера (на текущий принтер). Для первого случая используется первая строка **Печать**. На экран будет выдан стандартный диалог для

настройки параметров печати.

В этом диалоге следует установить необходимые параметры печати – выбрать принтер, диапазон страниц для печати, количество копий – и нажать кнопку **OK** для выполнения печати.

В случае использования текущих параметров печати нужно выбрать **Файл – Печать с текущими настройками (Ctrl + Shift + P)**.

Таблицу сочетаний клавиш для редактора текстов см. в справке при использовании программы.

2.3.12. Закрытие текстового документа

Закрытие текстового документа производится стандартным образом.

В связи с тем, что текстовые документы в основном используются для представления информации пользователю, для закрытия документа можно использовать клавишу **Esc**. Если текстовый документ не был модифицирован, то он закрывается, в противном случае на экран выводится запрос о необходимости сохранения документа.

Приложение 3. Редактор табличных документов

Для создания различных печатных форм в системе «1С:Предприятие» используется специализированный табличный редактор.

В данном приложении будут изложены принципы работы с табличным документом, необходимые для работы пользователей.

3.1. Табличный документ в системе «1С:Предприятие»

Табличный документ в системе «1С:Предприятие», как и любая другая таблица, представляет собой совокупность ячеек прямоугольной формы, организованных в строки и колонки.

Каждая колонка и каждая строка имеет свой уникальный номер. Строки и колонки нумеруются независимо, нумерация начинается с 1 и ведется от левого верхнего угла табличного документа. Таким образом, любая ячейка может быть обозначена парой чисел – номером строки и номером колонки, на пересечении которых она находится.

Кроме того, отдельным ячейкам и их диапазонам (компактным группам прямоугольной формы) можно присваивать собственные имена для удобства обращения к ним. Для присваивания имен и работы с ними используется окно [Имена](#) и палитра свойств.

Самая верхняя ячейка колонки называется заголовком колонки. В ней выводится номер колонки, кроме того, она используется для выделения колонки и изменения ее ширины. Аналогичное назначение имеет и самая левая ячейка строки, которая называется заголовком строки.

Табличный документ формируется программой (отчеты, обработки) или пользователем. Он может храниться (если требуется) только в файле на диске. Обычно он представляет «готовую» печатную форму и используется самостоятельно.

3.2. Работа с табличным документом

Редактор табличных документов системы «1С:Предприятие» может использоваться для создания, просмотра и редактирования документов табличного типа.

Отчеты обычно формируют результаты обработки данных в виде табличных документов.

Кроме того, пользователь может создать новый табличный документ или просмотреть результаты прежних отчетов, сохраненные в виде файлов в формате табличного документа.

Независимо от источника создания работа с табличным документом осуществляется табличным редактором.

3.3. Создание и открытие табличного документа пользователем

Чтобы создать новый табличный документ, нужно выбрать пункт [Файл – Новый](#). В появившемся запросе выбрать строку [Табличный документ](#) и следует нажать кнопку [ОК](#). Будет открыто окно табличного редактора, содержащее пустой документ.

Чтобы открыть существующий табличный документ, следует выбрать пункт [Файл – Открыть](#). На экран будет выдан стандартный диалог открытия файла.

В этом диалоге выбрать имя файла документа. В поле [Тип файла](#) можно выбрать формат открываемого табличного документа.

3.3.1. Импорт документов Microsoft Excel и OpenDocument

С помощью команды [Файл – Открыть](#) можно открывать табличные документы в форматах Microsoft Excel 97 – 2010 ([Книга Excel \(*.xls, *xlsx\)](#)) и OpenDocument ([Электронная таблица ODF \(*.ods\)](#)). При этом создается табличный документ и в него импортируется оригинальный файл.

Импорт файлов табличных документов имеет следующие особенности:

1. Не поддерживается импорт документов, защищенных паролем.
2. Не импортируются диаграммы.

3. Не импортируются оформления сводных таблиц.
4. При импорте цветов не поддерживается прозрачность.
5. Градиентные заливки поддерживаются частично: при импорте градиентная заливка заменяется на сплошную, при этом выбирается цвет первой точки градиента.
6. Не импортируются OLE-объекты.
7. Настройки оформления текста импортируются с ограничениями.
8. Не поддерживается заполнение автофигур текстурами и картинками.
9. Не импортируются графические эффекты, примененные к картинкам и фигурам (графические фильтры, размытие по краям, свечение, сглаживание и т.д.).
10. Не поддерживаются повороты автофигур и прикрепленного к ним текста.
11. Не поддерживается импорт гиперссылок (текст ссылок импортируется, но переход по ссылке не осуществляется).
12. Автофигуры: импортируются только эллипсы, прямоугольники, прямые линии.
13. Автофигуры: Для линий не поддерживается импорт параметров стрелки.
14. Не поддерживается импорт различных цветов границ для одной ячейки.
15. Импортируются только результаты формул.
16. При импорте документов Microsoft Excel из буфера обмена: при импорте цветов используется цветовая схема Excel по-умолчанию.
17. Условное форматирование: При применении правил форматирования учитываются только константы, значения формул и ссылки на значения других ячеек не поддерживаются.
18. Условное форматирование документов Microsoft Excel: не поддерживаются гистограммы, наборы значков, правила типа «формула равна».
19. При импорте книги, содержащей несколько непустых листов, после каждого листа выводится разрыв страницы.

3.4. Сохранение табличного документа

Любой табличный документ можно сохранить в файл для того, чтобы его в дальнейшем можно было бы открыть, просмотреть и распечатать. Для сохранения табличного документа нужно выбрать пункт **Файл – Сохранить**. При сохранении нового документа на экран выводится диалог сохранения файла, в котором нужно указать каталог и имя файла. При сохранении отредактированного текстового документа (файл уже был) диалог не вызывается.

Чтобы сохранить табличный документ под другим именем или в другом формате, используется пункт **Файл – Сохранить как**.

Обычно табличные документы сохраняются в специальном формате, используемом системой «1С:Предприятие» для хранения табличных документов. Такие файлы имеют расширение *.mxl. Также имеется возможность для сохранения табличного документа в других форматах – лист Excel (расширение *.xls), документ HTML, документ HTML3, документ HTML4, документ HTML5 (расширение *.htm), документ PDF (расширение *.pdf), текстовый файл и текстовый файл UNICODE (расширение *.txt), табличный документ v7.x (формата предыдущих версий системы «1С:Предприятие»), лист Excel95 (расширение *.xls), лист Excel97 (расширение *.xls), документ Word2007 (расширение *.docx), лист Excel2007 (расширение *.xlsx), электронная таблица ODF (расширение *.ods). Для этого нужно выбрать в поле **Сохранить как**: необходимый формат файла.

ПРИМЕЧАНИЕ 1. Необходимо иметь в виду следующие ограничения, существующие при экспорте в формате DOCX:

– при выводе текста с установленной ориентацией текст выводится либо горизонтально, либо вертикально, в зависимости от угла текста;

– при выводе рисунка типа **Текст** текст всегда выводится с горизонтальной ориентацией;

– размещение текста только в виде **Переносить**. Другие виды размещения текста выводятся как **Переносить**;

– если у прямоугольника выводится хоть одна граница, то выводятся все границы.

ПРИМЕЧАНИЕ 2. При открытии документа XLSX в Microsoft Excel версии ниже 2007 возможно несоответствие отображаемых цветов исходным цветам табличного документа.

ПРИМЕЧАНИЕ 3. При сохранении документа в формате Документ HTML, документ будет сохранен в формате HTML4: диаграммы и примитивные рисунки будут выводиться в результирующий документ при помощи встроенных рисунков формата SVG.

ПРИМЕЧАНИЕ 4. При сохранении документа в формате PDF указание количества экземпляров игнорируется и сохраняется только одна копия документа.

При экспорте в файл формата DOCX следующие параметры печати табличного документа сохраняются: размер страницы, ориентация страницы, масштаб печати, разбивка по страницам, настройки полей, настройки колонтитулов, отображение шапки на каждой странице, возможность отключения вывода колонтитулов на первой странице.

При выводе в табличный документ ячейка, содержащая многострочный текст, обрамляется двойными кавычками, двойные кавычки внутри многострочной строки заменяются двумя двойными кавычками.

При выборе формата Лист Excel табличный документ сохраняется в формате Microsoft Office Excel, используемом по умолчанию (эквивалентно листу Excel97).

Сохранение в формате документа HTML позволяет открывать сохраненный файл любым средством для просмотра HTML-документов, например, Microsoft Internet Explorer. При сохранении используется формат HTML-4 с сохранением большинства параметров форматирования. При выводе табличного документа в HTML-документ также сохраняются рисунки и диаграммы.

При сохранении табличного документа в форматах HTML-3 и HTML-4 необходимо учитывать следующие особенности:

- Если у непустой ячейки табличного документа свойство **Размещение** установлено в значение **Авто**, то ячейки объединяются до следующей заполненной ячейки, либо до конца табличного документа по горизонтали.
- Если рисунок расположен в нескольких ячейках табличного документа, то объединяются все перекрытые ячейки. При этом отображается только текст, расположенный в левой верхней ячейке из перекрытых ячеек.
- В табличном документе не поддерживается наложение текста и рисунка в ячейке. Отображается вначале рисунок, затем – текст.

3.5. Закрытие табличного документа

Закрытие табличного документа производится стандартным образом.

В связи с тем, что табличные документы в основном используются для представления информации пользователю (отчеты), для закрытия документа можно использовать клавишу **Esc**. Если табличный документ не был модифицирован, то он закрывается, в противном случае на экран выводится запрос о необходимости сохранения документа.

3.6. Просмотр табличного документа

Перемещение по таблице. Согласно общей концепции интерфейса Microsoft Windows окно табличного документа в любой момент времени показывает только часть таблицы, лежащей «под» этим окном. Для перехода к любой ячейке в видимой области достаточно щелкнуть эту ячейку мышью. Выбранная ячейка становится активной. Для вывода на экран скрытых за границами окна областей используются линейки прокрутки.

Для удобства прокрутки большого документа можно нажать колесо мыши и выполнять прокрутку, выбирая курсором необходимое направление прокрутки. Скорость прокрутки зависит от расстояния между текущим положением курсора и тем местом, где было нажато колесо (отображается специальным образом).

Управление видом таблицы. При помощи пунктов подменю **Таблица – Вид** можно управлять отображением различных областей и компонентов табличного документа: заголовков строк и колонок, сеткой таблицы и т. д.

Кратко поясним назначение пунктов меню **Вид**. Каждый пункт и каждая кнопка работает как переключатель: выбор (нажатие) вызывает или прекращает отображение соответствующей области.

Пункт меню	Действие
Зафиксировать таблицу	Фиксирует верхние строки и левые колонки табличного документа таким образом, чтобы при перелистывании табличного документа они постоянно присутствовали на экране
Отображать сетку	Включает и отключает показ сетки табличного документа – тонких линий, разделяющих ячейки таблицы
Отображать заголовки	Управляет показом заголовков строк и колонок табличного документа
Отображать группы	Включает и отключает показ групп – областей слева от заголовков строк и сверху от заголовков колонок
Редактирование	Включает и отключает запрет на любые изменения в табличном документе
Отображать примечания	Включает и отключает режим показа примечаний
Черно–белый просмотр	Отключает и включает показ цветов, заданных для табличного документа. При включении этого пункта (в меню напротив него появляется галочка) все цвета будут определяться установками операционной системы Microsoft Windows
Режим просмотра страниц	Отключает и включает показ минимального количества строк (определяется введенной информацией), области печати, разрывов страниц
Масштаб	Дискретно масштабирует изображение

Масштабирование изображения. Для удобства просмотра табличного документа изображение можно дискретно масштабировать. Для выбора масштаба изображения используется подменю **Таблица – Вид – Масштаб**: при выборе этого пункта открывается подменю, в котором можно выбрать желаемый масштаб изображения табличного документа.

Для масштабирования можно использовать мышь с колесом прокрутки. Следует нажать клавишу **Ctrl** и, не отпуская ее, повернуть колесо. Движением колеса от себя масштаб увеличивается, к себе – уменьшается.

Фиксация таблицы (шапки и боковика). Редактор табличных документов системы «1С:Предприятие» позволяет зафиксировать верхние строки и левые колонки табличного документа таким образом, чтобы при листании табличного документа они постоянно присутствовали на экране. Это удобно, когда требуется просматривать какую-либо информацию, оформленную в виде таблицы большого размера. В этом случае на экране постоянно будут присутствовать шапка и боковик табличного документа.

Чтобы зафиксировать верхние строки табличного документа, следует выделить строку табличного документа целиком и выбрать пункт **Таблица – Вид – Зафиксировать таблицу**. Зафиксированы будут все строки, расположенные выше выделенной строки.

Чтобы зафиксировать колонки табличного документа, следует выделить колонку табличного документа целиком. Зафиксированы будут колонки, расположенные слева от выделенной колонки.

Чтобы зафиксировать одновременно строки и колонки табличного документа, следует выделить одну ячейку табличного документа. Зафиксированы будут строки и колонки, расположенные соответственно выше и левее выделенной ячейки.

Если в табличном документе существуют зафиксированные строки и (или) колонки, для отмены фиксации следует повторно выбрать пункт **Зафиксировать таблицу**.

Защита от редактирования. Для запрета редактирования табличного документа используется пункт **Таблица – Вид – Редактирование**, который должен быть отключен.

Если ячейка содержит текст, который полностью не помещается в видимой области, то при подведении указателя мыши текст показывается в виде подсказки. Если размер текста достаточно большой, чтобы отобразить его в виде подсказки, показывается только часть текста. При редактировании такого текста подсказка не показывается, пока не будет сдвинут указатель мыши.

Прокрутка табличного документа. Просмотр информации, размещенной в табличном документе, выполняется

стандартными для Microsoft Windows способами. Если мышь имеет колесо прокрутки, то при его вращении содержимое прокручивается вертикально. При нажатой клавише **Shift** прокручивание производится по горизонтали.

3.6.1. Имена

Отдельным ячейкам табличного документа, диапазонам ячеек и графическим объектам можно присваивать имена.

Работа с именами производится при помощи диалога **Имена**.

Для присвоения имени следует выделить нужную область, выбором пункта **Таблица – Имена – Имена...** открыть диалог **Имена**, ввести имя в поле ввода и выбрать кнопку **Присвоить**.

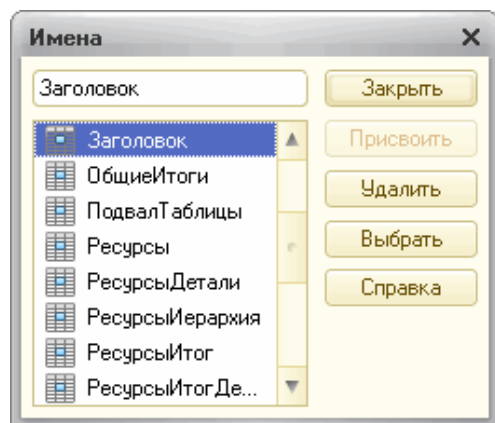


Рис. 242. Диалог «Имена»

Для удаления имени нужно выбрать его из списка и нажать кнопку **Удалить**. При нажатии кнопки **Выбрать** область, соответствующая выбранному имени, становится выделенной. Кнопка **Закреть** закрывает диалог.

Если целиком выделить строку или несколько строк (колонка или несколько колонок) и этой области назначить имя, то при установленном режиме **Отображение именованных строк/колонок** имена областей показываются слева (для строк) и сверху (для колонок). Внутренние именованные области показываются при установленном режиме **Отображать именованные ячейки** штриховкой. В центре области помещается имя области.

Областям табличного документа, не имеющим имени, присвоенного пользователем, программа формирует имена автоматически. Эти имена не присутствуют в списке диалога **Имена**. Их можно видеть на палитре свойств. Области табличного документа нельзя присвоить имя, совпадающее с автоматически формируемым именем. Эти имена имеют, например, следующий вид:

- **R5C7** – для ячейки, находящейся на пересечении пятой строки и седьмой колонки;
- **R5C7:R8C15** – для диапазона ячеек в виде прямоугольника, ограниченного по вертикали пятой и восьмой строками и по горизонтали седьмой и пятнадцатой колонкой;
- **R6** – для шестой строки;
- **R3:R9** – для диапазона строк с третьей по девятую;
- **C2** – для второй колонки;
- **C11:C20** – для диапазона колонок с одиннадцатой по двадцатую;
- **D12** – для рисунка, где 12 – порядковый номер графического объекта;
- **T** – для всего табличного документа.

Имена областей уникальны в пределах одного документа, т. е. различные области должны иметь различные имена, причем регистр букв в именах не различается.

3.6.2. Ввод текста в ячейку

Будем называть текстом любую комбинацию символов, включающую буквы, цифры, знаки препинания, пробелы и служебные символы. Числа, введенные в ячейку, также считаются текстом.

Текст может быть введен в любую ячейку табличного документа. При формировании документа или отчета текст ячейки обрабатывается в зависимости от установленных свойств ячейки и после обработки переносится в готовый отчет.

Ввод текста и установка его атрибутов выполняются только в активной ячейке. Чтобы сделать ячейку активной, нужно щелкнуть на ней левой кнопкой мыши.

Для ввода текста в пустую ячейку нужно просто начать набирать нужные символы на клавиатуре. Ячейка перейдет в режим редактирования – в ней появится текстовый курсор (мигающая вертикальная черта), и в ячейке будет отображаться вводимый текст. Для окончания ввода текста следует нажать **Enter**. Закончить редактирование текста можно также щелчком мыши по любой невыделенной ячейке. Это действие равносильно нажатию клавиши **Enter**.

Если редактируемая ячейка уже содержала какой-либо текст, то при вводе новых символов этот текст будет удален.

Если необходимо отредактировать текст в ячейке, не замещая, а дополняя существующий текст, перед вводом текста следует нажать клавишу **Enter** или **F2** или выбрать ячейку двойным щелчком мыши. Тогда ячейка перейдет в режим редактирования, и можно поместить курсор в нужную позицию текста, используя клавиши управления курсором или мышь.

ВНИМАНИЕ! Следует особо отметить комбинацию **Shift + Enter**. Нажатие этих клавиш позволяет начать новую строку текста в ячейке, не прекращая редактирование ячейки. Новую строку текста в ячейке также можно начать с помощью этой же комбинации при вводе текста в свойстве **Текст** палитры свойств ячейки.

Чтобы отменить изменения, сделанные в тексте ячейки, следует нажать **Esc** до окончания редактирования в ней.

Если текст полностью не помещается в ячейке, то для его показа будет использовано пространство соседних по горизонтали ячеек (слева, справа или с обеих сторон – зависит от типа выравнивания текста в ячейке).

Если необходимо ввести одинаковый текст сразу в несколько ячеек, то следует выделить эти ячейки и в свойстве **Текст** ввести нужный текст. Для размещения текста в ячейках нужно нажать кнопку **Сохранить** панели инструментов палитры свойств. Или запомнить нужный текст в буфере обмена, выделить группу ячеек (не обязательно последовательных) и вставить текст из буфера обмена.

Введенный текст можно расположить под произвольным углом наклона. Для этого нужно выделить ячейки, текст которых требуется расположить под углом, и в свойстве **Ориентация** указать нужный угол наклона от -360° до 360° .

Для каждой ячейки табличного документа можно создать примечание – произвольный текст, описывающий данную ячейку. Ячейка, содержащая примечание, помечается маркером в виде красного треугольника, расположенного в правом верхнем углу ячейки. Если установлен режим **Отображать примечания**, то текст примечания показывается всегда. Если режим не установлен, то примечание показывается в виде ярлыка при подведении указателя мыши к ячейке.

Во время ввода значения ячейки в свойстве **Маска** можно указать маску, которая будет определять ввод символов в строке.

Клавиши, используемые для редактирования. В процессе редактирования текста в ячейке табличного документа вы можете использовать те же клавиши и комбинации клавиш, которые используются в других приложениях Microsoft Windows.

Если какие-либо действия с табличным документом были сделаны по ошибке, то их можно отменить. Для этого в редакторе имеется средство отмены. После ошибочных действий нужно выбрать пункт **Правка – Отмена**. Редактор позволяет отменить все произведенные изменения, выполненные после открытия или создания табличного документа.

Если требуется вернуть назад отмененные действия, то следует выбрать пункт **Правка – Вернуть**.

3.7. Выделение ячеек, строк и колонок табличного документа

Все действия по редактированию данных, содержащихся в табличном документе, производятся в активной ячейке. Кроме этого, некоторые операции допустимо выполнять над произвольной группой ячеек или только над диапазоном ячеек (так мы будем называть компактную группу ячеек, имеющую прямоугольную форму).

Перед выполнением таких операций соответствующая ячейка, диапазон или группа ячеек табличного документа должны быть выделены. Номера колонок и строк выделенной группы ячеек также выделяются. Это позволяет

быстрее определить положение группы.

Выделение какого-либо элемента (ячейки) табличного документа или группы элементов выполняется стандартными для Microsoft Windows действиями.

Для снятия выделения нужно щелкнуть мышью любую ячейку.

3.8. Поиск и замена

Редактор табличных документов системы «1С:Предприятие» позволяет искать в табличном документе любую последовательность символов, а также заменять ее другой последовательностью.

Поиск. Для осуществления поиска необходимо выбрать пункт **Правка – Найти**. На экран будет выдан диалог для установки параметров поиска.

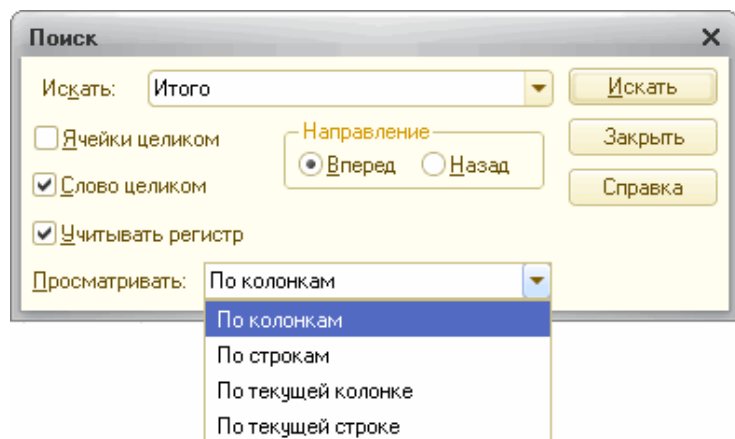


Рис. 243. Поиск в табличном документе

В поле **Искать** этого диалога следует ввести образец для поиска. Строка – образец поиска может быть сколь угодно длинной.

Чтобы различать при поиске прописные и строчные буквы, нужно включить опцию **Учитывать регистр**.

При включенной опции **Ячейки целиком** будут найдены только те ячейки, в которых с искомым текстом совпадает весь текст ячейки, а не его часть.

Всего имеется восемь вариантов обхода документа при поиске. Нужный вариант выбирается с помощью группы **Направление** и поля со списком **Просматривать**. Все варианты обхода являются циклическими.

Группа **Направление** позволяет указать общее направление поиска: **Вперед** – по строкам вправо и по колонкам вниз от текущей позиции; **Назад** – по строкам влево и по колонкам вверх от текущей позиции.

Поле со списком **Просматривать** позволяет задать диапазон и порядок перебора направлений поиска, предоставляя выбор из четырех вариантов:

- **По колонкам** – искать в пределах колонки как по текущей, после достижения конца (или начала) колонки менять колонку поиска, сдвигаясь вправо или влево на одну ячейку в указанном общем направлении, после достижения конца (или начала) документа циклически переходить к началу (или концу);
- **По строкам** – искать в пределах строки как по текущей, после достижения конца (или начала) строки менять строку поиска, сдвигаясь вниз или вверх на одну ячейку в указанном общем направлении, после достижения конца (начала) документа циклически переходить к началу (концу);
- **По текущей колонке** – искать, начиная от текущей ячейки, сдвигаясь вниз или вверх в пределах колонки в зависимости от указанного общего направления; после достижения конца колонки (или начала – при направлении назад) циклически переходить к началу (или, соответственно, к концу);
- **По текущей строке** – искать, начиная от текущей ячейки, вправо или влево в зависимости от указанного общего направления; после достижения конца строки (или начала – при направлении назад) циклически переходить к началу (или, соответственно, к концу).

После установки необходимых параметров для выполнения поиска следует нажать кнопку **Искать**. Если ячейка с указанной последовательностью символов будет найдена, то она станет текущей.

Независимо от ранее заданного общего направления можно повторить поиск, задав ему направление [Вперед](#), если выбрать пункт [Правка – Найти следующий](#) или нажать клавишу [F3](#).

Независимо от ранее заданного общего направления можно повторить поиск, задав ему направление [Назад](#), если выбрать пункт [Правка – Поиск – Найти предыдущий](#) или нажать клавиши [Shift + F3](#).

Замена. Для выполнения замены символьной строки необходимо выбрать пункт [Правка – Заменить](#). На экран будет выдан диалог для установки параметров замены.

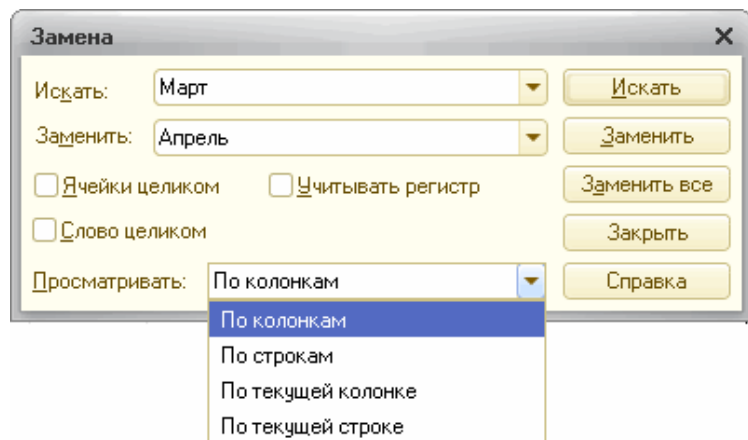


Рис. 244. Настройка замены в табличном документе

Действия в диалоге аналогичны действиям при поиске информации. В поле [Заменить](#) необходимо указать последовательность символов, на которую будет заменяться образец поиска, указанный в поле [Искать](#).

При замене не применяется выбор направления замены. Подразумевается режим [Вперед](#).

После установки необходимых параметров для начала процесса поиска и замены нужно нажать кнопку [Искать](#).

Если ячейка с указанной последовательностью символов будет найдена, то она станет текущей.

Для замены найденной последовательности символов следует нажать кнопку [Заменить](#). Будет произведена ее замена, и поиск продолжится дальше.

Чтобы заменить сразу все вхождения искомой строки, следует нажать кнопку [Заменить все](#).

3.9. Изменение высоты строк и ширины колонок

Можно изменять высоту строк и ширину колонок табличного документа вручную или автоматически.

Изменение высоты строк и ширины колонок вручную. Для изменения высоты строки нужно установить указатель мыши на нижнюю границу заголовка строки, пока курсор мыши не примет форму раздвоенной вертикальной стрелки. Теперь можно, перемещая мышью границу заголовка, установить нужную высоту строки.

Для изменения ширины колонки нужно установить указатель мыши на правую границу заголовка колонки, пока курсор мыши не примет форму раздвоенной горизонтальной стрелки. Теперь можно, перемещая мышью границу заголовка, установить нужную ширину колонки табличного документа. Чтобы установить минимально необходимую ширину колонки, достаточно дважды щелкнуть мышью границу заголовка. При этом редактор произведет анализ текста всех ячеек и установит такую ширину колонки, чтобы тексты ячеек полностью помещались в колонке.

Изменение ширины колонок и высоты строк можно производить с помощью мыши при нажатой клавише [Ctrl](#). Следует нажать клавишу [Ctrl](#) и подвести указатель мыши к границе любой ячейки, расположенной в нужной колонке или строке. Курсор мыши изменит форму. Затем следует установить нужный размер. Такой способ удобен, когда не показаны заголовки колонок и строк (обычно для табличного документа, размещенного в форме).

Изменение высоты строк и ширины колонок автоматически. Для изменения ширины колонки следует выделить нужную колонку. Затем выбрать пункт [Таблица – Ячейки – Ширина колонок](#). В выданном на экран запросе указать требуемую ширину колонки и нажать кнопку [OK](#) – ширина колонки будет установлена равной указанному значению.

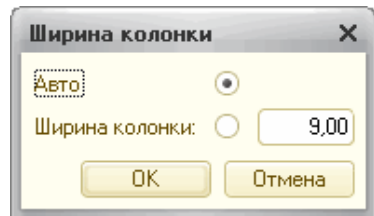


Рис. 245. Редактор табличных документов. Ширина колонки

Для автоматического вычисления высоты строки в зависимости от высоты введенного в строку текста следует установить флажок **Автовысота строки**. Если флажок **Максимальная высота строк** установлен, то в поле указывается значение максимально возможной высоты строки. При этом если реальная высота строки меньше, чем указанная, строка будет показана исходя из реальной высоты. Если реальная высота превышает максимальное значение, высота строки будет установлена в соответствии с максимальным значением. При этом часть строки будет не видна. Если флажок **Максимальная высота строк** не установлен, то высота строки будет всегда устанавливаться по значению реальной высоты строки.

Если флажок **Автовысота строки** не установлен, то в поле **Высота строк** задается фиксированный размер строки.

Порядок автоматической установки ширины колонки отличается от установки высоты строки. При создании нового табличного документа ширина всех колонок устанавливается равной стандартной ширине колонки, которая по умолчанию равна девяти символам шрифта, установленного для текста табличного документа.

Заданную по умолчанию стандартную ширину колонки можно изменить. Для этого необходимо выделить весь табличный документ, выбрать пункт **Таблица – Ячейки – Ширина колонок** и в появившемся запросе ввести число – новую стандартную ширину колонок, выраженную в количестве символов шрифта, установленного для текста табличного документа. После нажатия кнопки **ОК** ширина всех колонок будет изменена в соответствии с введенным значением. При выборе **Авто** устанавливается значение, указанное для ширины колонок всего табличного документа. Если для указания ширины колонок всего табличного документа также выбирается значение **Авто**, то устанавливается ширина 9.

Изменение ширины колонки производится аналогично изменению высоты строки.

Если в ячейках введена информация и требуется установить минимальную ширину колонок или высоту строк, то достаточно дважды щелкнуть мышью разделитель колонок или строк. При этом программа определит максимальную ширину или высоту по введенному тексту и изменит размер таким образом, чтобы введенный текст полностью размещался в ячейках.

3.10. Перемещение и копирование ячеек

Любую ячейку или диапазон ячеек табличного документа можно переместить в другое место табличного документа. Точнее перемещается не сама ячейка, а содержащийся в ней текст и атрибуты форматирования ячейки. Также можно копировать ячейку, перенося таким образом атрибуты форматирования ячейки-образца в другие ячейки табличного документа.

Перемещение ячеек. Для перемещения ячейки или диапазона ячеек нужно выполнить следующие действия:

- выделить ячейку или диапазон ячеек, которые требуется перенести;
- указать мышью на рамку, обрамляющую ячейки, так, чтобы указатель мыши принял форму стрелки ;
- перетащить ячейки на новое место.

Если необходимо переместить ячейки в ту часть табличного документа, которая скрыта границами окна, нужно подвести указатель мыши к краю видимой части табличного документа, и табличный документ начнет автоматически «прокручиваться».

Копирование ячеек. С помощью мыши легко выполняется копирование ячейки или диапазона ячеек. Для этого нужно выполнить те же действия, что и при перетаскивании, но при нажатой клавише **Ctrl**.

Использование буфера обмена. Для перемещения и копирования ячеек можно использовать также буфер обмена Microsoft Windows.

Разрешается копирование блока смежных ячеек (расположенных рядом).

Для копирования в буфер обмена можно использовать кнопку **M** (**Запомнить в буфер обмена как число**) панели инструментов. В этом случае в буфере обмена будет запомнено содержимое ячейки как число (если копируемая

ячейка содержала число). Если выделен диапазон ячеек, то при нажатии кнопки **M** в буфер обмена будет помещена сумма чисел, находящихся в выделенных ячейках.

Следует заметить, что копия ячейки остается в буфере обмена до тех пор, пока он не будет очищен или в него не будет помещена какая-либо другая информация. Это свойство можно использовать, когда необходимо создать несколько копий одной и той же ячейки. Например, если скопировать одну ячейку, затем выделить блок ячеек, то с помощью команды **Вставить** содержимое буфера обмена вставится в каждую ячейку блока. Таким образом можно скопировать сразу несколько смежных (расположенных рядом) ячеек. Но при выделении блока следует учитывать, что размеры (число и порядок ячеек) блока должны быть кратны размерам копируемых ячеек. В противном случае будет выдано предупреждение: **Выделение не соответствует вставляемому блоку**.

Если при перемещении или копировании вместо команды **Вставить** использовать команду **Специальная вставка**, на экран будет выдан запрос.

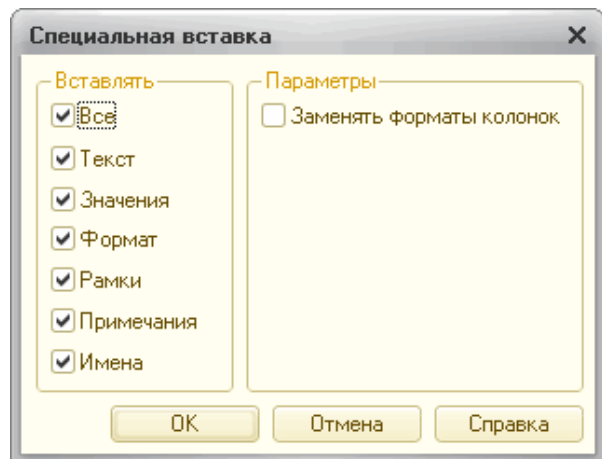


Рис. 246. Настройка специальной вставки в табличный документ

Установкой соответствующих опций этого запроса можно указать, что необходимо вставить в табличный документ: ячейку целиком, только текст, значения ячеек, формат, оформление, примечания или имена.

Опция **Заменять форматы колонок** позволяет сохранять исходный формат колонок (подробнее см. раздел «Очистка ячеек» [здесь](#)), установленный в исходном документе.

«Расширенное» перетаскивание. Для перемещения и копирования ячеек можно использовать также правую кнопку мыши. При этом осуществляется так называемая «расширенная» операция перетаскивания. Если не считать замены левой кнопки мыши на правую, способ осуществления операции тот же, как при описанном выше перемещении, но при отпускании кнопки возникает контекстное меню.

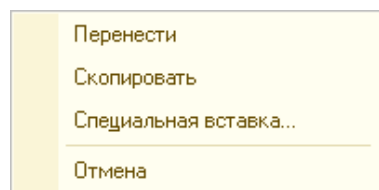


Рис. 247. Контекстное меню «расширенного» перетаскивания

Это меню позволяет определить, что именно будет сделано с помощью расширенной операции перетаскивания: перемещение, копирование или копирование с вышеописанной специальной вставкой. Можно также отменить операцию.

Переход к ячейке. Для перемещения к ячейке можно воспользоваться командой меню **Таблица – Ячейки – Перейти к ячейке**. В открывшемся диалоге ввести требуемые номера строки и колонки.

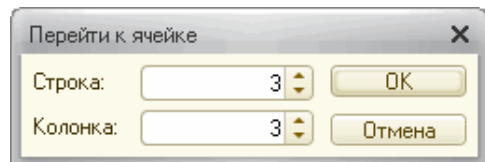


Рис. 248. Диалог перехода к ячейке

При нажатии кнопки **OK** будет выполнен переход к указанной ячейке.

3.11. Добавление и удаление ячеек

Достаточно часто возникает необходимость добавить в существующий табличный документ новую ячейку или вставить целиком строку или колонку, а также удалить существующие строки или колонки.

Вставка строки (колонки). Для вставки строки или колонки нужно выполнить следующие действия (следует учесть, что новые строки помещаются в табличный документ сверху от выделенных, а новые колонки – слева):

- выделить в табличном документе такое количество строк или колонок, которое необходимо вставить, и в том месте, которое должны будут занимать вставляемые строки или колонки;
- выбрать пункт [Таблица – Ячейки – Раздвинуть](#).

Также для вставки строк можно использовать соответствующие пункты меню [Таблица – Вставить строки выше/Вставить строки ниже](#).

Для вставки колонок ячеек можно использовать пункты меню [Таблица – Вставить колонки слева/Вставить колонки справа](#).

Удаление строки (колонки). Для удаления строки или колонки нужно:

- выделить в табличном документе те строки или колонки, которые необходимо удалить;
- выбрать пункт [Правка – Удалить](#).

Чтобы удалить выделенные строки (или колонки), также можно использовать пункт меню [Таблица – Удалить строки \(Удалить колонки\)](#).

Выделенные строки или колонки удаляются из табличного документа вместе с содержащимся в них текстом. Их место занимают строки, расположенные снизу, или колонки, расположенные справа.

Вставка ячейки. В любое место существующего табличного документа можно вставить новую ячейку.

Для вставки ячейки нужно выполнить следующие действия:

- выделите в табличном документе ячейку, на место которой должна быть вставлена новая ячейка;
- выбрать пункт [Таблица – Ячейки – Раздвинуть](#);
- в выданном на экран запросе выбрать, куда нужно сдвинуть ячейки, чтобы освободить место для вставляемой ячейки: горизонтально – будут сдвинуты ячейки, расположенные справа от выделенной ячейки, вертикально – расположенные ниже;
- нажать кнопку [ОК](#).

Вставка диапазона ячеек. Если необходимо, в табличный документ можно вставить диапазон ячеек.

Для вставки диапазона выполнить следующие действия:

- выделить в табличном документе диапазон такого же размера, какой необходимо вставить, и в том месте, которое должен будет занимать вставляемый диапазон;
- выбрать пункт [Таблица – Ячейки – Раздвинуть](#).
- в выданном на экран запросе выбрать, куда нужно сдвинуть ячейки, чтобы освободить место для вставляемого диапазона: горизонтально – будут сдвинуты ячейки, расположенные справа от выделенного диапазона, вертикально – расположенные ниже;
- нажать кнопку [ОК](#).

Удаление диапазона ячеек. Для удаления диапазона выполнить следующие действия:

- выделить в табличном документе диапазон, который необходимо удалить;
- выбрать пункт [Правка – Удалить](#);

- в выданном на экран запросе выбрать, какой диапазон займет место удаленного: горизонтально – место удаленного займет диапазон, расположенный справа от него; вертикально – диапазон, расположенный ниже;
- нажать кнопку **OK**.

3.12. Разбиение ячеек

Редактор табличных документов системы «1С:Предприятие» позволяет разбивать выбранные ячейки по горизонтали или вертикали.

Для разбиения ячейки следует указать нужную ячейку и выбрать пункт **Таблица – Ячейки – Разбить ячейку**. На экран выводится диалог с запросом вида деления.

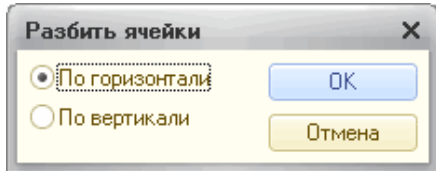


Рис. 249. Выбор деления ячейки

При выборе горизонтального разбиения результат будет таким, как показано на рисунке.

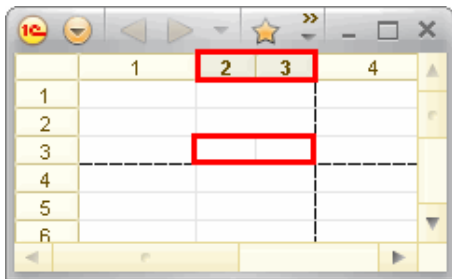


Рис. 250. Разбиение ячейки по горизонтали

Аналогично производится вертикальное разбиение.

3.13. Очистка ячеек

С помощью пунктов меню **Таблица – Ячейки – Очистить** можно очистить как формат содержимого (пункт **Формат**), вид рамки (пункт **Рамки**), содержимое ячейки (пункт **Содержимое** или клавиша **Del**), так и все изменения в ячейке целиком (пункт **Всё**).

3.14. Задание различной ширины колонок

В табличном редакторе системы «1С:Предприятие» существует возможность задания различной ширины колонок для различных строк. Нужно выделить одну или несколько строк табличного документа, как показано на рисунке.

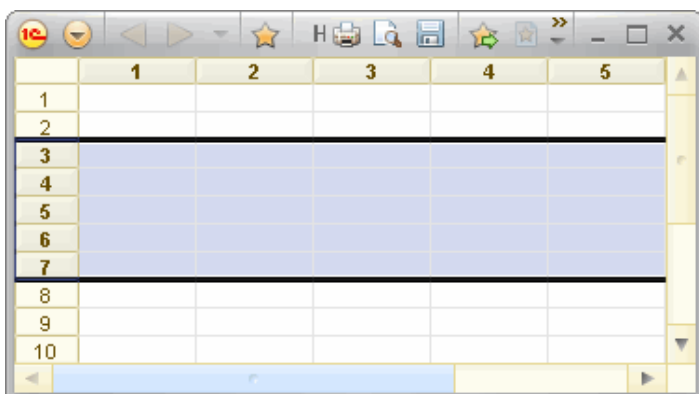


Рис. 251. Редактор табличных документов. Выделено несколько строк

Подвести указатель мыши к разделителю колонок, нажать левую кнопку мыши и, не отпуская ее, потянуть в нужную сторону (например, влево). Отпустить кнопку мыши. На экран выводится запрос на подтверждение

выполнения действий с сообщением **Для выделенных строк будет создан новый формат строк**. При нажатии кнопки **OK** макет будет выглядеть следующим образом:

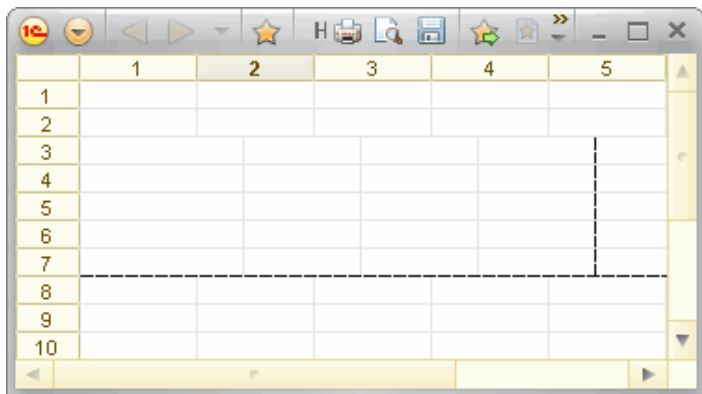


Рис. 252. Редактор табличных документов. Новый формат строк

3.15. Скрытие и отображение строк и колонок

Для удобства просмотра информации редактор табличных документов системы «1С:Предприятие» предоставляет возможность скрытия и отображения строк и колонок табличного документа.

Рассмотрим основные приемы работы на примере данных отчета по курсам валют.

1	2	3	4
1	Отчет по курсам валют		
2	курс валюты: "Доллар США"		
3	Дата	Значение	
4	18.04.2009	33,4677	
5	17.04.2009	33,4677	
6	16.04.2009	33,4677	
7	15.04.2009	33,4677	
8	14.04.2009	33,4677	
9	11.04.2009	33,4677	
10	10.04.2009	33,4677	
11	09.04.2009	33,4677	
12	08.04.2009	33,4677	
13	07.04.2009	33,1743	
14	04.04.2009	33,4095	
15	03.04.2009	33,7636	
16	02.04.2009	33,9456	
17	01.04.2009	33,9032	
18	31.03.2009	34,0134	
19	28.03.2009	33,4133	
20	27.03.2009	33,4668	

Рис. 253. Отчет по курсам валют

Как видно из табличного документа, курс за первые 9 дней не изменялся. Поэтому можно эту информацию скрыть. Для этого выделим строки с 5-й по 13-ю и в контекстном меню выберем пункт **Скрыть**. В результате отчет будет выглядеть следующим образом:

	1	2	3	4
1	Отчет по курсам валют			
2	курс валюты: "Доллар США"			
3		Дата	Значение	
4		18.04.2009	33,4677	
14		04.04.2009	33,4095	
15		03.04.2009	33,7636	
16		02.04.2009	33,9456	
17		01.04.2009	33,9032	
18		31.03.2009	34,0134	
19		28.03.2009	33,4133	
20		27.03.2009	33,4668	
21		26.03.2009	33,7268	
22		25.03.2009	33,2726	
23		24.03.2009	33,3034	
24		21.03.2009	33,4230	
25		20.03.2009	33,8222	
26		19.03.2009	34,4203	
27		18.03.2009	34,5318	
28				
29				

Рис. 254. Отчет по курсам валют. Строки с 5-й по 13-ю скрыты

Обратите внимание на то, как нумерация строк показывает наличие скрытых строк.

Для отображения скрытых строк табличного документа нужно выделить любую область, в которую входят скрытые строки (для нашего примера достаточно выделить строки 4 и 14), и в контекстном меню выбрать пункт [Отобразить](#).

Для отображения всех скрытых областей выделить весь табличный документ.

3.16. Объединение ячеек

При создании макета печатной формы документа часто возникает необходимость размещения текста в области, занимаемой несколькими ячейками (как по горизонтали, так и по вертикали). Для выравнивания текста в пределах нескольких ячеек можно использовать функцию объединения ячеек. Если несколько ячеек объединены в одну, то при выравнивании текста границами выравнивания будут считаться границы этой объединенной ячейки.

Для объединения ячеек необходимо выделить желаемый диапазон ячеек и выбрать пункт [Таблица – Ячейки – Объединить](#).

ПРИМЕЧАНИЕ. Также для объединения ячеек можно использовать сочетание клавиш **Ctrl + M** или соответствующую команду контекстного меню.

Если в объединяемых ячейках был текст, он пропадет, за исключением текста левой верхней ячейки, который станет текстом объединенной ячейки.

Для разъединения объединенной ячейки достаточно выделить ее и повторно выбрать пункт [Таблица – Ячейки – Объединить](#).

Если в объединенной ячейке был текст, при разъединении он попадет в левую верхнюю ячейку, независимо от условий форматирования (например, он мог располагаться в правом нижнем углу объединенной ячейки, будучи сдвинут вправо по горизонтали и вниз по вертикали).

3.17. Работа с именованными областями табличного документа

В табличном документе можно присвоить имя группе ячеек, чтобы по этому имени с помощью средств встроенного языка оперировать областью. Именованные области имеет смысл использовать при создании макета отчета. Для табличного документа именованные области носят второстепенный характер и поэтому не отображаются, как в макете.

Подготовка макета начинается с формирования «кирпичиков», из которых строится отчет. Этими кирпичиками являются именованные области.

3.17.1. Создание именованной области

Именованные области бывают горизонтальные (включают в себя отмеченные строки) и вертикальные (включают в себя отмеченные колонки), а также прямоугольные, включающие смежные ячейки.

Для создания именованной области нужно выделить требуемую область и выбрать пункт [Таблица – Имена – Назначить имя](#). В открывшемся диалоге указать имя области.

3.17.2. Режим просмотра именованных областей

Просмотр именованных строк и колонок возможен при включенном режиме [Таблица – Имена – Отображение именованных строк/колонок](#).

Имена областей показываются для строк между заголовками строк и самой областью, для колонки – между заголовками колонок и самой областью.

Просмотр прямоугольных именованных областей возможен при включенном режиме [Таблица – Имена – Отображать именованные ячейки](#).

Имена областей показываются в виде заштрихованной области. Имя выводится в середине области.

3.17.3. Удаление именованных областей

Для удаления именованной области нужно выделить область, затем выбрать пункт [Таблица – Имена – Убрать имя](#).

3.17.4. Изменение размеров именованных областей

Если область состоит из более чем одной строки или колонки, то для изменения размеров именованной области нужно выделить строки или колонки внутри области и затем выбрать пункт [Таблица – Ячейки – Раздвинуть](#).

Если область состоит из одной строки или колонки, то увеличить размер области нельзя. Нужно убрать имя и создать именованную область заново.

Для уменьшения размера области следует выделить нужное число строк или колонок, которые требуется удалить, и в контекстном меню выбрать пункт [Удалить](#).

Для прямоугольной области можно производить аналогичные действия. В случае, когда выделяются не строки или колонки, а отдельные ячейки, изменение размеров области не производится.

3.18. Работа с группами табличного документа

Выделенные строки или колонки можно объединить в группы для удобства просмотра табличных данных.

Группы можно создавать, удалять, расширять (включать в них дополнительные соседние строки или колонки), создавать вложенные или внешние группы.

3.18.1. Создание группы

Группы бывают горизонтальные (включают в себя отмеченные строки) и вертикальные (включают в себя отмеченные колонки).

Для создания группы нужно сначала отметить необходимые строки или колонки, затем выбрать пункт [Таблица – Группы – Объединить в группу](#).

ВНИМАНИЕ! Нельзя создать группу для нескольких последовательных ячеек.

3.18.2. Просмотр групп

Просмотр групп возможен при включенном режиме [Таблица – Вид – Отображать группы](#).

Группы показываются маркерами с идущими от них тонкими линиями, обозначающими размер группы.

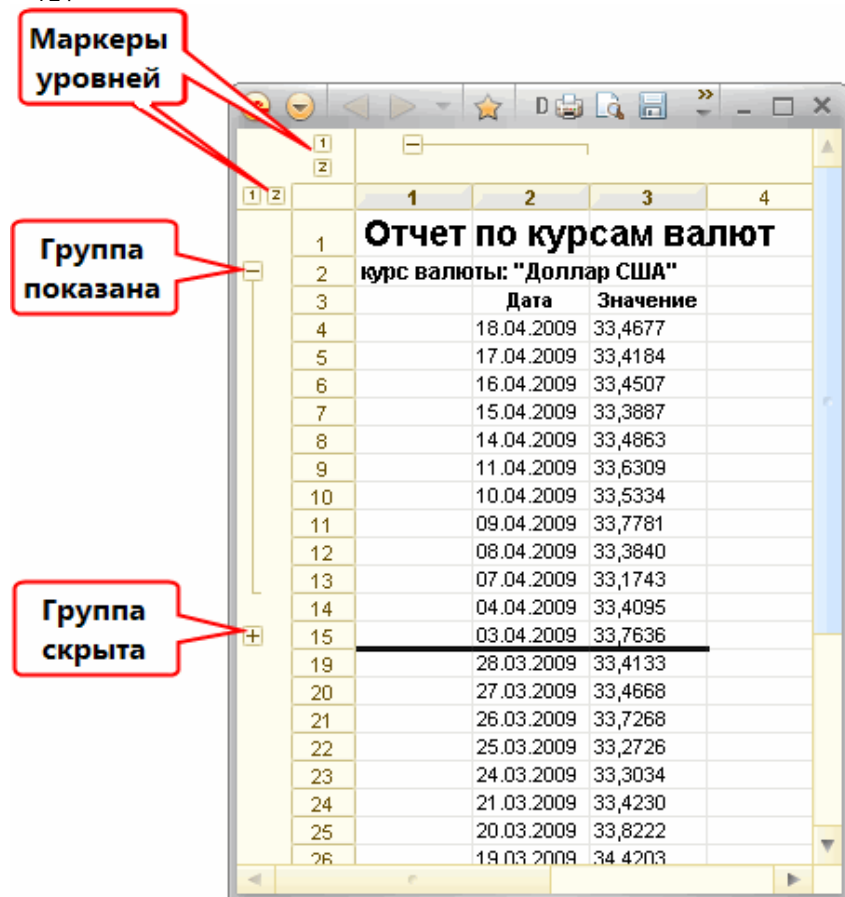


Рис. 255. Группы в табличном документе

Строками и колонками, объединенными в группы, можно управлять. Для управления удобно пользоваться специальными маркерами. Есть две группы маркеров (уровней и групп).

Маркеры уровней размещены в левом верхнем углу табличного документа и пронумерованы. Маркеры отображаются, если установлена возможность показа заголовков ([Таблица – Вид – Отображать заголовки](#)). Маркер с самым большим номером предназначен для отображения всех групп. Нажатие каждого предыдущего маркера скрывает все группы данного уровня, но не влияет на внешние группы.

Маркеры группы располагаются слева от строк или сверху от колонок. Они имеют два состояния – группа показана и группа скрыта (см. [рис. 255](#)). С помощью этих маркеров управляют показом конкретной группы. Указателем мыши нужно щелкнуть маркер группы. Группа скрывается, если была показана, и отображается, если была скрыта. Маркер при этом меняет свой вид.

ПРИМЕЧАНИЕ. Также для работы с группировками можно использовать команды контекстного меню документа [Уровни группировок](#) и следующие сочетания клавиш:

– чтобы свернуть текущую группировку, следует нажать **Ctrl + Num-;**

– чтобы свернуть группировку и все подчиненные, нажать **Shift + Alt + Num-;**

– чтобы свернуть все группировки, нажать **Ctrl + Shift + Num-;**

– чтобы развернуть текущую группировку, нажать **Ctrl + Num+;**

– чтобы развернуть все группировки, нажать **Ctrl + Shift + Num+;**

– чтобы развернуть группировку и все подчиненные, нажать **Shift + Alt + Num+.**

Строка или колонка, напротив которой расположен маркер, называется итоговой строкой. Положение итоговой строки можно менять. Для горизонтальных групп итоговая строка может располагаться сверху или снизу от группы. Для вертикальных групп итоговая строка может располагаться справа или слева от группы. Положение итоговой строки задается в свойствах табличного документа [Итоги по горизонтали](#) и [Итоги по вертикали](#). При смене положения итоговой строки положение маркера группы также меняется. Установка свойств расположения итоговых строк распространяется на все группы табличного документа.

3.18.3. Удаление группы

Для удаления группы нужно выделить ее (все строки или колонки, входящие в группу), затем выбрать пункт [Таблица – Группы – Исключить из группы](#).

3.18.4. Изменение размеров групп. Вложенные и внешние группы

При работе с группами можно изменять размеры группы (расширять – включать дополнительные соседние строки или колонки, сужать – исключать крайние строки или колонки), создавать вложенные или внешние группы.

Расширение группы. Для расширения группы необходимо выполнить следующие действия:

- выделить строки или колонки, которые необходимо включить в новую группу;
- выбрать пункт [Таблица – Группы – Объединить в группу](#).

В зависимости от взаимного расположения выделенного диапазона и существующей группы редактор выполнит одно из следующих действий:

- если отмеченная область строк или колонок не пересекается с существующими группами, то будет создана новая именованная группа;
- если отмеченная область строк или колонок частично пересекается с существующей группой, то существующая группа будет расширена за счет соседних отмеченных строк или колонок;
- если отмеченная область строк или колонок находится внутри существующей группы, то будет создана новая группа, вложенная в существующую;
- если отмеченная область строк или колонок охватывает полностью существующую группу, то будет создана новая группа, внешняя для существующей, таким образом, что существующая группа будет вложена в новую.

Сужение группы. Для сужения группы необходимо выполнить следующие действия:

- выделить нужные строки или колонки;
- выбрать пункт [Таблица – Группы – Исключить из группы](#).

В зависимости от взаимного расположения выделенного диапазона и существующей группы редактор выполнит одно из следующих действий:

- если отмеченная область строк или колонок не пересекается с существующими группами, то никаких действий не произойдет;
- если отмеченная область строк или колонок находится внутри существующей группы, то никаких действий не произойдет;
- если отмеченная область строк или колонок пересекается с существующей группой, то группа будет сужена за счет отмеченных строк или колонок;
- если отмеченная область строк или колонок охватывает полностью существующую группу, то группа будет удалена полностью.

Примером вложенных и внешних групп может служить приведенный ниже рисунок.

1	2	3	4	5
1	Подразделение	Сотрудник	Период	Расчеты
2	Отделение 07	Всего по отделению	за I кв.	109009
3	Отдел 1	Итого по отделу	за I кв.	60866
4		Иванов	за I кв.	43000
5			Январь	16000
6			Февраль	14000
7			Март	13000
8		Петров	за I кв.	17866
9			Январь	3566
10			Февраль	6500
11			Март	7800
12	Отдел 2	Итого по отделу	за I кв.	48143
13		Сидоров	за I кв.	48143
14			Январь	16733
15			Февраль	15766
16			Март	15644
17				

Рис. 256. Пример вложенных групп

В группу первого уровня «Отделение 07» (это внешняя группа) входят две вложенные группы отделов 1 и 2. Эти группы являются внешними по отношению к группам сотрудников.

3.19. Использование графических объектов

При создании табличного документа можно использовать графические объекты. Их применение позволяет улучшить оформление документов или внести в описание формы мелкие добавления, не затрагивая структуры макета.

К графическим элементам относятся линии, прямоугольники, овалы, тексты, картинки, внедренные объекты (OLE-объекты) и диаграммы.

Текстом в табличном редакторе системы «1С:Предприятие» называется рамка с текстом внутри.

Ниже будет рассмотрена работа со всеми графическими объектами, за исключением OLE-объектов.

Свойства графических объектов подробно описаны в разделах Руководства, относящихся к данным объектам.

В этом разделе описываются способы размещения графических объектов в табличном документе.

3.19.1. Вставка графических объектов

Вставка рисованных элементов в табличный документ производится при помощи выбора нужного пункта [Таблица – Рисунки](#) стандартным для Microsoft Windows образом. Настройка графических объектов производится в палитре свойств.

Для размещения картинки нужно предварительно выбрать ту ячейку табличного документа, где должен находиться верхний левый угол картинки, затем выбрать пункт [Таблица – Рисунки – Картинка](#). На экран будет выдан диалог выбора картинки.

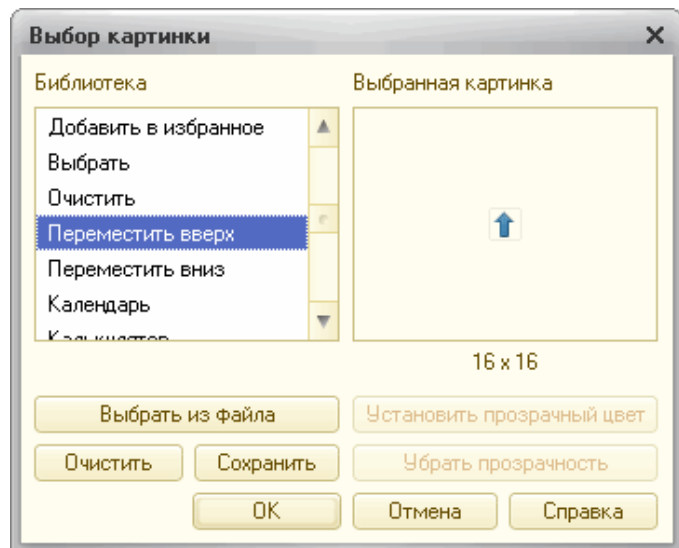


Рис. 257. Диалог выбора картинки для вставки в табличный документ

Картинку можно выбирать из картинок библиотеки (представлены в списке **Библиотека**) или из файла. Для выбора из файла следует нажать кнопку **Выбрать из файла**. На экран выводится стандартный диалог открытия файла, в котором можно указать имя файла, содержащего картинку. Если отказаться от указания имени файла, в табличном документе будет «зарезервировано место» под картинку. Вставляются изображения оригинального размера.

Если выбран файл, то для указанной картинки можно указать, какой цвет картинки будет прозрачным. Обычно в качестве такого цвета выбирают фон картинки, чтобы он не закрывал информацию, расположенную под картинкой.

Картинка всегда вставляется реального размера. Размер области для размещения картинки можно изменить. Для этого нужно выбрать объект **Картинка**, указателем мыши переместить границу или угол объекта в нужном направлении. Если требуется сохранить пропорции области, то предварительно следует нажать клавишу **Shift**.

В свойстве **Размер картинки** нужно выбрать способ размещения картинки в области:

- **Реальный размер** – картинка выводится в оригинальном размере. Если реальный размер меньше размера области, то остаток области не заполняется; если реальный размер больше размера области, то показывается только часть картинки;
- **Растянуть** – картинка полностью размещается в области, независимо от ее размеров;
- **Пропорционально** – при выводе картинки всегда сохраняются пропорции оригинального размера. Выбирается наименьший размер, который позволит разместиться в области; по исходной пропорции размера вычисляется другой размер картинки, и производится размещение картинки;
- **Черепица** – картинка всегда выводится в оригинальном размере. Если размеры области больше, чем оригинальный размер картинки, то программа повторяет выбранную картинку таким образом, чтобы заполнить всю оставшуюся часть области;
- **Авторазмер** – если картинка полностью помещается в поле, то она выводится в оригинальном размере; если картинка больше, то она выводится пропорционально.

Вставляемые стандартные графические объекты (прямоугольник, овал и текст) могут быть прозрачными или иметь различную заливку. По умолчанию при вставке графического объекта в табличный документ устанавливается сплошная заливка. Выбор вида заливки осуществляется в свойстве **Узор** категории свойств **Фон**. Также можно задать цвет фона и цвет узора.

У вставленных графических объектов в свойстве **Стиль линии** можно выбрать вид линии (границы), а в свойстве **Цвет линии** задать ее цвет.

3.19.2. Привязка графических объектов

Вставленные графические объекты автоматически привязываются к ячейкам табличного документа. Привязка осуществляется для верхнего левого и нижнего правого углов объекта. При изменении ширины или высоты строк или колонок, расположенных между этими углами, размер объекта также меняется.

3.19.3. Выделение и изменение размеров графических объектов

Перед тем как проводить какие-либо манипуляции с графическим объектом, его необходимо выделить.

Выделить объект можно двумя способами: непосредственным выбором объекта или с помощью режима выделения графических объектов.

В первом случае для выделения одного объекта необходимо щелкнуть на нем левой кнопкой мыши. Если выделяется прямоугольник, овал или текст, то вокруг него появятся восемь небольших квадратных маркеров: четыре из них расположены в середине каждой из четырех сторон рамки, и четыре – в ее углах.

Размер любого графического объекта можно изменить. Для этого используются маркеры, которые появляются вокруг объекта (или группы объектов) при его выделении. Изменение размеров производится стандартным для Microsoft Windows способом.

Если перед началом перемещения нажать клавишу **Shift** и, не отпуская ее, потянуть мышью за угловой маркер, то размеры объекта изменятся пропорционально существующим размерам объекта.

Для изменения размеров графического объекта можно использовать клавиши управления курсором с нажатой клавишей **Shift**.

Если выделяется линия, то маркеры появятся только в ее начальной и конечной точках. При перемещении с нажатой клавишей **Shift** линия только уменьшает или увеличивает длину, не изменяя угол наклона.

Для выделения нескольких графических объектов используется следующий способ: выделить первый объект, затем нажать клавишу **Shift** и, не отпуская ее, выделить остальные объекты.

Для снятия выделения с объекта или объектов достаточно щелкнуть мышью в любом свободном от объектов месте табличного документа.

Во втором случае выделение графического объекта производится с помощью указателя мыши, которым «рисуются» прямоугольная область. Все графические объекты, полностью попавшие в эту зону, выделяются. Режим выделения устанавливается с помощью пункта [Таблица – Рисунки – Выделение рисованных объектов](#).

Для выключения режима выделения графических объектов следует повторно выбрать пункт [Таблица – Рисунки – Выделение рисованных объектов](#).

3.19.4. Выравнивание группы графических объектов

Выравнивание графических объектов производится с помощью кнопок панели инструментов [Выравнивание](#). Команды доступны (выравнивание производится), если выбрано более одного объекта:

- [Выровнять влево](#) – производит выравнивание левых границ выбранных объектов. При этом их левые границы располагаются вдоль одной вертикальной прямой;
- [Выровнять вправо](#) – выравнивание правых границ;
- [Выровнять вверх](#) – выравнивание верхних границ;
- [Выровнять вниз](#) – выравнивание нижних границ.

Выборную группу объектов можно также выровнять по центру вертикально или горизонтально. Для этого предназначены команды [Выровнять по центру вертикально](#) и [Выровнять по центру горизонтально](#). При этом выравнивание производится по осям объектов.

3.19.5. Установка размера группы графических объектов

С помощью кнопок панели инструментов [Выравнивание](#) можно установить одинаковые размеры всех объектов, входящих в группу.

Следует обратить внимание на то, что за образец выбирается тот объект, маркеры которого имеют вид, как показано на [рис. 258](#). Этот объект называется мастером. Другие объекты имеют маркеры в виде черных квадратов.

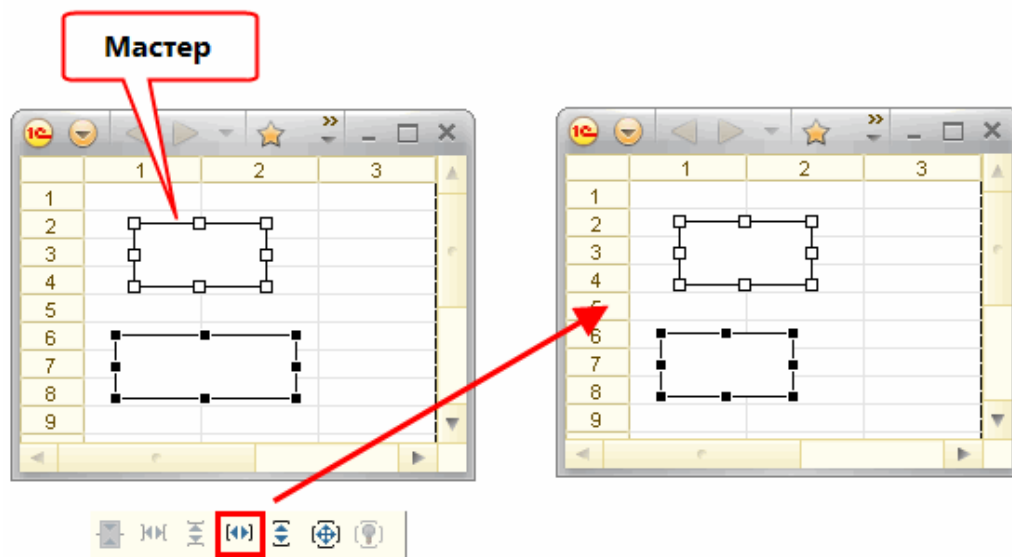


Рис. 258. Установка одинаковой ширины объектов в табличном документе

Чтобы сменить мастера, в группе объектов следует щелкнуть мышью нужный объект:

- **Одинаковая ширина** – устанавливает одинаковую ширину всех объектов группы;
- **Одинаковая высота** – устанавливает одинаковую высоту всех объектов группы;
- **Одинаковый размер** – устанавливает одинаковый размер всех объектов группы.

3.19.6. Перемещение и копирование графических объектов

Любой графический объект можно перемещать в табличном документе, а также копировать, создавая несколько объектов на основе одного объекта-образца.

Перемещение графического объекта. Для перемещения графического объекта нужно перетащить его мышью в новое место. Пунктирная рамка (или линия, если перемещается линия), которая обозначает объект при движении мыши, поможет разместить его в табличном документе.

Перемещение графического объекта происходит также при нажатии на клавиатуре клавиш управления курсором.

Копирование графического объекта. Для копирования графического объекта нужно перетащить его мышью при нажатой клавише **Ctrl**.

Если в процессе переноса копии объекта отпустить клавишу **Ctrl**, объект будет не скопирован, а перемещен.

Использование буфера обмена. Для перемещения и копирования графических объектов можно использовать буфер обмена Microsoft Windows.

Положение вставленного объекта (или группы объектов) будет определяться следующим образом: в левый верхний угол активной ячейки табличного документа помещается левый верхний угол воображаемой прямоугольной рамки, обведенной вокруг вставляемых объектов.

3.19.7. Изменение порядка расположения графических объектов

Когда несколько графических объектов частично или полностью перекрывают друг друга, важно задать для них порядок расположения для достижения требуемого изобразительного эффекта.

Новый рисованный объект всегда помещается на передний план и перекрывает объекты, оказавшиеся под ним. Если представить, что все объекты помещаются один над другим, то получится многослойная конструкция, в которой каждый объект имеет свой порядок следования (слой). В верхнем слое располагается объект, который всегда следует впереди всех объектов. С помощью команд управления порядком расположения можно принудительно поместить объект позади всех других объектов (в нижний слой) и, наоборот, перенести вперед всех (в верхний слой), а также на один слой вверх или вниз.

Для переноса объекта необходимо выделить объект и в подменю **Порядок** контекстного меню выбрать один из пунктов:

- [На передний план](#) – для переноса объекта вперед всех объектов;
- [На задний план](#) – для переноса объекта позади всех других объектов;
- [Переместить вверх](#) – для переноса объекта на один слой вверх;
- [Переместить вниз](#) – для переноса объекта на один слой вниз.

3.19.8. Удаление графических объектов

Для удаления рисованного объекта необходимо выделить его и выбрать пункт [Правка – Удалить](#). Удалить объект можно, если предварительно выделить весь диапазон ячеек, на которых располагается графический объект. Если при выделении какая-то часть объекта окажется на других ячейках, объект останется в табличном документе, однако его размеры при этом могут быть изменены.

3.19.9. Группировка графических объектов

Несколько графических объектов можно сгруппировать (объединить их в один графический объект).

Для группировки графических объектов нужно выделить несколько объектов любым способом.

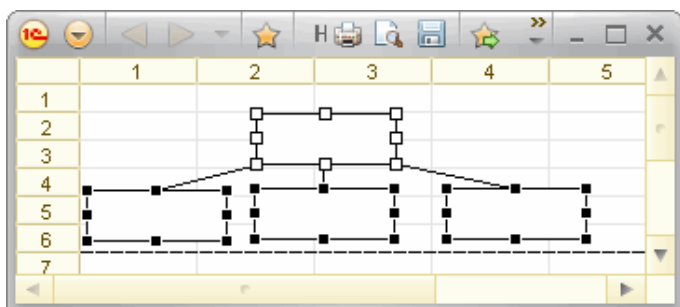


Рис. 259. Выделение нескольких графических объектов

Выбрать пункт [Таблица – Рисунки – Сгруппировать](#).

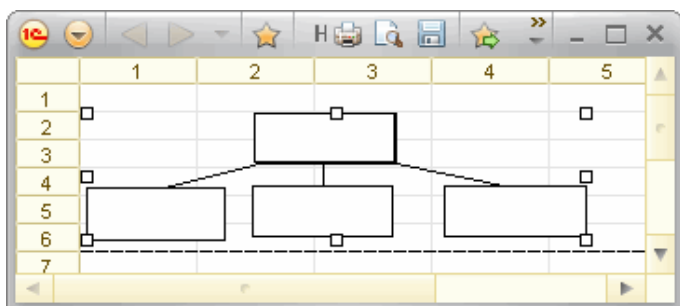


Рис. 260. Графические объекты сгруппированы

Над составным графическим объектом, полученным путем группировки нескольких объектов, можно производить действия, как над обычным графическим объектом.

Если требуется, составной объект можно разгруппировать. Для этого нужно указать составной графический объект и выбрать пункт [Таблица – Рисунки – Разгруппировать](#).

3.19.10. Имена графических объектов

Графическим объектам, как и ячейкам табличного документа, можно присваивать имена. Имена присваиваются при помощи диалога [Имена](#) (пункт [Таблица – Имена](#)).

Вновь создаваемым графическим объектам в режиме [Конфигуратор](#) присваиваются имена по умолчанию, состоящие из буквы D и порядкового номера графического объекта.

3.19.11. Работа с диаграммой

С помощью редактора табличных документов можно включать специальный графический объект – диаграмму, которая позволяет размещать в табличных документах системы «1С:Предприятие» диаграммы и графики

различного вида. Диаграмма может быть создана средствами встроенного языка (в отчетах) или вставлена пользователем для анализа табличных данных. Ниже приводится описание действий по размещению диаграммы пользователем.

3.19.12. Создание диаграммы

Для создания диаграммы необходимо выполнить следующие действия:

- выбрать пункт **Таблица – Рисунки – Диаграмма**;
- в табличном документе нарисовать мышью прямоугольник желаемого размера в том месте, где предполагается поместить диаграмму;
- указать диапазон ячеек, на основании которых будет построена диаграмма (диапазон можно изменить в любой момент).

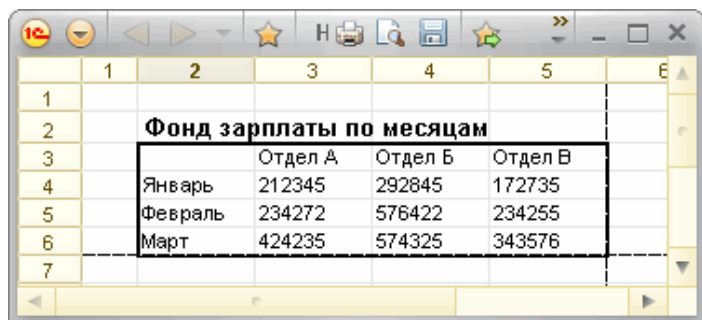
Если предварительно выделить нужный диапазон ячеек с исходными данными, то при вставке диаграммы редактор табличных документов не выдает запрос на ввод диапазона. При вставке диаграмма сразу производит представление данных, указанных в выделенной области.

После создания диаграмму нужно активизировать, например, двойным щелчком мыши, и настроить ее внешний вид с помощью выбора значений свойств в палитре свойств.

3.19.13. Область данных диаграммы

Для табличного документа можно так настроить диаграмму, чтобы она использовала данные, введенные в табличный документ.

Предположим, в табличном документе имеются данные.



	1	2	3	4	5	6
1						
2		Фонд зарплаты по месяцам				
3			Отдел А	Отдел Б	Отдел В	
4		Январь	212345	292845	172735	
5		Февраль	234272	576422	234255	
6		Март	424235	574325	343576	
7						

Рис. 261. Табличный документ. Область данных диаграммы

Для построения гистограммы на основе этих данных нужно выполнить следующие действия:

- выделить область ячеек, включающую таблицу (в данном примере обведена рамкой);
- выбрать пункт **Таблица – Рисунки – Диаграмма**;
- поместить диаграмму.

В табличном документе появится диаграмма, которая представляет указанные данные в графическом виде. Вид диаграммы, заголовок, размеры областей, цвета построения и многие другие параметры настраиваются в палитре свойств диаграммы.

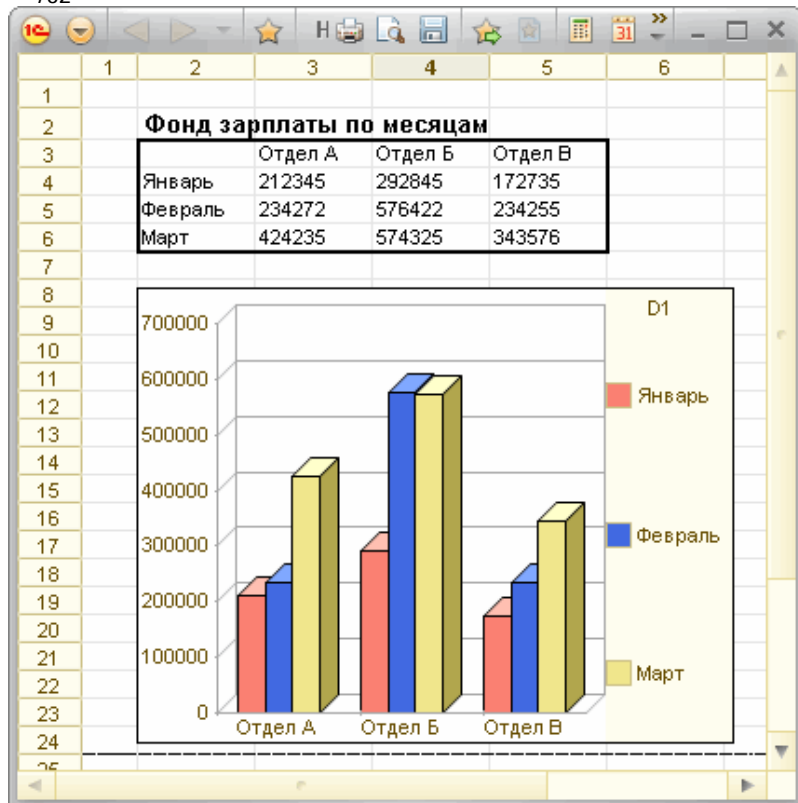


Рис. 262. Табличный документ. Вставка диаграммы

Для настройки диаграммы используется панель инструментов (пункт [Правка – Свойства](#) главного меню).

3.20. Работа с OLE-объектами

Одной из универсальных возможностей приложений, работающих в операционной системе Microsoft Windows, является возможность использования внедренных объектов, называемых также OLE-объектами. OLE-объект – это часть текущего документа, которая создана другим Windows-приложением. Такой OLE-объект, как и любой рисованный объект, является полноценной частью документа с той лишь разницей, что для редактирования такого объекта будет загружено приложение, которым этот объект создан.

Для создания OLE-объекта необходимо выполнить следующие действия:

- выбрать пункт [Таблица – Рисунки – Вставить объект](#);
- нарисовать мышью прямоугольник желаемого размера в том месте, куда предполагается поместить OLE-объект;
- в выданном на экран запросе выбрать тип внедряемого объекта.

Это вызовет загрузку той Windows-программы, которая поддерживает редактирование объектов указанного типа. При этом внешний вид окна режима [Конфигуратор](#) несколько изменится: в главном и контекстных меню появятся функции, а на рабочем поле – панели инструментов, характерные для вызванной программы.

ВНИМАНИЕ! Программы для создания OLE-объектов не входят в комплект поставки. Это отдельные Windows-приложения, являющиеся лицензионными программными продуктами. Об использовании вызванной программы следует читать в документации, поставляемой с ней.

- если требуется, отредактировать OLE-объект;
- закончить редактирование OLE-объекта, щелкнув мышью в любом месте рабочего поля табличного документа вне объекта.

После создания OLE-объект обладает всеми свойствами, присущими «обычным» рисованным объектам: можно менять его размер и положение, копировать, используя мышью или буфер обмена, как угодно компоновать с другими объектами и удалять за ненадобностью. О выполнении этих операций можно узнать из раздела [«Использование графических объектов»](#) [здесь](#).

Но кроме этого, OLE-объект может быть отредактирован – его содержимое можно изменить. Для редактирования

OLE-объекта необходимо дважды щелкнуть на нем мышью. В результате будет загружена та Windows-программа, в которой был создан данный объект, и можно будет отредактировать внедренный объект.

3.21. Работа с табличным документом в режиме «Макет»

Макет печатной формы (отчета, документа и проч.) обычно подготавливается на этапе проектирования конфигурации. Однако табличный документ, размещенный вне конфигурации (в файле), также может быть использован как макет печатной формы.

3.22. Общие принципы проектирования макета

Проектирование макета заключается в «рисовании» составных частей, кирпичиков, из которых затем будет «собрана» готовая выходная форма – документ или отчет. Так как практически все деловые документы имеют «прямоугольную» структуру, удобнее всего создавать макеты таких документов в редакторе, способном манипулировать прямоугольными элементами.

Именно таким редактором и является редактор табличных документов, входящий в систему «1С:Предприятие». В процессе создания макета можно вводить в ячейки табличного документа разнообразный текст; задавать параметры форматирования как тексту, так и ячейке в целом; изменять высоту строк и ширину колонок табличного документа; включать в макет рисованные элементы – линии и прямоугольники, а также другие графические объекты: картинки, OLE-объекты и диаграммы, различные элементы управления; определять оформление как всего табличного документа, так и отдельных ячеек или групп ячеек.

В окончательном виде макет представляет собой совокупность прямоугольных областей, каждая из которых служит для выдачи какой-то части готового документа или отчета: область для выдачи заголовочной части (наименования, даты и т. п.); область для выдачи шапки, табличной части и так далее.

The screenshot shows a window titled 'С:\ТребованиеНакладная.mxl - ... (1С:Предприятие)'. The main area displays a table layout with 9 columns and 16 rows. The layout includes a header row, a section for organizational data, a data table with columns for '№', 'Код', 'Материал', 'Мест', and 'Количество', and a signature section at the bottom.

Заголовок	1	2	3	4	5	6	7	8	9	
	1									
	2									
	3	<ТекстЗаголовка>								
	4									
	5	Организация:	<Организация>							
	6									
	7	Склад:	<Склад>							
	8	Подразделение:	<Подразделение>							
	9									
	10	№	Код	Материал			Мест	Количество		
	11									
Строка	12	№мПП>	<Код>	<Имя>			количествоМес	ИзмМес	<Количество>	ЕДИзм>
	13									
	14									
Подписи	15									
	16	Отпустил				Получил				

Рис. 263. Макет в табличном документе

Любой прямоугольной области макета можно присвоить уникальное имя и в дальнейшем ссылаться на поименованную область.

Сам процесс построения отчета на основе макета происходит следующим образом.

Вначале документ или отчет строится как пустая таблица. Необходимо иметь в виду, что макет не связан напрямую с готовым отчетом. Скорее макет представляет собой некий конструктор, набор кирпичиков, из которых в процессе работы программного модуля складывается готовый отчет.

В процессе обработки алгоритма отчета из макета в нужном порядке извлекаются поименованные области, транслируются (вместо имен переменных подставляются их значения) и копируются в таблицу готового отчета. Средства языка формирования отчетов позволяют наращивать отчет как вниз, так и вправо.

Другой способ построения отчета заключается в том, что пользователь вводит данные в предназначенные для этого ячейки табличного документа, после чего эти данные обрабатываются программой по заданным в самом табличном документе отчета формулам. Результаты появляются в других ячейках табличного документа, после чего отчет готов для просмотра и распечатки.

3.23. Настройка печати

Для просмотра результатов некоторых установок, описываемых в этом разделе, рекомендуем установить режим просмотра страниц (пункт [Таблица – Вид – Режим просмотра страниц](#)).

3.24. Колонтитулы

Колонтитулы – это области сверху и внизу страницы, служащие для вывода какого-либо повторяющегося текста на каждой странице документа: номера страницы, названия документа, даты и времени создания, другой информации.

Для создания колонтитула используется пункт [Таблица – Настройки печати – Колонтитулы...](#) При выборе этого пункта меню на экран выдается диалог, в котором следует выбрать, какой колонтитул необходимо создать – верхний или нижний, указать текст и стиль колонтитулов.

Для каждого колонтитула указывается, с какой страницы начинать печатать колонтитул, как его расположить и каким шрифтом печатать.

3.25. Разбиение на страницы

При печати табличных документов большого размера редактор табличного документа системы «1С:Предприятие» выполняет автоматическое разбиение документа на страницы в соответствии с установками параметров страницы. Кроме того, при разбиении на страницы учитываются принудительные переводы страницы и установки положения на странице.

Установки положения на странице не переносятся из табличного документа в готовый документ.

Принудительный перевод страницы. Для принудительной расстановки переводов страниц можно воспользоваться пунктом [Таблица – Настройки печати – Вставить разрыв страницы](#).

Для вставки горизонтального перевода страницы следует выделить ячейку или строку табличного документа и выбрать пункт [Таблица – Настройки печати – Вставить разрыв страницы](#). Перевод страницы будет вставлен над выделенной ячейкой или строкой. Он будет обозначен пунктирной линией.

Чтобы убрать горизонтальный перевод страницы, необходимо выделить ячейку или строку под переводом страницы и выбрать пункт [Таблица – Настройки печати – Удалить разрыв страницы](#).

Для вставки вертикального перевода страницы следует выделить колонку табличного документа и выбрать пункт [Вставить разрыв страницы](#). Перевод страницы будет вставлен слева от выделенной колонки, он будет обозначен пунктирной линией.

Чтобы убрать перевод страницы по вертикали, необходимо выделить колонку слева от перевода страницы и выбрать пункт [Таблица – Настройки печати – Убрать перевод страницы](#).

Положение на странице. Для любой строки или колонки табличного документа (а также диапазона строк или колонок) могут быть заданы параметры размещения на странице. Для этого необходимо выделить соответствующий диапазон и в категории [Параграф](#) палитры свойств ([Ячейки](#)) установить свойство [С новой страницы](#) или [Вместе со следующим](#).

Для изменения положения на странице следует вновь выделить соответствующий диапазон и снять свойство [С новой страницы](#) или [Вместе со следующим](#).

3.26. Автоматическое повторение строк и колонок

Для любой строки и колонки табличного документа (а также для диапазона строк и колонок) можно задать повторение при печати на каждой странице. Для этого необходимо выделить строку или диапазон строк (колонка или диапазон колонок) и выбрать пункт [Таблица – Настройки печати – Повторять на каждом листе](#).

Диапазон, который выбран для повторения, на первом листе будет размещен так, как он расположен в табличном документе, то есть в соответствии со своим положением на листе относительно заголовков строк и колонок. На последующих листах повторяющиеся строки будут располагаться сверху листа, а колонки – слева.

Убрать повторение на каждом листе можно, выбрав пункт [Таблица – Настройки печати – Удалить повторение](#). Настройка будет снята со всех диапазонов, для которых оно было установлено ранее.

Установки автоматического повторения строк и колонок не переносятся из табличного в готовый документ.

Кроме того, области повторения можно просмотреть и изменить в палитре свойств табличного документа.

3.27. Задание области печати

Вместо печати всего табличного документа целиком редактор позволяет задать область печати – диапазон, который будет напечатан.

Для того чтобы задать область печати, следует выделить диапазон в табличном документе и выбрать пункт [Таблица – Настройки печати – Задать область печати](#).

Отменить печать диапазона можно с помощью пункта [Таблица – Настройки печати – Убрать область печати](#).

Для табличного документа область печати может быть также указана в свойствах палитры свойств табличного документа.

3.28. Настройка параметров страницы

Режим настройки параметров страницы позволяет выбрать печатающее устройство и установить размер и ориентацию бумаги, на которой будет производиться печать табличного документа. Для вызова этого режима нужно выбрать пункт [Файл – Параметры страницы](#). На экран будет выдан диалог [Параметры страницы](#).

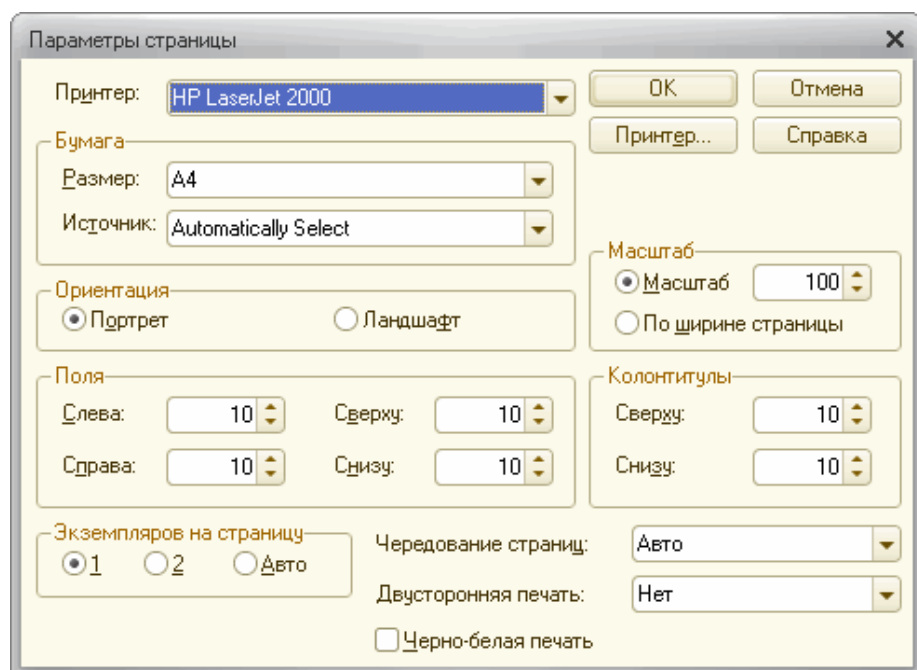


Рис. 264. Диалог настройки параметров страницы

Назначение управляющих элементов диалога в большинстве своем стандартно для многих программ Microsoft Windows. Пояснения требуют те особенности, которые касаются редактора табличного документа системы «1С:Предприятие».

Экземпляров на страницу. Группа [Экземпляров на страницу](#) позволяет выбрать количество экземпляров документа, которые при печати будут помещаться на одну страницу (экземпляры помещаются один под другим). Если выбрана опция [Авто](#), то программа сама будет определять количество экземпляров: если при выбранном размере листа и полях печати на странице помещаются два экземпляра документа целиком, то будет напечатано два экземпляра, а если не помещаются, то один.

Колонтитулы. Группа [Колонтитулы](#) позволяет задать размеры области листа, которая будет отводиться для печати колонтитулов. Размеры задаются в миллиметрах и отсчитываются от верхнего или нижнего поля в сторону соответствующего края листа.

Масштаб. Поле [Масштаб](#) позволяет задать масштаб печати табличного документа. Установленный масштаб не влияет на масштаб просмотра документа в редакторе.

По ширине страницы. Если включен данный режим, редактор табличного документа автоматически подбирает масштаб таким образом, чтобы документ при печати умещался по ширине на один лист.

Чередование страниц. Поле [Чередование страниц](#) позволяет установить при печати чередование четных и нечетных страниц:

- **Не использовать** – поля печатаются как обычно;
- **Зеркально слева** – нечетные страницы печатаются как обычно, а четные печатаются с переменной местами левого и правого поля границ;
- **Зеркально сверху** – нечетные страницы печатаются как обычно, а четные страницы печатаются с переменной местами нижнего и верхнего полей границ;
- **Авто** – если в поле **Двусторонняя печать** указано **Нет**, поля выводятся как обычно; если **Переворот влево**, то используется вывод со значением **Зеркально слева**; если **Переворот вверх**, то используется вывод со значением **Зеркально сверху**.

Двусторонняя печать. Поле позволяет настроить двустороннюю печать страниц:

- **Нет** – двусторонняя печать не используется;
- **Переворот влево** – двусторонняя печать, при которой страницу нужно переворачивать влево;
- **Переворот вверх** – двусторонняя печать, при которой страницу нужно переворачивать вверх.

Черно-белая печать. Если включен данный режим, цвета, заданные для табличного документа, игнорируются, и документ выводится на печать в черно-белом режиме.

3.29. Печать табличного документа

Просмотр документа перед печатью. Перед печатью можно просмотреть документ в том виде, как он будет напечатан. Для этого следует выбрать пункт **Файл – Предварительный просмотр**.

При просмотре можно использовать кнопки **След.** и **Пред.** для вывода на экран следующей и предыдущей страницы просматриваемого документа и кнопку **2 Стр./1 Стр.** для переключения в режим одновременного просмотра двух страниц документа и обратно.

Для изменения масштаба изображения можно использовать мышь или кнопки **Развернуть** и **Свернуть**.

Если указатель мыши находится в пределах просматриваемого документа, нажатие левой кнопки дискретно увеличивает масштаб изображения. Всего существует 3 фиксированных масштаба просмотра. При достижении самого крупного с помощью нажатия левой кнопки мыши можно вновь перейти к просмотру документа в самом мелком масштабе.

Нажатие кнопки **Развернуть** увеличивает масштаб на 1 шаг, нажатие кнопки **Свернуть** – уменьшает.

Для выхода из режима предварительного просмотра следует нажать кнопку **Заккрыть**.

При показе документа будут использоваться параметры печати, установленные в режиме настройки параметров страницы. Это размер бумаги, границы области печати, масштаб и др. Будут использоваться и параметры размещения строк и колонок, установленные в табличном редакторе. Это принудительное разбиение на страницы, повторение строк и колонок, параметры размещения строк и колонок, установленные в режиме **Положение на странице**, и др.

Печать документа. Для вывода документа на принтер нужно выбрать пункт **Файл – Печать**. На экран будет выдан стандартный диалог для настройки параметров печати.

В этом диалоге следует установить необходимые параметры печати – выбрать принтер, диапазон страниц для печати, количество копий – и нажать кнопку **ОК** для выполнения печати.

ПРИМЕЧАНИЕ. Обратите внимание, что в зависимости от настроек конфигурации масштаб печати табличного документа может влиять или не влиять на размер полей страницы.

3.30. Редактирование свойств табличного документа и ячеек

Настройка поведения табличного документа и поведения ячейки или группы ячеек производится в палитре свойств.

В свойствах табличного документа определяют направление перехода при нажатии клавиши **Enter**, задание области печати, области повторяющихся частей документа и т. д.

В свойствах ячейки или группы ячеек определяется тип информации, способ и формат представления этой информации в ячейках, размеры ячеек и другие параметры.

При открытии или создании табличного документа палитра свойств показывает свойства ячеек. Чтобы перейти к просмотру и редактированию свойств табличного документа, в верхней строке палитры свойств в списке нужно выбрать [Табличный документ](#). Для просмотра и редактирования свойств ячейки или группы ячеек в верхней строке палитры свойств в списке выбрать [Ячейки](#).

3.30.1. Свойства табличного документа

Настройка производится в палитре свойств табличного документа. Для их показа в верхней строке палитры свойств в списке нужно выбрать [Табличный документ](#).

Макет – если свойство установлено, то данный табличный документ является макетом.

Итоги по горизонтали – определяется положение итоговой строки для горизонтальной группы (см. раздел [«Удаление именованных областей»](#)).

Итоги по вертикали – определяется положение итоговой колонки для вертикальной группы (см. раздел [«Удаление именованных областей»](#)).

Картинка – выбор картинки фона (см. раздел [«Установка изображения в качестве фона листа»](#) [здесь](#)).

Фиксированный фон – если свойство установлено и выбрана фоновая картинка, то при прокрутке табличного документа фон остается неподвижен, в противном случае фон прокручивается вместе с табличным документом.

Переход после ввода – выбирается направление перехода после нажатия клавиши **Enter**. Возможные варианты выбора: **Нет** – не переходить; **Вниз** – переходить на ячейку, расположенную под текущей; **Вправо** – переходить на ячейку, расположенную справа от текущей.

Область печати – определяется область табличного документа, которая будет печататься (см. раздел [«Определение области печати»](#)).

Повторяющиеся строки – определяются строки табличного документа, которые будут печататься на каждом новом листе (см. раздел [«Определение области печати»](#)).

Повторяющиеся колонки – определяются колонки табличного документа, которые будут печататься на каждом новом листе (см. раздел [«Определение повторяющихся строк и колонок»](#)).

Для групп и заголовков можно осуществить настройку их оформления: выбрать цвет текста и фона.

Если созданный табличный документ предполагается использовать в дальнейшем, то установка свойства [Сохранять настройки](#) позволяет запомнить в документе заданные настройки отображения (отображения заголовков, сетки, групп, имен, позиции фиксации).

Свойства [Итоги по горизонтали](#) и [Итоги по вертикали](#) определяют те строки и колонки, которые логически относятся к группе, но должны оставаться при сворачивании показа самой группы.

Рассмотрим пример расчетов с использованием установки свойства [Итоги по вертикали](#) (выбрано расположение [Сверху](#)). Исходная таблица данных показана на рисунке.

1	2	3	4	5
1	Подразделение	Сотрудник	Период	Расчеты
2	Отделение 07	Всего по отделению	за I кв.	109009
3	Отдел 1	Итого по отделу	за I кв.	60866
4		Иванов	за I кв.	43000
5			Январь	16000
6			Февраль	14000
7			Март	13000
8		Петров	за I кв.	17866
9			Январь	3566
10			Февраль	6500
11			Март	7800
12	Отдел 2	Итого по отделу	за I кв.	48143
13		Сидоров	за I кв.	48143
14			Январь	16733
15			Февраль	15766
16			Март	15644

Рис. 265. Исходная таблица данных

При нажатии на маркер первого уровня таблица преобразуется к следующему виду.

1	2	3	4	5
1	Подразделение	Сотрудник	Период	Расчеты
2	Отделение 07	Всего по отделению	за I кв.	109009
17				
18				
19				

Рис. 266. Данные первого уровня

Получаем итоговые данные по отделению.

При нажатии на маркер второго уровня таблица преобразуется к следующему виду.

1	2	3	4	5
1	Подразделение	Сотрудник	Период	Расчеты
2	Отделение 07	Всего по отделению	за I кв.	109009
3	Отдел 1	Итого по отделу	за I кв.	60866
12	Отдел 2	Итого по отделу	за I кв.	48143
17				
18				

Рис. 267. Данные второго уровня

Получаем итоговые данные по отделам.

При нажатии на маркер третьего уровня таблица преобразуется к следующему виду.

1	2	3	4	5
1	Подразделение	Сотрудник	Период	Расчеты
2	Отделение 07	Всего по отделению	за I кв.	109009
3	Отдел 1	Итого по отделу	за I кв.	60866
4		Иванов	за I кв.	43000
8		Петров	за I кв.	17866
12	Отдел 2	Итого по отделу	за I кв.	48143
13		Сидоров	за I кв.	48143
17				

Рис. 268. Табличный документ. Нажали маркер третьего уровня

Получаем итоговые данные по сотрудникам.

3.30.2. Установка изображения в качестве фона листа

Для оформления табличного документа можно в качестве фона использовать рисунок, который выбирается из библиотеки картинок или из внешнего файла.

Для выбора фоновой картинке в палитре свойств табличного документа в свойстве [Картинка](#) следует нажать кнопку выбора.



Рис. 269. Кнопка выбора в свойстве «Картинка»

На экран выводится окно выбора картинки.

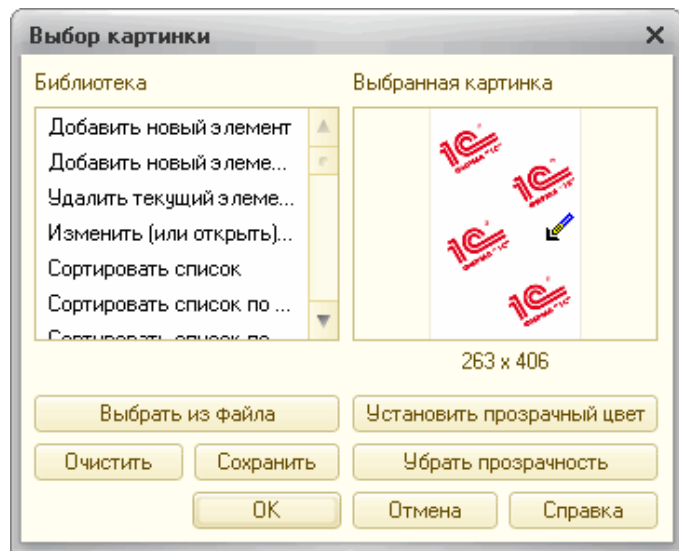


Рис. 270. Выбор фоновой картинки для табличного документа

Нужно выбрать картинку из библиотеки или из файла. В выбранной картинке можно установить прозрачным (убрать) любой из имеющихся на картинке цветов. Для этого следует щелкнуть кнопку [Установить прозрачный цвет](#). Указатель мыши изменит свой вид. Повернуть указатель к цвету картинки, который нужно сделать прозрачным, и щелкнуть левой кнопкой мыши. Установленную прозрачность можно отменить, для этого следует нажать кнопку [Убрать прозрачность](#).

Для установки фона следует нажать кнопку [ОК](#).

	1	2	3	4
1	Отчет по курсам валют			
2	курс валюты: "Доллар США"			
3		Дата	Значение	
4		18.04.2009	33,4677	
5		17.04.2009	33,4184	
6		16.04.2009	33,4507	
7		15.04.2009	33,3887	
8		14.04.2009	33,4863	
9		11.04.2009	33,6309	
10		10.04.2009	33,5334	
11		09.04.2009	33,7781	
12		08.04.2009	33,3840	
13		07.04.2009	33,1743	
14		04.04.2009	33,4095	
15		03.04.2009	33,7636	
19		28.03.2009	33,4133	
20		27.03.2009	33,4668	
21		26.03.2009	33,7268	
22		25.03.2009	33,2726	
23		24.03.2009	33,3034	
24		21.03.2009	33,4230	
25		20.03.2009	33,8222	
26		19.03.2009	34,4203	
27		18.03.2009	34,5318	

Рис. 271. Табличный документ с установленным фоном

Чтобы убрать фоновую картинку из табличного документа, в окне фоновой картинки нужно нажать кнопку [Очистить](#).

В палитре свойств при установленном свойстве [Фиксированный фон](#) выбранный фон остается неподвижным при прокручивании табличного документа.

3.30.3. Определение области печати

Если требуется, чтобы печатался не весь табличный документ, а только его часть (последовательно расположенные строки, или колонки, или группа ячеек), то в свойстве [Область печати](#) следует указать область печати.

Область печати может быть указана явно (в формате диапазона, например, R5C7:R8C15 для диапазона ячеек в виде прямоугольника или R3:R9 для диапазона строк с третьей по девятую, D12 для рисунка) или именами (заранее присвоенными областям). Если используются имена, то после ввода имя области автоматически заменится указанием диапазона.

Область печати может быть указана с помощью диалога [Диапазон](#), вызываемого кнопкой выбора в свойства [Область печати](#).

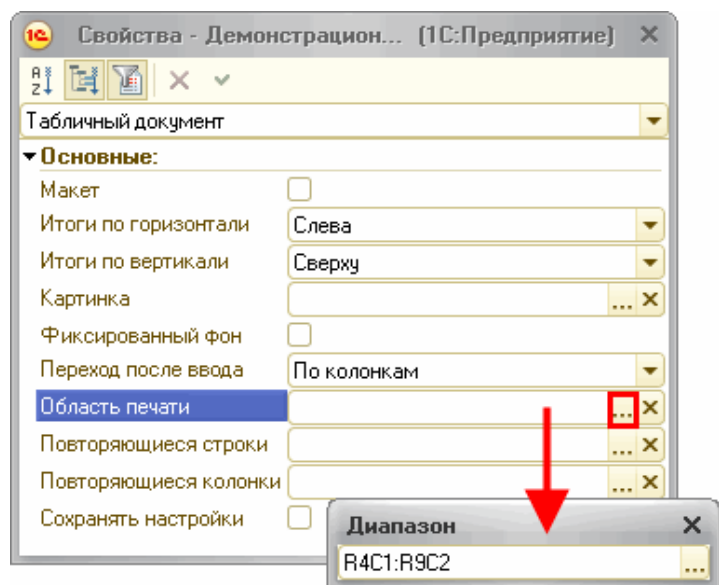


Рис. 272. Указание области печати для табличного документа

Когда диалог вызван, то с помощью мыши или клавиатуры (см. раздел «[Имена](#)» [здесь](#)) нужно указать требуемый диапазон, затем нажать кнопку выбора диалога [Диапазон](#). Выбранный диапазон будет указан в свойстве [Область печати](#).

Область печати может быть также указана с помощью команд меню (см. раздел «[Задание области печати](#)», [здесь](#)).

3.30.4. Определение повторяющихся строк и колонок

При разработке различных отчетов следует учитывать, что табличный документ может не поместиться на один лист целиком. В этом случае желательно, чтобы на каждом следующем листе повторялась шапка табличного документа (для длинного табличного документа) или боковик (для широкого).

Редактор табличных документов системы «1С:Предприятие» позволяет легко настраивать просмотр табличного документа. Для этих целей используется механизм определения повторяющихся строк и колонок.

Область повторяющихся строк указывается в свойстве [Повторяющиеся строки](#), колонок – в свойстве [Повторяющиеся колонки](#).

Способ указания диапазона подробно описан в разделе «[Определение области печати](#)».

3.31. Свойства ячеек табличного документа

В табличном документе можно настроить поведение каждой отдельной ячейки или группы ячеек.

Настройка производится в палитре свойств ячеек. Для просмотра и редактирования свойств ячейки или группы ячеек в верхней строке палитры свойств в списке нужно выбрать [Ячейки](#).

Набор свойств, которые открыты для просмотра и редактирования, относится ко всем выделенным ячейкам табличного документа. Если установить значение некоторого свойства, то это значение будет установлено для всей выделенной области.

3.31.1. Категория свойств «Основные»

Текст – текст, размещенный в ячейке. Свойство показывается, если в категории [Макет](#) не установлено свойство [Значение](#) и в свойстве [Заполнение](#) категории [Макет](#) выбрано значение [Текст](#) или [Шаблон](#).

Размещение – пункты свойства [Размещение](#) позволяют установить, как показывать текст, если он не помещается в ячейке целиком (см. таблицу).

Расположение текста	Пояснение
Авто	Не помещающийся по ширине текст будет показан в соседних по горизонтали ячейках, если в них, в свою очередь, нет текста
Обрезать	Видимая часть текста будет ограничена левой и правой границами ячейки
Забивать	Видимая часть текста будет ограничена левой и правой границами ячейки, при этом все символы видимой части текста будут заменены символами #. (Применяется в том случае, когда обрезанный текст может восприниматься ошибочно. Пример: число)
Переносить	Не помещающийся по ширине текст будет переноситься на следующие строки ячейки

Имя – имя области.

Защита – если установлено, то выбранные ячейки и рисунки защищены от изменения (если установлено свойство табличного документа [Только просмотр](#)).

Гиперссылка – если свойство установлено, то для данной ячейки устанавливается режим обработки расшифровки по одинарному щелчку мыши. Для табличного документа, размещенного в форме, вызывается процедура-обработчик события [Выбор](#), и, если флажок стандартной обработки установлен, запускается процедура-обработчик события [Обработка расшифровки](#). Если табличный документ используется самостоятельно (например, как результат выполнения отчета), то вызывается стандартная обработка (например, если в ячейке содержится ссылка на элемент списка [Номенклатура](#), то будет открыта форма элемента). Подробнее см. ниже описание свойства [Параметр расшифровки](#).

3.31.2. Категория свойств «Значения»

Формат – задает формат содержимого ячейки. При нажатии кнопки выбора вызывает **Конструктор форматной строки**, с помощью которого устанавливается формат.

Маска – задает маску, определяющую ввод символов в строке ячейки табличного документа.

3.31.3. Категория свойств «Положение»

По горизонтали – задает выравнивание текста в ячейке табличного документа по горизонтали.

Лево	Выравнивание текста по левой границе ячейки
Право	Выравнивание текста по правой границе ячейки
Центр	Текст размещается вдоль вертикальной линии, проходящей через центр ячейки
По ширине	Текст выравнивается одновременно по правой и по левой границам ячейки; ячейки автоматически устанавливаются такими, чтобы текст в ячейке был виден полностью
Авто	В зависимости от типа содержимого ячейки. Например, для строк будет выбран тип выравнивания Лево , для чисел – Право

По выделенным колонкам – если установлено, то текст, введенный в самой левой ячейке выделенной в данный момент области, будет расположен в соответствии с выбранным вариантом горизонтального положения в пределах всей выделенной области. Этот режим используется прежде всего для того, чтобы отцентрировать текст в пределах нескольких ячеек в строке.

По вертикали – задает выравнивание текста в ячейке табличного документа по вертикали.

Низ	Текст располагается вдоль нижней границы ячейки
Верх	Текст располагается вдоль верхней границы ячейки
Центр	Текст размещается вдоль горизонтальной линии, проходящей через центр ячейки

Ориентация – определяет угол наклона текста в ячейке (изменяется от -360° до 360°).

Отступ – размер отступа от края ячейки, по которому произведено выравнивание текста в ячейке.

Автоотступ – если указано значение больше 0, то при использовании группировок отступ будет определяться как произведение уровня группировки и значения, указанного в свойстве.

Свойства **Отступ** и **Автоотступ** не исключают друг друга.

3.31.4. Категория свойств «Оформление»

В категории свойств определяется шрифт текста (гарнитура, размер, стиль, цвет), выделение отрицательных чисел, цвет фона и узора, границы ячейки (указывается необходимость показа каждой границы и выбирается стиль линии).

Для установки оформления в свойстве границы следует нажать кнопку выбора. На экран выводится диалог выбора линии.

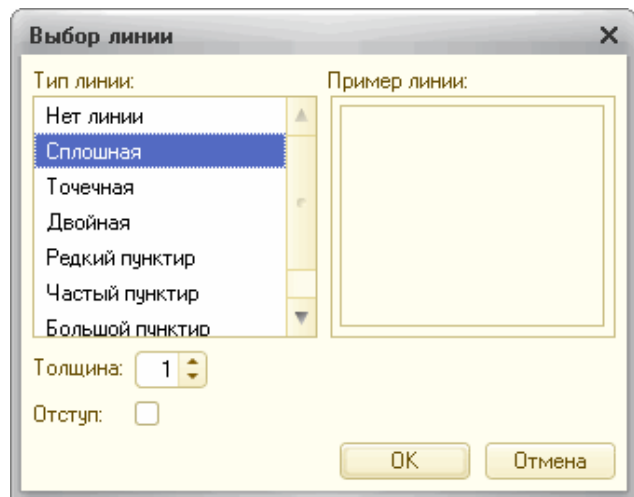


Рис. 273. Диалог выбора линии

Линию можно установить, выбрав ее тип и толщину.

Пример выбора показывается в диалоге.

Если требуется, чтобы линия выводилась не по всей стороне области ячеек, то нужно установить флажок **Отступ**. В этом случае линия будет отображаться с отступами от краев ячеек. Например, так:

2	3	4	5
---	---	---	---

Рис. 274. Нижняя граница ячеек с отступом

3.31.5. Категория свойств «Картинка»

В категории свойств определяются свойства картинки, расположенной в ячейке.

Картинка – Для выбора картинки в поле свойства следует нажать кнопку выбора. На экран выводится диалог выбора картинки.

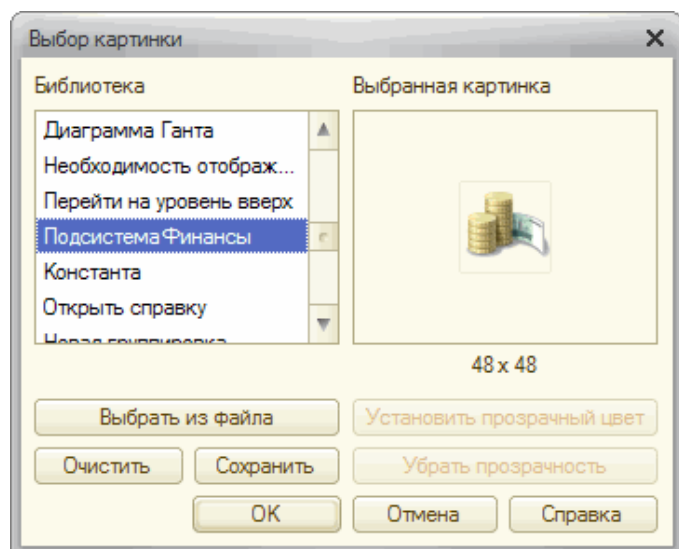


Рис. 275. Выбор картинки для ячейки табличного документа

В данном диалоге следует выбрать картинку из библиотека картинок или выбрать картинку из файла.

Размер картинки – если установлено значение **Черепица** или **Растянуть**, то текст отображается поверх картинки. По умолчанию картинке устанавливается размер **Пропорционально**.

Положение текста относительно картинки – с помощью данного свойства устанавливается выравнивание текста относительно картинки. Если установлено значение **Авто**, тогда:

- Если картинка выровнена **влево**, текст отображается справа от картинки.
- Если картинка выровнена **вправо**, текст отображается слева от картинки.
- Если картинка выровнена **вверх**, текст отображается снизу от картинки.
- Если картинка выровнена **вниз**, текст отображается сверху от картинки.
- Иначе текст отображается поверх картинки.

Приложение 4. Редактор HTML-документов

4.1. Назначение

С помощью HTML-редактора создаются гипертекстовые документы (HTML-документы).

4.2. Создание HTML-документа

Для создания нового HTML-документа нужно выбрать пункт [Файл – Новый](#). В открывшемся диалоге выбора вида документа выбрать [HTML-документ](#) и нажать кнопку [ОК](#). Программа создаст пустой документ и выведет на экран окно для его редактирования.

На закладке [Редактирование](#) производится ввод и редактирование документа в визуальном режиме, на закладке [Текст](#) можно редактировать создаваемый документ в формате HTML. На закладке [Просмотр](#) просматривают результат редактирования.

4.3. Сохранение

HTML-документ можно сохранить в файл для того, чтобы его в дальнейшем можно было бы открыть, просмотреть и отредактировать. Чтобы сохранить HTML-документ в виде файла нужно выбрать пункт [Файл – Сохранить](#).

В диалоге сохранения файла следует выбрать каталог и указать имя файла. Затем нажать кнопку [Сохранить](#).

4.4. Редактирование

Создание HTML-документа напоминает работу с обычным текстовым редактором.

Однако имеются некоторые особенности. Так, например, в документ можно вставлять картинки, HTML-таблицы (далее будет использоваться термин таблица), надписи, а также размещать ссылки на другие документы.

4.5. Ввод текста

Ввод текста подробно описан в разделе «Текстовый редактор». Поэтому в текущем разделе будут указаны только особенности ввода текста при создании HTML-документа.

4.6. Форматирование текста

Текст HTML-документа можно форматировать, т. е. устанавливать его стиль (фонт, размер, наклон, отступы, выравнивание и т. д.).

При копировании текста из других текстовых редакторов, например, Microsoft Office Word, сохраняется исходное форматирование текста.

4.7. Вставка и работа с таблицей

В HTML-документ можно вставлять таблицы. Для вставки таблицы нужно выбрать пункт [Таблица – Вставить таблицу](#). В открывшемся диалоге указать количество строк и колонок.

Если в дальнейшем в процессе редактирования выяснится, что указанного количества строк или колонок не хватает, их можно добавить.

Для работы с таблицей (вставка, удаление строк, колонок и ячеек) предназначены команды, выполняемые при выборе пунктов меню [Таблица](#) главного меню.

Пункт меню	Действия
------------	----------

«Таблица»	
Вставить строку	Сверху от текущей строки добавляется новая строка
Удалить строку	Удаляется текущая строка
Вставить колонку	Справа от текущей колонки добавляется новая колонка
Удалить колонку	Удаляется текущая колонка
Вставить ячейку	В таблицу слева от текущей ячейки добавляется ячейка. При этом все ячейки справа сдвигаются
Удалить ячейку	Удаляет текущую ячейку. Все ячейки справа сдвигаются влево
Объединить ячейки	Объединяет две ячейки: текущую и ячейку, расположенную справа от текущей. При этом введенная информация в этих ячейках также объединяется. Если требуется объединить больше чем две ячейки, нужно выполнить эту операцию несколько раз
Разбить ячейку	Разделяет текущую ячейку на две ячейки

4.8. Вставка картинки

В HTML-документ можно вставлять картинки.

Для вставки картинки нужно выбрать пункт **Элементы – Картинка**. В открывшемся диалоге указать адрес картинки (путь и имя файла картинки) или выбрать файл в стандартном диалоге выбора файла. В поле **Текст** ввести поясняющий текст, который будет появляться в режиме просмотра в виде ярлыка при помещении указателя мыши над картинкой.

В группе реквизитов **Расположение** указать, если требуется, выравнивание картинки (как располагается картинка в документе).

Значение выравнивания	Действия
Не задано	Картинка располагается в тексте
Лево	Картинка располагается в новой строке, прижата к левой границе документа
Право	Картинка располагается в новой строке, прижата к правой границе документа
Поверх текста	Картинка располагается в тексте с выравниванием по центру картинки
Низ	Картинка располагается в тексте с выравниванием текста по нижней границе картинки
Центр	Картинка располагается в тексте с выравниванием текста по центру картинки
Верх	Картинка располагается в тексте с выравниванием текста по верхней границе картинки

Если требуется поместить картинку в рамку, в реквизите диалога **Рамка** нужно указать толщину рамки. Если толщина не указана или равна 0, то рамка не рисуется.

Для вставки картинки следует нажать кнопку **ОК**.

Указанные в диалоге реквизиты можно изменить в панели свойств картинки.

Вставленную картинку можно переместить. Для этого нужно указать картинку и перетащить ее мышью в нужное место документа. При этом свойство картинки **Положение** не меняется, т. е. если установлено не **Лево** и не **Право**, то картинку можно расположить в тексте, в противном случае картинка будет

располагаться в новой строке.

Картинку можно копировать. Копирование осуществляется стандартным образом (с помощью буфера обмена или с помощью мыши при нажатой клавише **Ctrl**).

Картинку можно расположить в любой части HTML-документа (текст, таблица и надпись).

4.9. Надпись

Надпись – специальный объект HTML-документа. Он представляет собой область, в которую можно поместить текст, таблицу, картинку или другую надпись. При просмотре надписи вложенные объекты непрерывно прокручиваются справа налево.

Для вставки линии нужно выбрать пункт **Элементы – Надпись**.

Размер надписи можно изменить. Для этого нужно выделить надпись и с помощью мыши переместить маркер надписи.

Надписи можно располагать в тексте и в таблице.

4.10. Ссылки

Для создания гипертекста в HTML-документе можно создавать ссылки.

Нужно выделить объект (текст или картинку), выбрать пункт **Элементы – Ссылка**. На экран выводится диалог формирования ссылки.

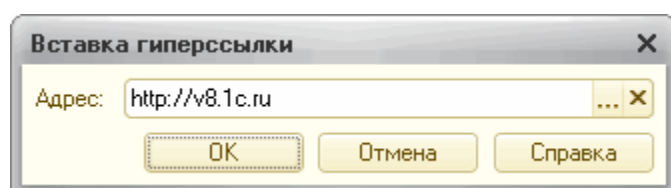


Рис. 276. Диалог формирования ссылки

В реквизите **Адрес** указать адрес гиперссылки или имя закладки. Для создания ссылки следует нажать кнопку **OK**.

Выделенный объект форматируется в соответствии со стилем гиперссылки. Обычно это синий цвет текста с подчеркиванием. Картинка помещается в рамку синего цвета.

В режиме просмотра при подведении указателя мыши к объекту, содержащему ссылку, указатель меняет форму на форму «руки».

4.11. Закладки

Для создания внутренней ссылки в HTML-документе можно создавать закладки.

Нужно выделить объект (текст или картинку), в главном меню выбрать пункт **Элементы – Закладка**. На экран выводится диалог формирования закладки.

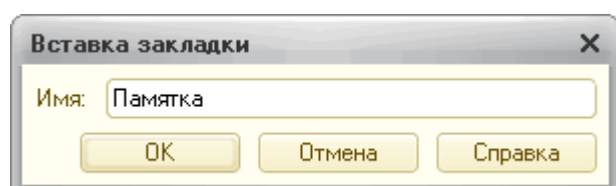


Рис. 277. HTML-документ. Вставка закладки

Выделенный объект при этом не форматируется.

Закладки используются для организации внутренней ссылки следующим образом:

- создается ссылка (см. выше);
- в поле [Адрес](#) после указания гиперссылки записывается символ <#> и наименование закладки. Например: [v8help://1cv8/PictInfo#Закладка1](#).

4.12. Линия

С помощью линий производится зрительное разделение документа.

Для создания линии нужно установить указатель в строку, в которую будет вставлена линия, и выполнить команду [Элементы – Линия](#).

4.13. Форматирование HTML-документа

Для оформления HTML-документа используются средства, предоставляемые командами панелей инструментов [Форматирование](#) и [Редактирование HTML](#).

С помощью панели инструментов [Редактирование HTML](#) для введенного текста можно выбрать стиль и сместить текст.

С помощью панели инструментов [Форматирование](#) выбирают параметры шрифта, цвет текста и фона, производят выбор вида рамок и положение текста.

4.14. Редактирование в HTML-формате

Возможности редактора HTML-документов, предоставляемые на закладке [Редактирование](#), ограничены. На закладке [Текст](#) можно отредактировать документ в формате HTML-кода.

4.15. Просмотр результата

Для просмотра результата редактирования HTML-документа нужно перейти на закладку [Просмотр](#) и проверить работу объектов документа.

Приложение 5. Редактор графической схемы

Графическая схема – композиция элементов схемы, отражающая структуру или отношения объектов. Ее можно использовать для рисования структурных схем, бизнес-схем, схем потоков данных, схем взаимодействия объектов.

Графическая схема представляет собой прямоугольную область экрана, которая в самом общем случае содержит различные элементы схемы, например, декорации, соединительные линии или точки маршрута.

5.1. Редактирование схемы

Редактирование схемы можно осуществлять в поле графической схемы на форме или в документе [Графическая схема](#).

Документы этого типа сохраняются в файле с расширением GRS.

5.2. Вставка элементов в графическую схему

Для вставки элемента нужно выбрать его тип в командной панели поля графической схемы (команда меню [Вставка](#)) или выполнить соответствующую команду меню [Графическая схема – Вставка элементов схемы](#). После этого указать прямоугольную область для вставки элемента или начальную и конечную точки в случае вставки декоративной линии. После вставки элемента режим вставки сбрасывается аналогично редактору форм.

Каждый размещаемый элемент имеет прямоугольную рамку выделения с маркерами в виде небольших квадратов по углам прямоугольника и на серединах сторон.

Точное позиционирование и задание размеров элемента производятся в категории [Расположение](#) палитры свойств.

Перемещение и изменение размеров элементов производятся стандартными для Microsoft Windows способами при помощи мыши или стрелок клавиатуры.

Настройка элемента (указание элементу правил поведения, оформления и других свойств) производится в палитре свойств элемента.

5.2.1. Копирование элементов графической схемы

Новые элементы можно вставить путем копирования их из любой схемы (в том числе и карты маршрута) и из любой конфигурации.

Для копирования нужно выделить любым способом элемент или группу элементов, поместить их в буфер обмена, затем в текущей схеме выполнить вставку из буфера обмена. Если имя вставленного элемента уникально, то оно не меняется. В противном случае новому элементу будет присвоено новое имя.

При копировании элемента или группы элементов с использованием буфера обмена запоминаются также имена элементов, которые можно вставить в любой текст.

Скопировать элемент или группу элементов можно также путем перетаскивания с нажатой клавишей [Ctrl](#).

Если требуется скопировать изображение части или всей графической схемы в объект графического редактора, то нужно выделить необходимые элементы и выполнить команду копирования. Если перед копированием не выделено ни одного элемента, то в буфере обмена сохраняется вся графическая схема.

5.3. Разметочная сетка

Одним из простых приемов выравнивания элементов является выравнивание их границ по разметочной сетке. Она представляет собой сетку из горизонтальных и вертикальных линий, ориентируясь на которые можно задавать точное положение элементов в окне графической схемы и устанавливать их

размеры. Вы можете управлять шагом разметочной сетки, включать и выключать привязку элементов к сетке, а также показывать сетку или скрывать ее.

При создании новой схемы размер шага разметочной сетки (по умолчанию) равен 20 точкам экрана (и по горизонтали, и по вертикали), а режим показа сетки – [Рисовать линии](#).

При перетаскивании элемента выравнивание всегда производится по верхнему левому углу. Если размер элемента не был кратным шагу сетки, то при перетаскивании не производится изменение его размера.

Управление разметочной сеткой производится в категории свойств графической схемы [Редактирование](#). Если для схемы установлено свойство [Выравнивать по сетке](#), разметка изображается точками, расположенными в местах пересечения линий разметки (можно использовать режим показа точек сетки в шахматном порядке), либо линиями (в зависимости от свойства [Режим рисования сетки](#)).

Если свойство [Выравнивать по сетке](#) установлено, все располагаемые элементы будут автоматически размещаться вдоль линий разметки. Размер вставляемых элементов также будет автоматически устанавливаться кратным шагу сетки.

При изменении элемента изменяемая граница всегда размещается точно по разметочной сетке, несмотря на то, выровнена ли по разметочной сетке противоположная граница элемента или нет.

Можно задать шаг сетки – размер ячеек сетки по горизонтали и вертикали. Для этого в свойствах схемы [Гор. шаг сетки](#) или [Верт. шаг сетки](#) нужно ввести целое число от 1 до 99. Числа, задающие размер ячеек по горизонтали и вертикали, могут быть разными.

При выборе шага сетки нужно руководствоваться следующими соображениями: при маленьком шаге сложнее производить выравнивание и требуется выполнить больше действий, а при большом остаются большие промежутки между элементами, что приводит к увеличению размеров схем. Рекомендуется устанавливать шаг сетки, равный 20 (и по вертикали, и по горизонтали), так как он удачно сочетает разумный промежуток между элементами и достаточную точность изменения их размеров.

5.4. Выполнение действий с группой выделенных элементов

Некоторые действия по выравниванию, изменению положения и другие выполняются с группой выделенных элементов.

Выбор группы производится стандартным образом с помощью мыши.

Положение и размер группы элементов отображаются в панели состояния, как если бы это был один элемент схемы.

С группой выделенных элементов можно выполнять различные действия по выравниванию и изменению размеров с помощью специальных команд.

К таким действиям относятся:

- управление выравниванием,
- распределение элементов,
- установка размера.

Для группы элементов в палитре свойств можно установить одинаковые значения свойств (например, для декораций указать вертикальное и горизонтальное положение, размер и цвет шрифта и т. д.).

ПРИМЕЧАНИЕ 1. Для любой выделенной группы можно выполнять только изменение или установку общих свойств элементов, входящих в группу.

ПРИМЕЧАНИЕ 2. Групповые операции выполняются для всех выделенных элементов, кроме соединительных линий (т. к. соединительные линии перестраиваются автоматически, и пользователь не может управлять координатами их отдельных сегментов).

5.4.1. Управление выравниванием и распределением элементов схемы

Выравнивание элементов производится с помощью пунктов [Графическая схема – Выравнивание](#) или соответствующих этим командам команд меню [Выравнивание и размер](#).

Пункты меню доступны (выравнивание производится), если выбрано более одного элемента схемы.

Команды	Действия
Выровнять по левому краю	Производит выравнивание левых границ выбранных элементов. При этом их левые границы располагаются вдоль одной вертикальной прямой
Выровнять по центру	Производит вертикальное выравнивание элементов по центру активного элемента ("мастера")
Выровнять по правому краю	Производит выравнивание правых границ
Выровнять по верхнему краю	Производит выравнивание верхних границ
Выровнять по середине	Производит горизонтальное выравнивание элементов по середине активного элемента ("мастера")
Выровнять по нижнему краю	Производит выравнивание нижних границ

5.4.2. Распределение элементов схемы

Чтобы обеспечить равномерное распределение группы элементов графической схемы, удобно воспользоваться командами меню поля графической схемы или командами подменю [Графическая схема – Распределить](#).

Команда	Действия
Распределить по вертикали	Равномерно распределяет все элементы группы между самым верхним и самым нижним элементом
Распределить по горизонтали	Равномерно распределяет все элементы группы между самым левым и самым правым элементом

Команды доступны, если выбрано более двух элементов схемы.

При выполнении команды крайние элементы остаются на местах, а все остальные распределяются таким образом, чтобы промежутки между всеми выбранными элементами были одинаковыми. Средние элементы распределяются без привязки к разметочной сетке, даже если режим использования разметочной сетки установлен.

5.4.3. Установка размера

С помощью команд подменю [Графическая схема – Размер](#) можно установить одинаковые размеры всех элементов, входящих в группу. Следует обратить внимание на то, что за образец выбирается тот элемент, маркеры которого имеют вид «белый квадрат». Этот элемент называется мастером (на рисунке – верхний элемент). Другие элементы имеют маркеры вида «черный квадрат».

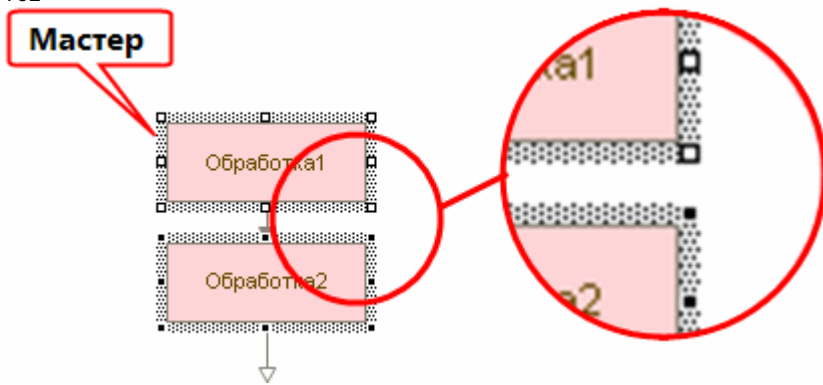


Рис. 278. Установка размера объектов

Чтобы сменить мастера, в выделенной группе элементов нужно щелкнуть мышью нужный элемент.

Команда	Действия
Одинаковая ширина	Устанавливает одинаковую ширину всех элементов группы
Одинаковая высота	Устанавливает одинаковую высоту всех элементов группы
Одинаковый размер	Устанавливает одинаковый размер всех элементов группы

Результат применения команды [Одинаковая ширина](#) для различных объектов-мастеров иллюстрирует следующий рисунок (в примере 1 мастером является верхний элемент, в примере 2 – нижний).



Рис. 279. Результат применения команды «Одинаковая ширина»

5.4.4. Порядок элементов

Когда элементы пересекаются (накладываются), то один элемент закрывает частично или полностью второй элемент. Установка порядка следования элементов производится с помощью команд меню поля графической схемы [Порядок](#) или команд подменю [Графическая схема – Порядок](#).

Команда	Действия
Поместить вперед	Перемещает указанный элемент на один слой вверх
Поместить назад	Перемещает указанный элемент на один слой вниз
На передний план	Перемещает указанный элемент на передний план
На задний план	Перемещает указанный элемент на задний план

Эти действия применяются как к одному элементу, так и к группе элементов.

5.4.5. Масштабирование

Для удобства просмотра всей графической схемы с помощью пункта [Графическая схема – Масштаб](#) (а также с помощью колеса прокрутки мыши с нажатой клавишей [Ctrl](#)) можно менять масштаб просмотра.

5.4.6. Прокрутка графической схемы

Для удобства прокрутки большой схемы можно нажать колесо мыши и выполнять прокрутку, выбирая курсором необходимое направление прокрутки. Скорость прокрутки зависит от расстояния между текущим положением курсора и тем местом, где было нажато колесо (отображается специальным образом).

5.4.7. Печать и предварительный просмотр

При редактировании графической схемы можно осуществлять печать и предварительный просмотр графической схемы с помощью пунктов меню **Файл – Печать** и **Файл – Предварительный просмотр**. С помощью пункта **Файл – Параметры страницы** можно менять настройки печати – большинство из них стандартны для Windows-программ.

Черно-белая печать – содержит признак печати и предварительного просмотра графической схемы в черно-белом виде.

Масштаб – дает возможность выбора режима масштабирования при печати. Варианты режима масштабирования:

- **Авто** – изменение масштаба не производится (режим по умолчанию).
- **По ширине страницы** – режим пропорционального масштабирования по ширине страницы. Если ширина графической схемы (правая координата ее крайнего справа элемента) больше ширины страницы, то производится пропорциональное масштабирование (пропорциональное изменение размеров по X и по Y) так, чтобы по ширине схема уместилась на страницу.
- **Пропорционально** – режим пропорционального масштабирования. Если высота (нижняя координата крайнего снизу элемента) или ширина графической схемы больше высоты (ширины) страницы, то производится пропорциональное масштабирование (пропорциональное изменение размеров по X и по Y) так, чтобы вся схема уместилась на одну страницу.

5.5. Свойства графической схемы

5.5.1. Категория свойств «Оформление»

В свойствах категории **Оформление** указывается оформление всей графической схемы.

Свойство **Цвет фона** определяет цвет фона графической схемы.

5.5.2. Категория свойств «Редактирование»

Выравнивать по сетке – использовать сетку и производить выравнивание элементов по сетке.

Гор. шаг сетки – горизонтальный шаг сетки.

Вер. шаг сетки – вертикальный шаг сетки.

Режим рисования сетки – режим показа линий сетки (**Рисовать линии**, **Рисовать точки**, **Рисовать шахматную сетку**, **Не рисовать сетку**).

5.6. Элементы графической схемы

В разделе описываются все типы элементов графической схемы в алфавитном порядке.

Поведение элементов настраивается в палитре свойств путем установки и выбора значений. Некоторые свойства элементов присущи всем или большинству типов элементов. Полное описание свойств для каждого элемента приводится в книге «1С:Предприятие 8.3. Описание встроенного языка».

Общие свойства элементов графической схемы.

Существует ряд свойств, которые присущи всем или большинству типов элементов. Обычно такие свойства имеют одинаковое назначение и редактируются одинаковым образом. Ниже будут описаны

такие свойства, а в разделах, посвященных отдельным типам элементов, будет изложен порядок редактирования уникальных свойств элементов.

Для удобства описания свойства будут сгруппированы по категориям палитры свойств.

Некоторые свойства, описываемые ниже, для некоторых элементов могут отсутствовать.

5.6.1. Категория свойств «Основные»

Имя – имя элемента. По имени элемента производится обращение к элементам графической схемы средствами встроеного языка.

Наименование – наименование элемента (многоязыковая строка, которая выводится внутри области, занимаемой элементом, либо поверх соединительной линии).

5.6.2. Категория свойств «Оформление»

Свойства данной категории позволяют осуществить настройку оформления (выбрать цвет текста, ввести текст подсказки и других параметров).

Цвет фона – цвет фона элемента. Цвет выбирается из палитры цветов или из стилей.

Цвет рамки – цвет рамки, отображаемой вокруг элемента.

Цвет текста – цвет текста элемента.

Шрифт – шрифт текста элемента. Задается вид шрифта, размер, начертание или выбирается из стиля.

Элемент (кроме соединительной линии) может содержать картинку (свойство **Картинка**). Положение картинки относительно основной части элемента задается в свойстве **Положение картинки**. Режим определения размера картинки в зависимости от размера элемента задается в свойстве **Размер картинки**.

Для всех элементов, кроме соединительных линий, есть свойства:

- **Рамка** – рамка, отображаемая вокруг элемента;
- **Прозрачный фон** – если свойство установлено, то отображение элемента производится с прозрачным фоном.

5.6.3. Категория свойств «Характеристики»

Горизонтальное положение – выбирается выравнивание текста по горизонтали.

Вертикальное положение – выбирается выравнивание текста по вертикали.

Подсказка – текст подсказки, которая будет выдаваться для пояснения назначения элемента.

5.6.4. Категория свойств «Расположение»

Свойства данной категории позволяют позиционировать элемент в схеме. Для всех элементов, кроме линий, есть свойства:

- **Лево** – позиция левой границы расположения элемента;
- **Верх** – позиция верхней границы расположения элемента;
- **Ширина** – ширина элемента;
- **Высота** – высота элемента.

5.6.5. Декоративная линия

Элемент графической схемы [Декоративная линия](#) предназначен для рисования линий, стрелок и соединения различных элементов. Элемент [Декоративная линия](#) автоматически перестраивается при изменении взаимного положения соединяемых им элементов. Возможности вмешиваться в алгоритм построения нет, за исключением ручного передвижения некраевых колен линии. В случае ручного изменения положения колен они запоминаются и по возможности учитываются при последующих автоматических перестроениях линии.

При построении линии система руководствуется правилом выбора кратчайшего пути, состоящего из вертикальных и горизонтальных отрезков линий, который не пересекает другие элементы схемы.

Колена декоративных линий могут быть строго вертикальными или строго горизонтальными. Место изгиба декоративной линии скругляется, что позволяет не терять направления при пересечении линий.

Автопостроение применяется только для линий, которые соединяют два элемента схемы.

Направление стрелки при вставке декоративной линии определяется соотношением сторон выделенного при вставке прямоугольника: если высота больше ширины, то выбирается вертикальное направление, иначе – горизонтальное.

Разделение декоративных линий на сегменты осуществляется без учета выравнивающей сетки.

5.6.6. Категория свойств «Оформление»

[Линия](#) – тип и толщина линии, которой рисуются отрезки элемента [Соединительная линия](#).

[Цвет линии](#) – цвет, которым рисуются отрезки элемента [Соединительная линия](#).

[Стрелка начала](#), [стрелка конца](#) – стиль стрелки у начала и конца линии (нет стрелки, заполненная или незаполненная).

5.6.7. Категория свойств «Характеристики»

[Позиция текста](#) – положение текста относительно линии (в первом сегменте – текст рисуется поверх первого сегмента линии или в середине линии (середине суммы длин всех сегментов линии)).

5.6.8. Декорация

Элемент графической схемы [Декорация](#) предназначен для отображения фигур с текстом и/или картинкой.

Особенность отрисовки декорации с установленным свойством [Прозрачный фон](#) в том, что в прозрачных участках декорации поверх нее рисуется сетка графической схемы и лежащие под ней элементы схемы.

5.6.9. Категория свойств «Оформление»

[Фигура](#) – тип отображаемой фигуры из числа predetermined:

- [Нет](#),
- [Блок](#),
- [Папка](#),
- [Файл](#),
- [Стрелка вправо](#),
- [Стрелка влево](#),

- Стрелка вверх,
- Стрелка вниз,
- Стрелка влево–вправо,
- Стрелка вверх–вниз,
- Документ,
- Скобки вертикальные,
- Скобки горизонтальные,
- Эллипс.

В зависимости от типа выбранной фигуры изменяется область текста и картинки. Например, у декорации с фигурой вида **Стрелка вправо** область текста и картинки ограничена прямоугольным основанием стрелки.

5.6.10. Точка действия

Элемент графической схемы вида **Действие** отображает точку бизнес-процесса, по которой выдаются и выполняются задачи.

5.6.11. Категория свойств «Адресация»

Пояснение – строка, характеризующая адресацию точки действия.

Групповая – если установлено в **Истина**, то задачи на данной точке бизнес-процесса будут выданы каждому члену группы (отдела). В противном случае выдается одна задача на всю группу, например, «Отдел продаж». Выполняет ее один человек (первый взявший ее).

Реквизиты адресации – этих свойств столько, сколько реквизитов адресации у задачи, указанной в свойстве **Задача** данного бизнес-процесса. В палитре свойств можно выбрать значение из предопределенных данных, тип которых задан в реквизите адресации задачи (например, из списка **Отделы** или **Исполнители**).

5.6.12. Точка разделения

Элемент графической схемы **Разделение** отображает точку бизнес-процесса, в которой поток исполнения разделяется на несколько параллельных веток, идущих одновременно. По умолчанию размещается на схеме с тремя исходящими соединительными линиями.

Для добавления выходящей линии в контекстном меню нужно выбрать пункт **Добавить линию**. Для удаления – выделить линию и выбрать пункт **Удалить**. При этом нельзя удалить единственную оставшуюся исходящую линию.

5.6.13. Точка условия

Элемент графической схемы **Условие** отображает точку бизнес-процесса, из которой есть два выхода, отражающих результат выполнения логического условия.

По умолчанию элемент **Условие** вставляется в графическую схему с двумя исходящими слева и справа соединительными линиями. Справа располагается ветка с комментарием **Да**, а слева – с комментарием **Нет**. Ветки условия можно поменять местами. Для этого нужно выбрать линию, исходящую из элемента **Условие**, и, захватив мышью прямоугольник у начала линии, перенести его на противоположную сторону элемента **Условие**.

5.6.14. Точка завершения

Элемент графической схемы [Завершение](#) отображает точку бизнес-процесса, в которой завершается бизнес-процесс.

5.6.15. Точка старта

Элемент графической схемы [Старт](#) отображает точку бизнес-процесса, с которой начинается выполнение бизнес-процесса.

5.6.16. Точка слияния

Элемент графической схемы [Слияние](#) отображает точку бизнес-процесса, в которую сходятся параллельные пути исполнения, начавшиеся в точке [Разделение](#).

5.6.17. Точка вложенного бизнес-процесса

Элемент графической схемы [Вложенный бизнес-процесс](#) отображает точку бизнес-процесса, в которой запускается на исполнение вложенный бизнес-процесс.

5.6.18. Точка обработки

Элемент графической схемы [Обработка](#) отображает точку бизнес-процесса, выполняемую в автоматическом режиме.

5.6.19. Точка выбора варианта

Элемент графической схемы [Выбор варианта](#) отображает точку бизнес-процесса, имеющую несколько выходов (вариантов), из которых выбирается только один. Для добавления нового варианта в контекстном меню нужно выбрать пункт [Добавить вариант](#). Для удаления варианта нужно, предварительно выбрав вариант, выбрать в контекстном меню пункт [Удалить вариант](#). Перемещать варианты нельзя.

По умолчанию элемент [Выбор варианта](#) вставляется в графическую схему с тремя вариантами. Из каждого варианта исходит соединительная линия. Ее нельзя удалить, а можно только передвинуть на противоположную сторону области варианта. При удалении варианта вместе с ним удаляется и исходящая соединительная линия. При добавлении варианта, соответственно, добавляется исходящая линия.

5.6.20. Категория свойств «Вариант»

Имя – имя варианта. По имени варианта производится обращение к варианту в обработчике `ОбработкаВыбораВарианта` средствами встроеного языка.

Наименование – наименование варианта (строка, которая выводится внутри области, занимаемой вариантом, внутри элемента [Выбор варианта](#)).

Цвет фона – цвет фона варианта. Этим цветом заливается область, занимаемая вариантом, внутри элемента [Выбор варианта](#).

Приложение 6. Редактор географической схемы

Географическая схема представляет собой совокупность топологических данных, которые описывают географические объекты, и произвольных данных, связанных с географическими объектами; причем эти данные могут быть сгруппированы в логические серии.

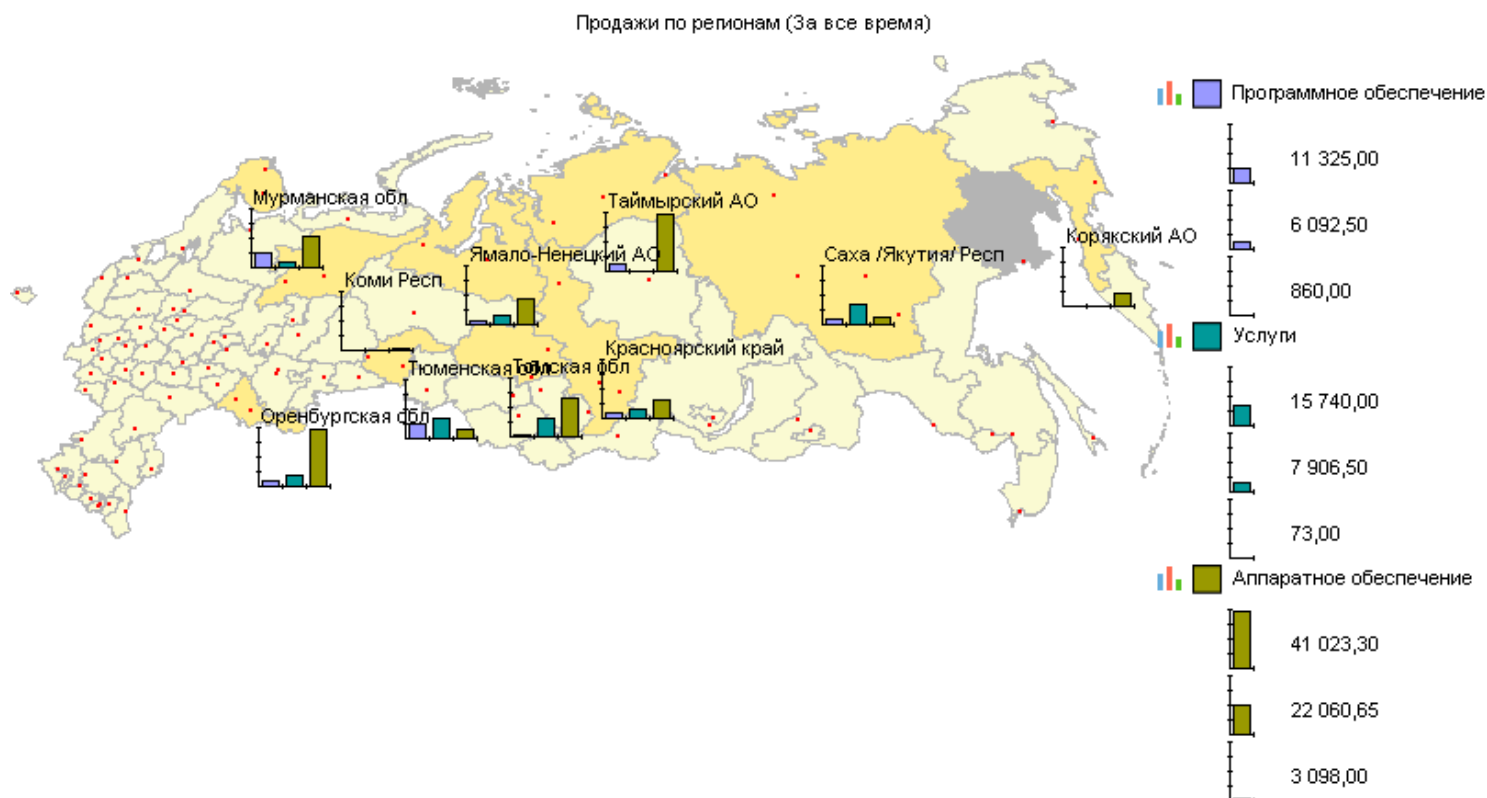


Рис. 280. Географическая схема

Географическая схема может использоваться самостоятельно (в документе [Географическая схема](#)), может располагаться в табличном документе. В этом случае для нее в табличном документе может быть выделена область данных, отображаемых в географической схеме. Также географическая схема может располагаться в форме. В последнем случае настройка и подготовка исходных данных выполняются при конфигурировании.

Географическая схема содержит в себе географические объекты, которые обычно сгруппированы по типу или по некоторым признакам с помощью слоев.

Слой – совокупность географических объектов, сгруппированных по некоторому признаку. Например, слой «Реки» содержит объекты, описывающие геометрию рек в регионе, а слой «Районы» – описание районов города.

Объект слоя – элементарный объект, описывающий топологические данные конкретного географического объекта. Например, объект может описывать контуры области.

Объекты делятся на следующие классы:

- **Точечный** – представляет собой объект, который можно представить точкой. Например, на схеме большого масштаба города можно представлять точечными объектами.
- **Многоточечный** – объект, который можно представить несколькими точками. Например, города в области можно сгруппировать по количеству населения, и каждую такую группу представить таким объектом.
- **Полилинейный** – объект, который можно отобразить линией. Например, такими объектами на схеме можно представить реку или шоссе.
- **Полигональный** – объект, описывающий некоторую площадь. Например, Владимирскую область на географической схеме можно представить полигональным объектом.

Серия – произвольные данные, сгруппированные по некоторому признаку. Примером такой серии могут быть продажи по регионам за год.

Значение серии – значение серии данных, ассоциированное с конкретным объектом. Например, для серии данных «Продажи по городам за год» значением серии данных будет являться значение продаж по отдельному городу за определенный год.

Шейп-файл (Shapefile) – файл с топологическими данными, созданный с помощью программ семейства ESRI ArcGIS. Может содержать связанные с топологическими объектами данные. Каждый файл содержит один слой. Географическая схема позволяет считывать такие файлы в качестве слоя.

ВНИМАНИЕ! «1С:Предприятие» не предоставляет средств создания шейп-файлов.

Географическая схема представляет собой прямоугольную область экрана, которая содержит области построения, заголовка и легенды.

Области можно перемещать, изменять их размер, настраивать их свойства или просто отключить их показ (кроме области построения). При изменении размеров географической схемы размеры областей меняются пропорционально.

Область построения – содержит координатное пространство и отображает в себе данные.

Область заголовка – несет информативную нагрузку и отображает произвольный заголовок.

Область легенды – вспомогательная область, в которой отображаются условные обозначения, помогающие пониманию отображенных в области построения данных.

6.1. Создание географической схемы

Для создания географической схемы нужно выбрать пункт **Файл – Новый** и в открывшемся диалоге выбрать тип документа **Географическая схема**.

Для размещения географической схемы в табличном документе выбрать пункт **Таблица – Вставить рисунок – Географическая схема**.

6.2. Сохранение географической схемы

Для создания географической схемы нужно выбрать пункт **Файл – Сохранить**. В открывшемся диалоге выбрать каталог и указать имя файла.

Документы **Географическая схема** сохраняются в файле с расширением GEO.

6.3. Настройка географической схемы

Задача настройки состоит в определении топологических объектов, к которым будут привязаны анализируемые данные, формата, специфического для разных областей, их размера и положения.

Если географическая схема расположена в табличном документе, можно загрузить в нее документ **Географическая схема**, существующий на диске, воспользовавшись командой **Прочитать** в палитре свойств.

Для очистки можно воспользоваться ссылкой **Очистить**.

Для управления слоями географической схемы используется ссылка **Настроить**. На экран выводится окно настройки слоев.

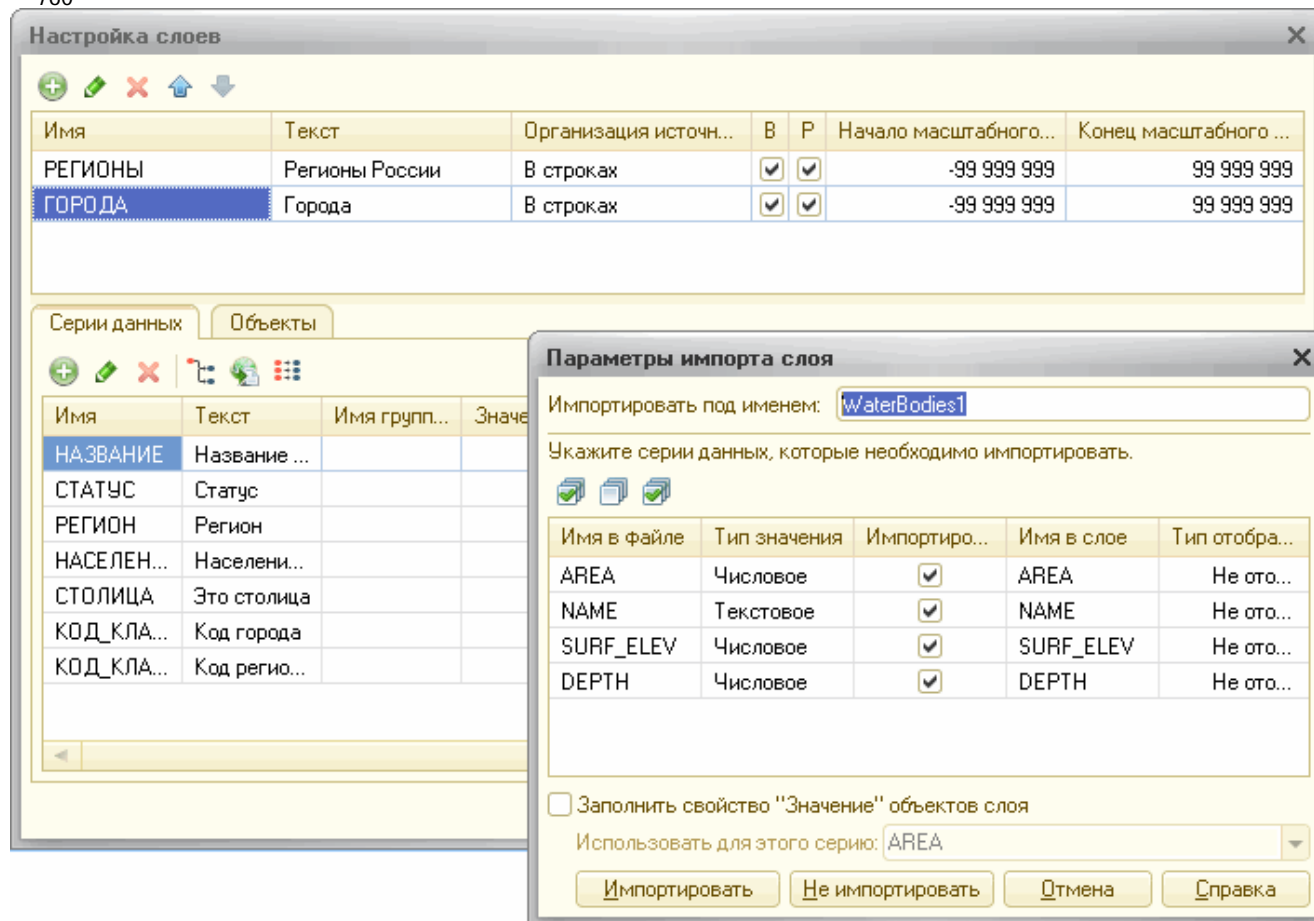


Рис. 281. Географическая схема. Настройка слоев

С помощью кнопок панели инструментов осуществляется добавление, редактирование и удаление слоя.

При добавлении слоя сначала открывается диалог выбора слоя, а затем окно параметров импорта слоя (на рисунке).

В окне следует указать, под каким именем этот слой будет импортирован в схему, какие серии данных необходимо импортировать, под каким именем и как они будут отображаться.

Каждый топологический объект в географической схеме имеет специальное свойство **Значение**, которое предназначено для хранения произвольных данных и в общем случае позволяет легко идентифицировать объекты.

В окне **Параметры импорта слоя** для заполнения значения для каждого объекта используется значение какой-либо серии данных. Для импорта значений нужно установить флажок **Заполнить свойство ""Значение"" объектов слоя** и указать, из какой серии необходимо импортировать данные.

После того как слои выбраны, можно переходить к их настройке.

Для **слоя** настраиваются:

- **Текст** – произвольный текст, описывающий слой;
- **Организация источника данных** – устанавливает организацию источника данных, если такой источник используется. Подробно об этом – в разделе, описывающем особенности работы с табличным документом;
- **Видимость** – определяет видимость слоя на схеме;
- **Разрешить выбор** – определяет возможность интерактивного выбора объектов слоя на схеме;
- **Начало масштабного диапазона** – определяет начало диапазона масштабирования, при попадании в который слой будет виден на схеме;
- **Конец масштабного диапазона** – определяет конец диапазона масштабирования, при попадании в который слой будет виден на схеме.

Для **серий** настраиваются:

- **Текст** – произвольный текст, описывающий серию;
- **Формат** – формат отображения данных;
- **Тип отображения** – тип отображения серии. Значения серии могут:
 - не отображаться;
 - отображаться в виде текста;
 - отображаться в виде гистограммы;
 - отображаться в виде кругов, размер которых зависит от значения;
 - отображаться в виде кругов, цвет которых зависит от значения;
 - отображаться в виде кругов, оттенок цвета которых зависит от значения;
 - отображаться круговой диаграммой;
 - отображаться картинкой;
- **Базовый цвет** – цвет, в зависимости от которого будет отображаться в режиме **Гистограмма** серия кругов разного диаметра, кругов разного оттенка, круговой диаграммы;
- **Цвет текста** – цвет, которым будут отображаться текстовые данные;
- **Шрифт текста** – шрифт, которым будут отображаться текстовые данные;
- **Ориентация текста** – угол поворота текста.

Для **точечных и многоточечных объектов** настраиваются:

- **Тип отрисовки объекта** – в виде маркера, в виде указанного символа, в виде картинки;
- **Цвет** – цвет, которым будет отображаться маркер или символ;
- **Маркер** – отображаемый маркер;
- **Шрифт** – шрифт, которым будет выводиться символ;
- **Символ** – выводимый символ;
- **Картинка** – отображаемая картинка;
- **Опорная точка** – точка в картинке, которая будет принята за начало отсчета при выводе.

Для **полилинейных объектов** настраиваются:

- **Цвет** – цвет, которым будет отображаться линия;
- **Тип линии** – тип линии, которым будет отображаться линия.

Для **полигональных объектов** настраиваются:

- **Цвет** – цвет заливки;
- **Цвет границы** – цвет, которым будет отображаться граница полигона.

6.3.1. Настройка элементов легенды

Для настройки элементов легенды нужно выбрать область легенды и в палитре свойств нажать ссылку **Настроить**. На экран выводится диалог.

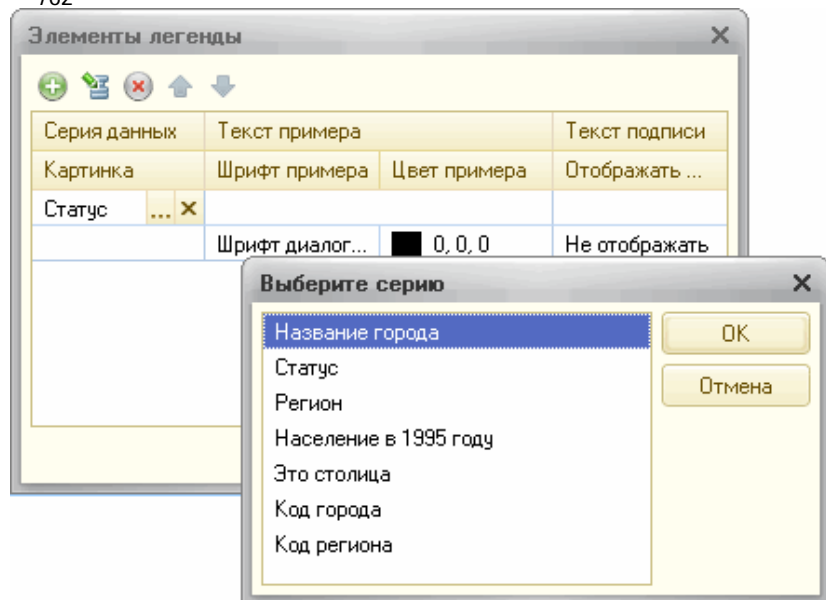


Рис. 282. Настройка элемента легенды географической схемы

Управление элементами производится с помощью кнопок командной панели. При добавлении элемента сначала выводится список слоев, а после выбора слоя выводится список серий.

Следует выбрать нужную серию и указать ее параметры настройки.

6.3.2. Свойства объекта

После того, как в документ импортированы слои, можно настроить свойства отдельных объектов. Для этого нужно дважды щелкнуть указателем мыши нужный объект. На экран выводится окно свойств объекта.

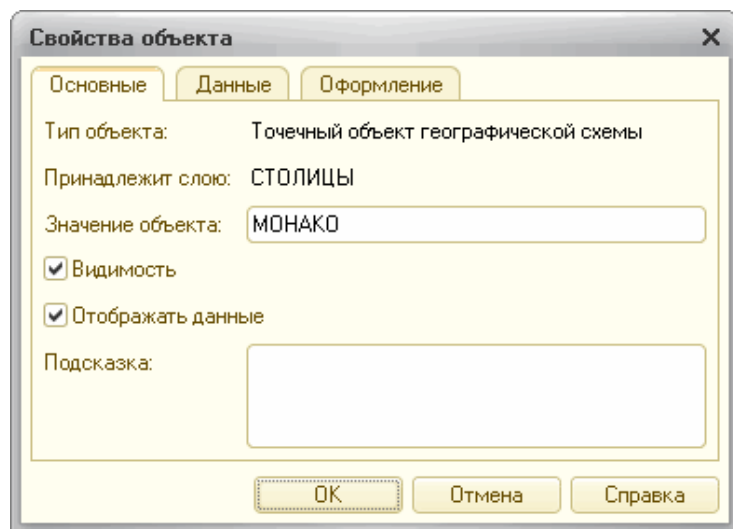


Рис. 283. Основные свойства объекта географической схемы

На закладке [Основные](#) указываются основные свойства объекта.

На закладке [Данные](#) приводится список серий, типы их отрисовки и идентифицирующие их значения.

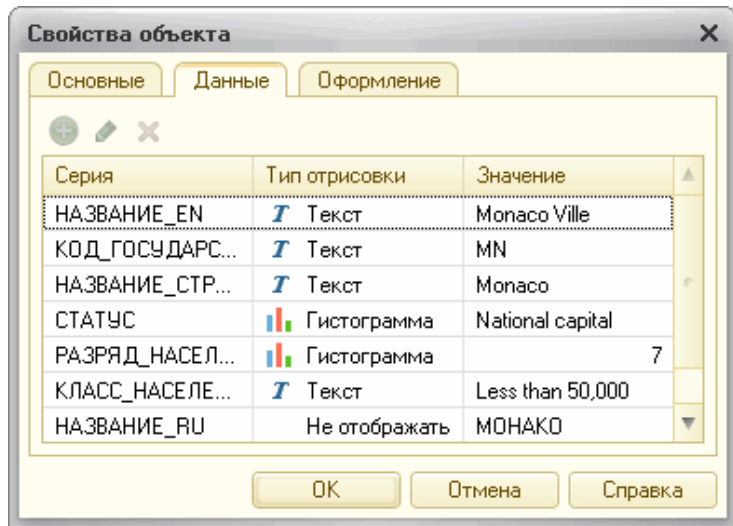


Рис. 284. Данные объекта географической схемы

На закладке **Оформление** выбирается оформление элемента.

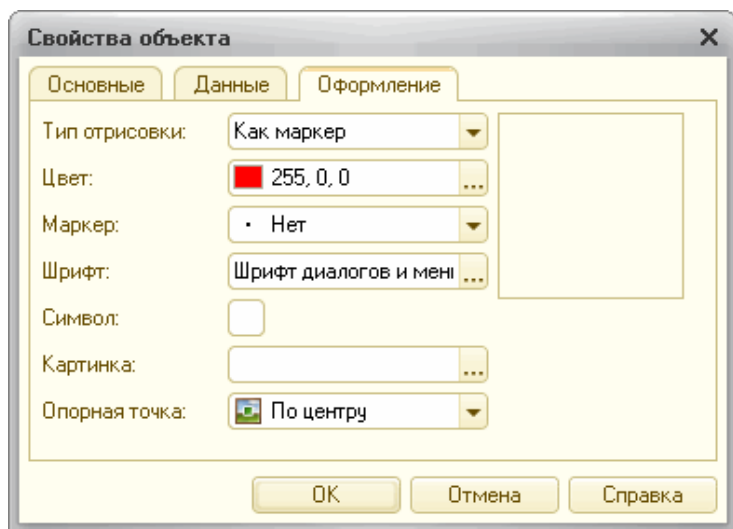


Рис. 285. Оформление объекта географической схемы

6.4. Свойства географической схемы

6.4.1. Категория свойств «Оформление»

Прозрачный фон – устанавливает прозрачность фона.

Цвет фона – устанавливает цвет фона географической схемы.

Поддержка масштаба – свойство позволяет задать режим масштабирования, в зависимости от которого будут показываться данные.

Доступные варианты:

- **Все данные** – данные масштабируются так, чтобы они уместились целиком в видимую область;
- **Задается масштабом** – данные масштабируются в зависимости от масштаба;
- **Заданная область** – на экран выводится только заданная координатами область;
- **Масштаб** – масштаб, в котором будут выводиться данные в режиме **Задается масштабом**.
- **Проекция** – географическая проекция, в которой будет отображаться географическая схема;
- **Сдвиг по широте** – величина в градусах, на которую будет сдвинута точка отсчета координат по широте;

- **Сдвиг по долготе** – величина в градусах, на которую будет сдвинута точка отсчета координат по долготе;
- **Отображать координаты** – если установлено, то координаты точки, находящейся под курсором мыши, выводятся в панель состояния. Данное свойство доступно не для всех видов проекций;
- **Слева** – левая граница заданной области. Доступно, если выбран режим масштабирования **Заданная область**. Задается в десятичных градусах;
- **Снизу** – нижняя граница заданной области. Доступно, если выбран режим масштабирования **Заданная область**. Задается в десятичных градусах;
- **Справа** – правая граница заданной области. Доступно, если выбран режим масштабирования **Заданная область**. Задается в десятичных градусах;
- **Сверху** – верхняя граница заданной области. Доступно, если выбран режим масштабирования **Заданная область**. Задается в десятичных градусах;
- **Заголовок** – управляет видимостью области заголовка. Доступно, если географическая схема расположена в табличном документе или в форме;
- **Легенда** – управляет видимостью области легенды. Доступно, если географическая схема расположена в табличном документе или на форме.

6.4.2. Категория свойств «Серии»

Категория свойств доступна, если географическая схема расположена в табличном документе. В расположенных в категории свойствах выполняется настройка источника данных для слоя.

Активный слой – указывается слой, для которого будет настраиваться источник данных.

Тип содержимого – указывается тип организации источника данных.

Возможны два варианта расположения данных: в строках и на пересечении.

Объект 1	Серия данных 1	Значение 1
Объект 2	Серия данных 2	Значение 2
Объект 3	Серия данных 3	Значение 3
Объект 4	Серия данных 4	Значение 4
Объект 5	Серия данных 5	Значение 5

	Серия 1	Серия 2	Серия 3	Серия 4
Объект 1	Значение 1	Значение 1	Значение 1	Значение 1
Объект 2	Значение 2	Значение 2	Значение 2	Значение 2
Объект 3	Значение 3	Значение 3	Значение 3	Значение 3
Объект 4	Значение 4	Значение 4	Значение 4	Значение 4

Диапазон данных – диапазон данных в табличном документе, который будет использоваться как источник данных (аналогично диаграмме).

6.5. Свойства заголовка географической схемы

Свойства доступны, если географическая схема расположена в табличном документе.

6.5.1. Категория свойств «Оформление»

Текст – текст, отображаемый в заголовке.

Шрифт – шрифт текста заголовка.

Цвет заголовка – цвет текста заголовка.

Выравнивание – определяет выравнивание текста заголовка.

Рамка – тип рамки области заголовка.

Цвет рамки – цвет рамки области заголовка.

Прозрачный фон – определяет прозрачность фона области заголовка.

Цвет фона – цвет фона области заголовка.

6.6. Свойства легенды географической схемы

Свойства доступны, если географическая схема расположена в табличном документе.

6.6.1. Категория свойств «Оформление»

Шрифт – шрифт текста легенды.

Цвет – цвет текста легенды.

Рамка – тип рамки области легенды.

Цвет рамки – цвет рамки области легенды.

Прозрачный фон – определяет прозрачность фона области легенды.

Цвет фона – цвет фона области легенды.

Масштабная линейка – определяет видимость масштабной линейки в области легенды.

Элементы легенды – ссылка [Настроить](#) открывает диалог **Элементы легенды**, в котором выполняется настройка элементов легенды.

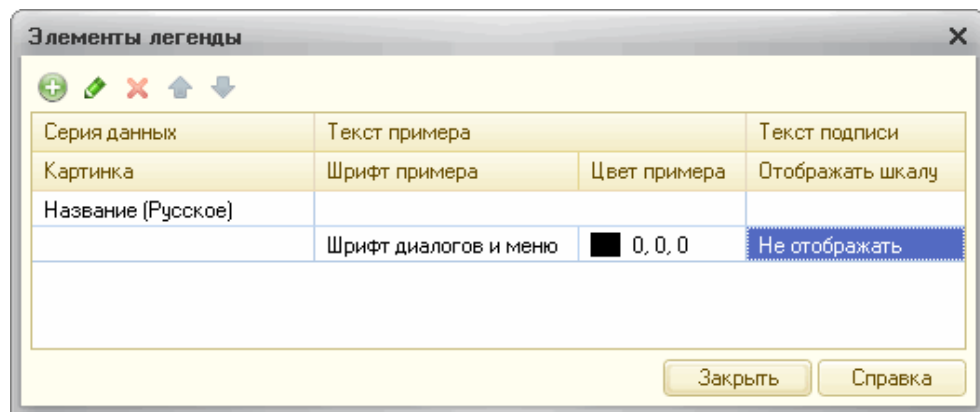


Рис. 286. Диалог настройки элементов легенды географической схемы

Серия данных – серия данных, связанная с элементом легенды. Для добавления серии следует нажать кнопку **Добавить** панели инструментов, выбрать слой, а затем выбрать серию.

Картинка – картинка, дополнительно расшифровывающая элемент легенды.

Текст примера – текст, как он отображается на схеме.

Шрифт примера – шрифт текста примера.

Цвет примера – цвет текста примера.

Текст подписи – текст пояснительной подписи.

Если элемент легенды связан с серией данных и текст примера не задан, выводится текст серии данных.

6.7. Поиск и замена текста

Для всех текстовых свойств ([Текст](#), [Формат](#), [Текст подписи](#), [Текст примера](#), [Заголовок](#)) и текстовых значений объектов возможен поиск и замена текста (см. раздел «[Поиск и замена](#)» [здесь](#)).

Для просмотра и редактирования значений объекта следует использовать пункт [Свойства объекта](#) контекстного меню объекта географической схемы.

1С:Предприятие 8.3.6

Руководство разработчика
83.002.11 от 30.06.2015

- Глава 1. Концепция системы
- Глава 2. Работа с конфигурацией
- Глава 3. Интерфейс приложения
- Глава 4. Встроенный язык
- Глава 5. Объекты конфигурации
- Глава 6. Командный интерфейс
- Глава 7. Формы
- Глава 8. Работа с запросами
- Глава 9. Работа с данными
- Глава 10. Система компоновки данных
- Глава 11. Бухгалтерский учет
- Глава 12. Периодические расчеты
- Глава 14. Анализ данных и прогнозирование
- Глава 15. Механизмы обмена данными
- Глава 16. Механизм XDTO
- Глава 17. Работа с JSON
- Глава 19. Механизм заданий
- Глава 20. Механизм полнотекстового поиска в данных
- Глава 22. Журнал регистрации
- Глава 23. Механизм криптографии
- Глава 24. Внешние источники данных
- Глава 25. Механизм разделения данных
- Глава 26. Разработка решений для мобильной платформы
- Глава 27. Инструменты разработки
- Глава 28. Отладка и тестирование прикладных решений
- Глава 29. Механизм сравнения и объединения конфигураций
- Глава 30. Групповая разработка конфигурации
- Глава 31. Поставка и поддержка конфигурации
- Глава 32. Расширение конфигурации
- Глава 33. Сервисные возможности
- Глава 34. Внешние компоненты
- Глава 35. Особенности разработки кроссплатформенных прикладных решений

- Приложение 01. Конвертация информационных баз системы «1С:Предприятие 7.7»
- Приложение 02. Форматы навигационных ссылок
- Приложение 03. Правила формирования текстов стандартных команд и автоматических заголовков форм
- Приложение 04. Перечень автоматически сохраняемых настроек
- Приложение 05. Поисквые выражения полнотекстового поиска
- Приложение 06. Описание прав доступа
- Приложение 07. Особенности поведения системы в различных режимах
- Приложение 08. Особенности работы с различными СУБД
- Приложение 09. Работа с XBase
- Приложение 10. Особенности использования отдельных механизмов
- Приложение 11. Правила автоматического формирования имен элементов формы
- Приложение 12. Описание сущностей, через стандартный интерфейс OData
- Приложение 13. Соответствие синхронных методов асинхронным аналогам

Руководство разработчика

ПРАВО ТИРАЖИРОВАНИЯ
ПРОГРАММНЫХ СРЕДСТВ И ДОКУМЕНТАЦИИ
ПРИНАДЛЕЖИТ ФИРМЕ «1С»

Приобретая систему «1С:Предприятие», вы тем самым даете согласие не допускать копирования программ и документации без письменного разрешения фирмы «1С».

© ООО «1С», 1996 – 2015

Фирма «1С», Москва, 123056, а/я 64

Отдел продаж: Селезневская ул., 21,

телефон: +7 (495) 737-92-57,

факс: +7 (495) 681-44-07,

e-mail: 1c@1c.ru

URL: <http://www.1c.ru>, <http://www.v8.1c.ru>, <http://users.v8.1c.ru>

Группа разработки программ – А. Алексеев, Р. Алейников, А. Безбородов, Д. Бескоровайнов, А. Бушнев, П. Василец, А. Виноградов, Я. Вирковский, А. Волков, И. Гольштейн, Е. Горностаев, И. Гусаров, Г. Дамье, А. Даровских, О. Дерут, Н. Евграфов, Б. Евтифеев, Д. Зарецкий, Д. Ивашов, И. Коваленко, С. Копиенко, Н. Корсаков, С. Кравченко, В. Кудрявцев, П. Кукушкин, А. Кунченко, А. Лакутин, М. Лейбович, Г. Леонтьев, А. Лехан, А. Макеев, А. Малышенок, А. Медведев, Е. Митрошкин, С. Мурзин, С. Нуралиев, Л. Онучин, М. Отставнов, Д. Павленко, А. Плякин, П. Романов, А. Рукин, Д. Русанов, М. Саблин, Е. Силин, Д. Службин, А. Смирнов, В. Соколов, П. Солодкий, А. Соляник, В. Сосновский, Д. Сысоенков, Д. Тишков, А. Топорков, В. Тунегов, А. Трубкин, А. Тюшкин, В. Филиппов, А. Хасанов, А. Цилябин, В. Ческис, В. Черемисинов, П. Чиков, А. Чичерин, А. Шевченко, А. Щербинин, В. Шульга.

Документация – В. Байдаков, В. Дранищев, Е. Королькова, А. Краюшкин, И. Кузнецов, М. Лавров, А. Монищев, А. Плякин, М. Радченко.

Группа консультационной поддержки – О. Акулова, С. Алексеева, О. Багрова, О. Баклушина, Э. Гарифуллина, В. Давыдова, О. Дмитренко, Л. Ермакова, М. Ершова, Ю. Жестков, О. Завальская, Н. Заявлиная, М. Звонилов, М. Иванова, Г. Коробка, Ю. Лаврова, С. Лепешкина, С. Мазурин, С. Марков, Ю. Мисан, А. Павликов, И. Панин, О. Пехтерева, С. Постнова, А. Прокуровский, Е. Романова, Г. Степаненко, Н. Степанов, Т. Токарева, Е. Широкова.

Группа тестирования – С. Баталин, А. Беляков, Б. Зиатдинов, А. Лапин, Е. Медведев, С. Потапкин.

Корректурa – Е. Семененко.

Верстка – О. Шестакова.

Наименование книги:	1С:Предприятие 8.3. Руководство разработчика.
Номер издания:	83.002.11
Дата выхода:	30 июня 2015 года

ЛИНИЯ КОНСУЛЬТАЦИЙ

Для пользователей системы «1С:Предприятие 8» услуги линии консультаций предоставляются по линии

информационно-технологического сопровождения (ИТС).

Получение консультаций возможно только после регистрации программного продукта (для чего необходимо заполнить регистрационную анкету и выслать ее в фирму «1С») и оформления подписки на ИТС.

В стоимость комплекта поставки программного продукта включено обслуживание по линии ИТС в течение периода, указанного в вашей регистрационной анкете. Купон на бесплатное обслуживание по линии ИТС, а также конверт с маркой и адресом входят в состав приобретенного вами комплекта программ.

По истечении периода бесплатного обслуживания получение консультаций возможно только по договору на ИТС. Для заключения договора обращайтесь к партнерам фирмы «1С».

Телефон и адрес электронной почты для консультаций указаны на диске ИТС.

Линия консультаций работает с 9:30 до 17:30 по московскому времени, кроме суббот, воскресений и праздничных дней (по распорядку московских предприятий).

Обращаясь к линии консультаций, следует находиться рядом со своим компьютером, иметь под рукой настоящее руководство и свою половину регистрационной анкеты. Желательно заранее уточнить типы используемого компьютера и принтера.

Набрав телефон линии консультаций, вы должны услышать ответ дежурного сотрудника. После этого сообщите наименование вашей организации, номер вашего экземпляра системы (он обозначен на CD-ROM комплекта поставки и на вашей половине регистрационной анкеты) и, возможно, другую регистрационную информацию по запросу сотрудника линии консультаций. Названная вами информация будет проверена по данным, указанным в отосланной в фирму «1С» половине регистрационной анкеты.

Отвечая на возникшие у вас вопросы, дежурный сотрудник может воспроизвести возникшую ситуацию на своем компьютере. Он даст вам рекомендации сразу же или после обсуждения с разработчиками. Вам не нужно просить к телефону конкретных специалистов: мы отвечаем за работу всего персонала. Работа линии консультаций регистрируется, поэтому при повторных обращениях по сложным проблемам вы можете сослаться на дату и время предыдущего разговора.

МЫ ВСЕГДА РАДЫ ВАМ ПОМОЧЬ!

Введение

Настоящая книга является руководством по разработке прикладных решений для системы «1С:Предприятие».

Данный документ предназначен для специалистов, занимающихся разработкой, модификацией и внедрением прикладных решений на платформе «1С:Предприятие».

Структура Руководства

Руководство содержит описание общей концепции системы «1С:Предприятие», а также рассказывает о приемах работы с объектами системы, о создании форм, командного интерфейса, использовании различных сервисных механизмов системы.

Глава 1 описывает концепцию системы «1С:Предприятие»: заложенные в нее принципы, общая схема работы и другое.

Глава 2 посвящена работе с конфигурацией в целом: редактированию свойств конфигурации, копированию объектов конфигурации, сохранению конфигурации, объединению конфигураций, вспомогательным режимам работы.

Глава 3 описывает интерфейс приложения и его устройство.

Глава 4 содержит общие сведения о встроенном языке, описание используемых типов данных, операторов и синтаксических конструкций, а также основные приемы работы со встроенным языком.

Глава 5 посвящена основным объектам конфигурации и их особенностям.

Глава 6 посвящена описанию командного интерфейса и порядку его разработки.

Глава 7 содержит описание формы, ее составных частей, а также описание работы с формой из встроенного языка.

Глава 8 содержит описание языка запросов и основные приемы работы с запросами во встроенном языке.

Глава 9 рассказывает об общих принципах и особенностях работы с данными системы «1С:Предприятие».

Глава 10 посвящена описанию системы компоновки данных.

Глава 11 посвящена организации бухгалтерского учета в системе «1С:Предприятие».

Глава 12 посвящена работе с механизмами периодических расчетов, используемых в системе «1С:Предприятие».

Глава 13 описывает бизнес-процессы и задачи.

Глава 14 содержит описание механизма анализа данных и прогнозирования.

Глава 15 посвящена описанию механизма обмена данными.

Глава 16 описывает механизм XDTO.

Глава 17 описывает работу с JSON.

Глава 18 дает описание механизма Web-сервисов, HTTP-сервисов и стандартного интерфейса OData.

Глава 19 описывает механизм заданий.

Глава 20 описывает средства полнотекстового поиска в базе данных.

Глава 21 описывает работу с временным хранилищем данных и применение временного хранилища для операций с файлами.

Глава 22 описывает работу с журналом регистрации.

Глава 23 рассказывает о механизме криптографии.

Глава 24 описывает работу с внешними источниками данных.

Глава 25 посвящена описанию механизма разделения данных.

Глава 26 описывает разработку приложений для мобильных устройств.

Глава 27 рассказывает об использовании специализированного редактора форм, текстового редактора, редактора табличных документов, редактора карты маршрута и редактора картинок.

Глава 28 посвящена отладке и тестированию конфигураций.

Глава 29 описывает механизм сравнения и объединения конфигураций.

Глава 30 посвящена организации и использованию групповой разработки конфигураций.

Глава 31 посвящена описанию механизмов создания файлов поставки и обновления, а также комплекта поставки. Данная глава будет интересна разработчикам тиражных конфигураций. Также в данной главе описывается использование файлов поставки и обновлений для типовых конфигураций, находящихся на поддержке разработчиков пользовательских конфигураций.

Глава 32 описывает расширения конфигурации – механизм, предназначенный для доработки расширяемой конфигурации без изменения этой конфигурации.

Глава 33 рассказывает о сервисных режимах: настройке параметров конфигуратора, работе с синтаксис-помощником, настройке шаблонов, использовании встроенного калькулятора и календаря, сравнении файлов, особенностях работы с окнами, локализации конфигураций в системе «1С:Предприятие».

Глава 34 описывает механизм подключения внешних компонентов.

Глава 35 содержит рекомендации по разработке кроссплатформенных прикладных решений.

Что вы должны знать

Характер изложения материала данного Руководства предполагает, что вы знакомы с операционной системой компьютера, на котором работает система «1С:Предприятие» (Microsoft Windows 2000, Microsoft Windows XP, Microsoft Windows Vista, Microsoft Windows 7, далее – Microsoft Windows), и владеете базовыми навыками работы в ней.

Вам должны быть знакомы следующие понятия и навыки:

- использование меню **Пуск (Start)** для вызова программ;
- приемы работы с окнами;
- приемы работы с мышью;
- стандартные приемы работы с текстом и табличными документами (электронными таблицами) – ввод текста, ввод значений в ячейки табличного документа, редактирование, форматирование, печать

и др.;

- работа с меню;
- использование управляющих элементов диалогов;
- стандартные диалоги;
- понятие буфера обмена Microsoft Windows и приемы работы с ним (далее буфер обмена);
- настройка операционной системы Microsoft Windows с помощью панели управления.

Если вы недостаточно хорошо владеете перечисленными выше понятиями и навыками, рекомендуем обратиться к документации по операционной системе.

Книги документации

В состав документации входят книги по технологической платформе «1С:Предприятие»:

- «1С:Предприятие 8.3. Руководство пользователя». Книга содержит описание общих приемов работы с программными продуктами, созданными на базе платформы «1С:Предприятие», и может не поставляться отдельно. Данная книга не поставляется в составе продукта «1С:Предприятие 8.3. Версия для обучения программированию».
- «1С:Предприятие 8.3. Руководство разработчика». Книга необходима для изменения и настройки конфигурации под особенности учета конкретной организации, а также для разработки новых конфигураций.
- «1С:Предприятие 8.3. Руководство администратора». Книга описывает администрирование системы «1С:Предприятие», включая информацию об особенностях построения клиент-серверных систем. Данная книга не поставляется в составе продукта «1С:Предприятие 8.3. Версия для обучения программированию».
- «1С:Предприятие 8.3. Клиент-серверный вариант. Руководство администратора». Книга описывает особенности работы системы «1С:Предприятие» с информационными базами в варианте клиент-сервер в части установки и эксплуатации. Данная книга не поставляется в составе продукта «1С:Предприятие 8.3. Версия для обучения программированию».
- Синтаксис встроенного языка и языка запросов представлен в книге «1С:Предприятие 8.3. Руководство разработчика». Описание объектной модели полностью включено в поставку в электронном виде (в разделах справки конфигуратора и синтакс-помощнике). Описание объектной модели также содержится в книге «1С:Предприятие 8.3. Описание встроенного языка», которая распространяется отдельно.

ВНИМАНИЕ! Комплект поставки конкретного продукта может включать лишь некоторые из перечисленных книг документации.

Методические материалы и дополнительные возможности

Фирма «1С» осуществляет методическую поддержку освоения и внедрения системы программ «1С:Предприятие». Методическая поддержка включает в себя разнообразные формы предоставления информации, необходимой для грамотной и эффективной разработки и использования прикладных решений.

Сопроводительные файлы комплекта поставки «1С:Предприятия»

В процессе установки платформы «1С:Предприятие» выполняется копирование на жесткий диск ряда сопроводительных файлов, содержащих описание изменений, реализованных в данной версии платформы, и инструкции по переходу с предыдущих версий.

Все сопроводительные файлы располагаются в каталоге установочных файлов конкретной версии системы «1С:Предприятие», в каталоге `\docs\ru`. Если при установке системы использовался каталог, предложенный по умолчанию, то эти файлы будут располагаться в каталоге `C:\Program Files\1cv82\НомерВерсии\docs\ru`. Здесь `НомерВерсии` означает номер установленной версии. Так, для версии 8.2.9.100 каталог будет иметь следующий вид: `C:\Program Files\1cv82\8.2.9.100\docs\ru`.

- [V8Update.htm](#) – в этом файле содержатся отличия текущей версии платформы от предыдущих версий и особенности перехода на новую версию.

ИТС – информационно-технологическое сопровождение

Фирма «1С» осуществляет платную методическую поддержку пользователей в рамках информационно-технологического сопровождения (ИТС) программ системы «1С:Предприятие».

Ежемесячные выпуски ИТС содержат большое количество постоянно обновляемой информации, позволяющей более эффективно использовать продукты системы «1С:Предприятие». Отметим наиболее важные составляющие ИТС:

- обновления технологической платформы «1С:Предприятия» и прикладных решений;
- методические материалы по технологической платформе «1С:Предприятие»;
- методические материалы по типовым прикладным решениям фирмы «1С»;
- конфигурация «Конвертация данных» для настройки правил обмена между информационными базами «1С:Предприятия», имеющими различную конфигурацию;
- система стандартов и методик разработки конфигураций для платформы «1С:Предприятие» (предназначена для ознакомления партнеров и пользователей фирмы «1С» с техническими и проектными решениями, используемыми при разработке типовых конфигураций на платформе «1С:Предприятие»);
- советы линии консультаций, помогающие пользователям «1С:Предприятия» получить ответы на наиболее часто встречающиеся вопросы и избежать типичных ошибок;
- информация по обучению работе с платформой «1С:Предприятие» и прикладными решениями фирмы «1С»;
- демонстрационные ролики программных продуктов, позволяющие получить первое представление о возможностях прикладных решений;
- справочники по заполнению деклараций по налогам (налог на прибыль, НДС, налог на имущество, ЕСН, взносы в ПФР);
- обширная подборка бухгалтерской периодики, включая текущие выпуски журналов и архивы;
- база аналитических обзоров законодательства и арбитражной практики;
- рекомендации по составлению квартальной и годовой отчетности в «1С:Бухгалтерии»;
- справочник типовых хозяйственных операций хозрасчетного предприятия;
- справочник по оформлению расчетов с персоналом по оплате труда и правовым аспектам трудовых отношений;
- справочная правовая система «Гарант» – полный набор нормативных документов законодательства Российской Федерации, в том числе по бухгалтерскому учету, налогам и предпринимательству.

Начиная с 2005 года, стандартный сервис по поддержке пользователей – подписчиков ИТС включает

доступ к сайту поддержки пользователей системы «1С:Предприятие».

Более подробно о проекте ИТС можно прочитать на сайте фирмы «1С»:
<http://www.1c.ru/rus/support/its/its.htm>.

Оставить заявку на демонстрацию дисков и приобретение подписки на ИТС можно на сайте фирмы «1С»:
<http://www.1c.ru/rus/support/its/zajavka.jsp>.

Оформить подписку на ИТС можно у партнеров фирмы «1С». Список партнеров, имеющих опыт массового обслуживания пользователей в рамках проектов ИТС, опубликован на сайте фирмы «1С»:
<http://www.1c.ru/rus/partners/service.jsp>.

Информация по «1С:Предприятию»

Адрес сайта: <http://v8.1c.ru/AllInfo>.

Данный сайт представляет собой набор ссылок на часто используемую информацию для следующих категорий пользователей:

- пользователи прикладных решений,
- разработчики прикладных решений,
- партнеры фирмы «1С».

Сайт системы программ «1С:Предприятие»

Адрес сайта: <http://v8.1c.ru/>.

Сайт содержит информацию по технологической платформе системы «1С:Предприятие» и по типовым прикладным решениям, выпущенным фирмой «1С» на ее основе.

Пользовательский сайт

Адрес сайта: <http://users.v8.1c.ru/>.

На сайте поддержки пользователей системы «1С:Предприятие» представлена информация о номерах версий платформы и конфигураций, дате их выхода, выпусках ИТС, на которых опубликовано обновление. По каждой версии представлена следующая информация:

- для технологической платформы:
 - отличия данной версии от предыдущих и особенности перехода;
 - ошибки, исправленные при выпуске данной версии;
 - дистрибутив обновления;
 - файл [readme.htm](#);
- для прикладных решений:
 - новое в релизе;
 - полный список изменений;
 - список основных изменений;

- порядок обновления;
- дистрибутив обновления;
- номер версии платформы, необходимой для использования релиза конфигурации.

Также на пользовательском сайте публикуются рекомендации по администрированию системы «1С:Предприятие».

Основное преимущество, которое дает пользователям «1С:Предприятия» поддержка на данном сайте, – это возможность обновления технологической платформы и прикладных решений через Интернет до получения дисков ИТС.

Также на сайте публикуются дополнительные компоненты, используемые системой «1С:Предприятие» (например, СУБД PostgreSQL), а также тестовые версии платформы и прикладных решений.

ВНИМАНИЕ! К сайту поддержки пользователей имеют право доступа зарегистрированные пользователи программных продуктов системы «1С:Предприятие». При этом пользователи продуктов, для которых предусмотрено обслуживание по линии информационно-технологического сопровождения (ИТС), должны иметь действующую подписку на ИТС.

Зарегистрироваться на сайте поддержки пользователей можно самостоятельно либо обратиться к партнеру фирмы «1С».

Для самостоятельной регистрации необходимо с главной страницы сайта перейти по ссылке [Самостоятельная регистрация пользователей по PIN-коду](#) и выполнить указанные на сайте действия.

В процессе регистрации потребуется PIN-код, входящий в комплект поставки программного продукта.

Если в комплекте поставки отсутствует PIN-код для регистрации на сайте, следует обратиться к партнеру фирмы «1С».

Для регистрации программного продукта необходимо заполнить регистрационную анкету на программный продукт (она является частью регистрационной карточки) и отправить ее в фирму «1С» по почте или факсу.

Для оформления подписки на ИТС можно обратиться к любому из сервис-партнеров фирмы «1С». Список партнеров, имеющих опыт массового обслуживания пользователей в рамках проектов ИТС, опубликован на сайте фирмы «1С»: <http://www.1c.ru/rus/partners/service.jsp>.

Принятые обозначения

Для лучшего понимания излагаемого материала в настоящем Руководстве приняты некоторые общие приемы выделения отдельных элементов текста. Соглашение о таких приемах приведено ниже.

Обозначения клавиш. Клавиши, такие как **Enter**, **Esc**, **Del** и подобные, будут обозначаться, как показано выше, без кавычек.

Для ссылок на клавиши управления курсором (клавиши со стрелками) будет использоваться фраза **клавиши управления курсором**, когда необходимо сослаться сразу на все эти клавиши. Если необходимо упомянуть эти клавиши по отдельности, будут использоваться выражения **Стрелка вверх**, **Стрелка вниз**, **Стрелка вправо** и **Стрелка влево**.

Комбинации клавиш. Когда для выполнения какой-либо команды необходимо нажать комбинацию из двух клавиш, она дается в виде **Ctrl + F3**.

Обозначения кнопок. Наименования кнопок в форме будут даваться их названиями без кавычек, например, **ОК**, **Отмена**, **Удалить** и так далее.

Ключевые слова встроенного языка. Ключевые слова встроенного языка системы «1С:Предприятие» выделяются шрифтом и пишутся так, как в программных модулях: *РабочаяДата*. В тексте также будут встречаться ссылки на описания разделов или элементов встроенного языка (свойства, методы и т. д.). С данными описаниями можно ознакомиться в справке (ветвь [Встроенный язык](#)).

Описание действия с помощью меню. Для описания выбора пункта меню используется следующая конструкция: [Меню](#) – [Подменю](#) – [Подменю](#) – ... – [Пункт](#). Например: «Для выбора масштаба изображения используется пункт [Таблица](#) – [Вид](#) – [Масштаб](#), что эквивалентно тексту: «Для выбора масштаба изображения используется пункт [Масштаб](#) подменю [Вид](#) меню [Таблица](#) главного меню программы». Если выбор осуществляется не из главного меню программы, то это указывается дополнительно.

Режимы работы системы «1С:Предприятие». Система «1С:Предприятие» работает в двух режимах: настройки и проверки конфигурации (далее в Руководстве – режим Конфигуратор или конфигурирование, когда в Руководстве описывается работа по созданию или изменению конфигурации) и исполнения конфигурации (далее в Руководстве – режим 1С:Предприятие).

В данном Руководстве пользователем будет называться специалист, выполняющий разработку или сопровождение конфигурации.

Выражение %APPDATA% означает переменную окружения ОС Windows, которая содержит путь к каталогу (в профиле пользователя), где приложения хранят свои данные. В случае стандартной установки (и для пользователя [Ivanov](#)) этот путь выглядит следующим образом:

[Копировать в буфер обмена](#)

C:\Documents and Settings\Ivanov\Application Data

Для ОС Windows Vista и выше этот путь выглядит следующим образом:

[Копировать в буфер обмена](#)

C:\Users\Ivanov\AppData\Roaming

Выражение %LOCALAPPDATA% означает переменную окружения ОС Windows Vista и старше, содержащую путь к каталогу (в профиле пользователя), в котором находятся данные приложения, специфичные для пользователя. В случае стандартной установки (и для пользователя [Ivanov](#)) этот путь выглядит следующим образом:

[Копировать в буфер обмена](#)

C:\Users\Ivanov\AppData\Local

Глава 1. Концепция системы

«1С:Предприятие» является универсальной системой автоматизации деятельности предприятия. За счет своей универсальности система «1С:Предприятие» может быть использована для автоматизации самых разных участков экономической деятельности предприятия: учета товарных и материальных средств, взаиморасчетов с контрагентами и др.

1.1. Конфигурируемость

Основной особенностью системы «1С:Предприятие» является ее конфигурируемость. Собственно система «1С:Предприятие» представляет собой совокупность механизмов, предназначенных для манипулирования различными типами объектов предметной области. Набор объектов, структуры информационных массивов, алгоритмы обработки информации, соответствующих поставленной задаче, определяет конкретная конфигурация. Вместе с конфигурацией система «1С:Предприятие» выступает в качестве уже готового к использованию программного продукта, ориентированного на определенные типы предприятий и классы решаемых задач.

Конфигурация создается и сопровождается (поддерживается) штатными средствами системы. Конфигурация обычно поставляется в качестве типовой для конкретной области применения, но может быть изменена, дополнена пользователем системы, а также разработана заново. Система «1С:Предприятие» обеспечивает поддержку типовых конфигураций стандартными средствами.

1.2. Функционирование системы

Функционирование системы делится на два процесса – разработка (описание модели предметной области средствами системы) и исполнение (обработка данных предметной области).

На этапе разработки производится:

- **формирование структуры** обрабатываемой **информации**;
- **создание форм** для ввода исходных данных, просмотра различных списков данных;
- организация **хранения** введенной и итоговой **информации**;
- **написание отчетов** и обработок;
- формирование **командных интерфейсов** для различных групп пользователей;
- формирование **списка пользователей**;
- назначение пользователям определенных **прав**.

Результатом разработки является программный продукт (конфигурация), который представляет собой модель предметной области.

В режиме [Конфигуратор](#) можно создавать новые конфигурации, редактировать имеющиеся, а также производить сравнение и объединение нескольких конфигураций.

На этапе разработки система оперирует такими универсальными понятиями (объектами), как документ, журнал документов, справочник, реквизит, форма, регистр и другие. Совокупность этих понятий и определяет концепцию системы. В свою очередь, процесс конфигурирования распадается на несколько составляющих (деление носит условный характер), определяющих последовательность написания и назначение томов описания. Это «визуальное» конфигурирование (создание структуры конфигурации, форм диалогов и выходных документов, механизм работы пользователей с данными (интерфейс) и права доступа различных групп пользователей к различной информации) и написание программ на встроенном языке «1С:Предприятия» для обработки входных и выходных данных.

На уровне системы определены сами понятия объектов и стандартные операции по их обработке. Средства конфигурирования позволяют описать структуру информации, входящей в эти объекты, и алгоритмы, описывающие специфику их обработки, для отражения различных особенностей учета.

Информационная структура проектируется на уровне предусмотренных в системе типов обрабатываемых объектов предметной области (константы, справочники, документы, регистры, перечисления и др.).

В процессе исполнения система уже оперирует конкретными понятиями, описанными на этапе конфигурирования (справочниками товаров и организаций, счетами, накладными и т. д.).

При работе пользователя в режиме 1С:Предприятие обработка информации выполняется как штатными средствами системы, так и с использованием алгоритмов, созданных на этапе конфигурирования.

1.3. Основные понятия системы

В этом разделе рассматриваются основные понятия, которыми оперирует система «1С:Предприятие». Данный раздел будет полезен тем, кто еще не знаком с системой «1С:Предприятие».

Описание тех или иных механизмов будет сопровождаться примерами. Возможно, что в описании будут встречаться еще незнакомые вам понятия и термины. Продолжайте чтение: смысл используемых терминов будет ясен в процессе изложения, а для более подробной информации всегда можно обратиться к соответствующим главам настоящего Руководства.

1.3.1. Понятие «конфигурация»

Основу концепции составляет понятие «конфигурация».

Конфигурацией в системе «1С:Предприятие» называется совокупность взаимосвязанных составных частей:

- подсистемы;
- структуры учетных данных, их форм ввода, выбора, печати;
- состава механизмов учета итоговых данных и движений учетных данных;
- состава различных отчетов и обработок;
- командного интерфейса;
- набора ролей (прав доступа);
- набора общих процедур и функций (модуль приложения, модуль управляемого приложения, модуль внешнего соединения, модуль сеанса, общие модули), макетов табличных документов и др.;
- вспомогательных объектов:
 - функциональных опций и их параметров;
 - хранилищ настроек;
 - средств работы с Web (Web-сервисы, WS-ссылки);
 - различной вспомогательной информации (картинки, шаблоны, стили и т. д.).

Фактически структура конфигурации является моделью предметной области. Создание конфигурации выполняется при помощи конфигуратора. Созданная конфигурация используется системой «1С:Предприятие» для реализации программного окружения, пригодного для выполнения необходимых учетных задач.

Роли в системе «1С:Предприятие» определяют полномочия пользователей на работу с информацией, которая обрабатывается в системе. Совокупность предоставляемых пользователю полномочий определяется, как правило, кругом его обязанностей.

Операция назначения ролей пользователю решает две основные задачи:

- с одной стороны, ограничивается круг пользователей конфиденциальной информации, которая, безусловно, всегда присутствует в любой системе учета;
- с другой стороны, запрет выполнения определенных операций (в первую очередь операций удаления и корректировки данных) позволяет в какой-то степени предотвратить возможные потери информации.

Все составные части конфигурации тесно связаны между собой и требуют, как правило, согласованного внесения изменений (особенно это касается пользовательских прав).

Так, назначение ролей может выполняться только для существующих объектов конфигурации (конкретных документов, журналов, справочников, отчетов). Добавление в структуру конфигурации нового объекта должно сопровождаться внесением соответствующих изменений в роли.

Система учитывает назначение прав на объекты при построении командного интерфейса. Если, например, пользователю запрещен просмотр какого-либо справочника, то команда открытия формы списка этого справочника будет удалена из командного интерфейса автоматически. Формы также автоматически учитывают наличие прав при отображении форм.

1.3.2. Объект конфигурации

Под объектом конфигурации в системе «1С:Предприятие» понимается формальное описание группы понятий (предметной области, средств взаимодействия пользователя с системой) со сходными характеристиками и одинаковым предназначением.

Приведем такой пример. Объект конфигурации **Справочник** в системе «1С:Предприятие» предназначен для ведения списков однородных элементов данных – справочников, картотек, нормативных сборников и т. п. Использование объектов конфигурации этого типа позволяет организовать ведение любых справочников, необходимых для автоматизации деятельности предприятия.

Как правило, объекты конфигурации типа **Справочник** являются компьютерными аналогами реально существующих на предприятии видов справочников, например, справочника сотрудников или номенклатуры товаров, хотя могут использоваться и для организации списков, не имеющих явных физических аналогов.

Следует иметь в виду, что объект конфигурации описывает не конкретное значение, а только его вид. Например, справочник **Физические лица** описывает не конкретного человека, а содержит перечень реквизитов (набор видов характеристик о физическом лице), а также формы для ввода их значений, формы просмотра списков и макеты для печати информации. Другими словами, в конфигурации создается схема описания, с помощью которой учитываются все однородные объекты предметной области (в приведенном примере справочника **Физические лица** одно описание используется как для Петрова, Иванова, так и для любого другого физического лица).

Реализованный в системе «1С:Предприятие» при помощи объекта конфигурации компьютерный аналог конкретного понятия предметной области будем называть объектом конфигурации.

1.3.2.1. Свойства объекта конфигурации

Каждый объект конфигурации обладает уникальным набором свойств. Этот набор описан на уровне системы и не может быть изменен в процессе настройки конфигурации задачи. Набор свойств объекта конфигурации определяется в основном его назначением в системе «1С:Предприятие».

Главным свойством любого объекта конфигурации является имя – краткое наименование объекта конфигурации. При создании нового объекта конфигурации ему автоматически присваивается условное

имя, состоящее из слова, определяемого по виду объекта, и цифры (например, при создании реквизита создается реквизит с именем **Реквизит1**, при создании документа – **Документ1** и т. д.). Это имя можно изменить в процессе редактирования свойств объекта конфигурации, при этом система отслеживает уникальность имен. Имя объекта конфигурации не может быть пустым и быть длиннее 255 символов.

Некоторые свойства из всего набора свойств, присущих объекту конфигурации, доступны для редактирования и могут быть так или иначе изменены в процессе конфигурирования системы. Характер изменений и их пределы также задаются на уровне системы. Специалист, осуществляющий конфигурирование системы, целенаправленным изменением свойств объекта конфигурации может добиться требуемого поведения объекта при работе системы. Однако такие изменения не затрагивают сущности объекта и не позволяют добиться от него действий, не свойственных объектам данного типа.

Приведем такой пример.

Объект конфигурации **Константа** в системе «1С:Предприятие» предназначен для хранения информации, которая не изменяется во времени или изменяется очень редко. При этом не важны предыдущие значения константы. Простым примером константы может служить название предприятия: оно, как правило, не меняется в процессе деятельности предприятия (если предполагается, что значения каких-либо учетных данных, изменяемых во времени, нужно выбирать с учетом времени, то для таких данных необходимо использовать не константу, а регистр сведений без измерений).

Константа обладает большим набором редактируемых свойств, из которых наиболее важными являются:

- имя константы,
- синоним,
- комментарий,
- тип данных,
- режим управления блокировкой,
- ссылка, позволяющая открыть модуль менеджера константы.

В наиболее общем случае значение в константу вводится один раз (как, например, название предприятия). С точки зрения использования константы не важно, что именно хранится в константе; главным является то, что константа сохранила записанное в нее значение.

Способность сохранять введенное в нее значение – неотъемлемая особенность константы в системе «1С:Предприятие». Редактирование свойств константы на эту способность не влияет.

1.3.2.2. Основные виды объектов конфигурации

Все объекты конфигурации, которые существуют в системе «1С:Предприятие», образуют несколько основных видов. Каждый вид объектов конфигурации представляет собой как раз те «строительные элементы», из которых будет создаваться конфигурация.

Формально объекты конфигурации объединяются в виды в дереве конфигурации. Названия видов пользователь видит на первом уровне дерева конфигурации, когда открывает окно **Конфигурация** в конфигураторе.

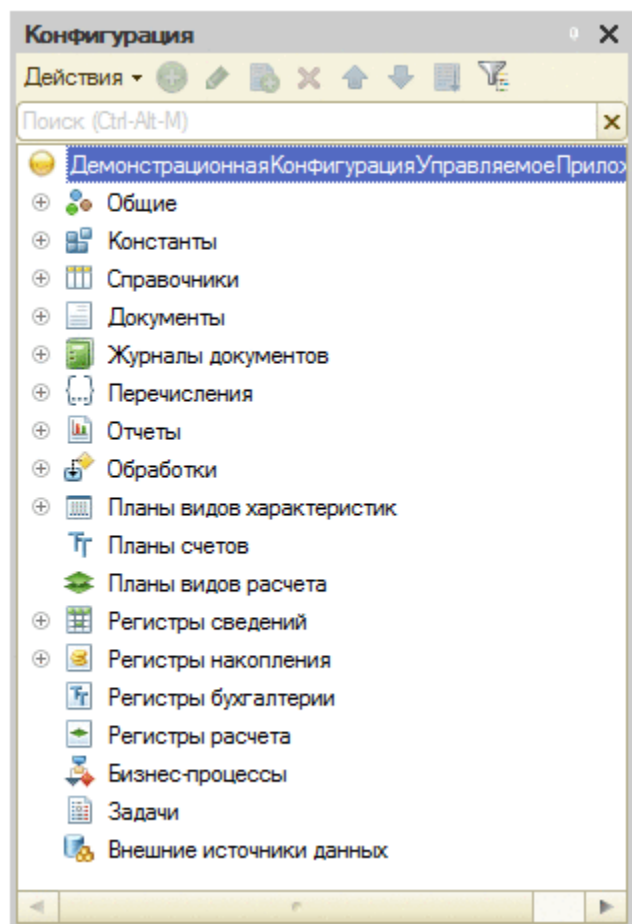


Рис. 1. Дерево метаданных

Несмотря на отсутствие формального определения, названия видов объектов конфигурации широко используются при работе с системой «1С:Предприятие».

Например, специалист, осуществляющий конфигурирование системы «1С:Предприятие», видит свою цель в разработке необходимого набора справочников, документов, отчетов, журналов, которые будут реализовывать требуемую систему учета. Конечный пользователь системы «1С:Предприятие» – руководитель, бухгалтер, менеджер, кладовщик – также оперирует конкретными справочниками, документами и т. д. для решения стоящих перед ним задач. Общение между двумя этими категориями пользователей также будет происходить в понятиях видов объектов конфигурации.

Объект данных какого-либо вида является уже конкретным документом, отчетом, журналом, константой и так далее. Как правило, каждый объект используется для работы со вполне определенной информацией предметной области.

Ниже приведена краткая характеристика основных видов объектов конфигурации системы «1С:Предприятие». Подробная информация об объектах конфигурации, объединяемых в каждом из этих видов, будет изложена далее.

1.3.2.2.1. Константы

Для работы с постоянной и условно постоянной информацией в системе используются объекты типа **Константа**. Информация, хранящаяся в константах, редко изменяется, но, как правило, часто используется в работе. Например, в константах может храниться наименование предприятия, его ИНН, фамилии директора и главного бухгалтера и другая подобная информация.

В системе может быть описано неограниченное количество констант.

1.3.2.2.2. Справочники

Для работы с постоянной и условно постоянной информацией с некоторым множеством значений в

системе используются объекты типа [Справочник](#).

Обычно справочниками являются списки материалов, товаров, организаций, валют, сотрудников и др.

Механизм поддержки справочников позволяет спроектировать и поддерживать самые различные справочники. На этапе конфигурирования можно описать, какими свойствами обладает каждый конкретный справочник. К настраиваемым свойствам относятся, например, длина и тип кода, количество уровней иерархии, поддержка уникальности кодов, набор реквизитов справочника.

Помимо кода и наименования, механизм работы со справочниками позволяет создавать набор реквизитов для хранения любой дополнительной информации об элементе справочника (например, для номенклатуры это может быть закупочная и отпускная цены, производитель, для сотрудника – должность, образование, адрес места жительства и т. д.), а также табличные части. В табличных частях хранится однотипная информация, число которой может быть переменным, например, описание комплектующих изделия, состав семьи сотрудника, телефоны организации и т. д.

Для каждого справочника может быть задано несколько типов форм: элемента, группы, списка, выбора, выбора группы. По каждому типу форм может быть создано произвольное число форм.

Для описания соподчиненных сущностей можно использовать подчиненные справочники. В этом случае в подчиненном справочнике каждый элемент «принадлежит» к определенному элементу справочника-владельца.

В конкретной конфигурации создается необходимое количество справочников для хранения данных об объектах, используемых при автоматизации данной предметной области. Например, это могут быть справочники [Организации](#), [Товары](#), [Сотрудники](#) и т. д.

1.3.2.2.3. Перечисления

Перечисления используются в системе «1С:Предприятие» для описания постоянных наборов значений, не изменяемых в процессе работы конфигурации.

На этапе конфигурирования можно описать практически неограниченное количество видов перечислений. В отличие от справочника, значения перечислений задаются на этапе конфигурирования и не могут быть изменены на этапе исполнения.

Типичными примерами перечислений являются виды оплаты (наличная, безналичная, бартер), статус клиента (постоянный, разовый) и т. д.

Одной из главных особенностей перечислений, отличающей их от справочников, является то, что набор значений перечисления не изменяется при работе конечного пользователя с программой. Например, алгоритм конфигурации может быть ориентирован на то, что каждый клиент имеет один из двух статусов – либо постоянный, либо разовый. В этом случае указание статуса клиента выполняется путем выбора одного из значений перечисления. Пользователь не может добавить новый статус.

В отличие от перечислений, для справочников конкретные значения обычно вводятся пользователем при работе с программой, например: наименования товаров, контрагентов и прочее.

1.3.2.2.4. Документы

Документы предназначены для отражения хозяйственных событий предприятия, которые имеют отношение к автоматизируемой предметной области. Например, в конфигурации, предназначенной для учета торговых операций, могут быть такие документы, как счет, приходная накладная, расходная накладная и проч. При помощи документов отражаются и платежи с расчетного счета, и операции по кассе, и движения по складу, и прочие подобные события.

В процессе конфигурирования настраивается произвольное количество видов документов. Типичными примерами видов документов являются такие, как [Платежное поручение](#), [Счет](#), [Приходная накладная](#), [Расходная накладная](#), [Накладная на внутреннее перемещение](#), [Приходный кассовый ордер](#) и другие. Каждый вид документа предназначен для отражения своего типа событий. Это определяет его структуру

и свойства, которые описываются в конфигурации.

Каждый вид документа может иметь неограниченное количество реквизитов и табличных частей. Несколько табличных частей требуются в тех случаях, когда одним документом необходимо зарегистрировать разные по сути, но связанные события, например: отразить поступление товара на склад и зарегистрировать понесенные дополнительные затраты – оплату транспорта, грузчиков и др.

Для документа создаются формы ввода – экранные аналоги реальных документов. Если в других формах используются данные документов, то для включения этой информации разрабатываются формы для выбора. Для просмотра списка документов одного вида создаются формы списков. Число форм неограниченно.

Каждый документ также может иметь неограниченное число печатных форм.

Все документы характеризуются номером, датой и временем. При настройке для документа также задается длина номера документа, условия поддержки уникальности номеров и другие.

Документы играют центральную роль для основных механизмов, реализуемых системой. Все документы образуют единую хронологическую последовательность. Фактически она отражает реальную последовательность событий. Внутри даты последовательность документов определяется их временем, при этом время документа является не столько средством отражения реального (астрономического) времени ввода документа, сколько средством, позволяющим четко упорядочить документы внутри одной даты. Данные, вводимые в документ (в реквизиты и табличные части документа), обычно содержат информацию о происшедшем событии: например, в накладной – информацию о том, с какого склада, каких товаров и сколько отгружено, какие дополнительные затраты произведены при приобретении товаров.

Для документа весьма важным действием является его проведение. Если документ не является «проводимым», это значит, что событие, которое он отражает, не влияет на состояние учета, который ведется в данной конфигурации. Если документ проводится, то он изменяет состояние тех или иных учитываемых данных. При проведении документ может отразить зафиксированное им событие в механизмах, реализуемых различными регистрами.

Например, в торговом предприятии выписка клиенту счета на оплату не изменяет состояния товарных или денежных средств предприятия, так как счет в данном случае – это только выражение намерения клиента приобрести товар. В этом случае в конфигурации для учета торговых операций документ **Счет** может не отражаться на регистрах учета.

Однако если выписка счета сопровождается резервированием товара для данного клиента, то в этом случае документ **Счет** должен отражаться на регистрах учета, так как операция выписки счета еще и «замораживает», временно выключает из оборота определенное количество товаров. В этом случае конфигурация для учета торговых операций должна уметь отслеживать зарезервированный товар.

1.3.2.2.5. Журналы документов

Журналы документов предназначены для просмотра документов разных видов. Каждый вид документа может быть показан в нескольких журналах. Журнал документов не добавляет новые данные в систему, а является средством для отображения в едином списке документов нескольких видов.

Например, может быть создан журнал **Складские документы**, в котором будут отображаться все приходные и расходные накладные и накладные на внутреннее перемещение.

Для журнала могут быть определены графы журнала, предназначенные для отображения реквизитов документов разного вида, отнесенных к данному журналу. Например, журнал торговых документов может содержать графу **Контрагент**, в которой будет отражаться реквизит **Комитент** документа **Прием на комиссию**, реквизит **Организация** документа **Приходная накладная** и т. д.

Каждый журнал может иметь неограниченное число форм визуального представления и печатных форм.

1.3.2.2.6. Отчеты и обработки

Для описания отчетов и процедур обработки информации на этапе конфигурирования может быть создано неограниченное число отчетов и обработок. Отчеты и обработки могут иметь несколько форм, предназначенных, например, для ввода параметров формирования отчета или параметров обработки данных. Например, для выдачи складской справки – выбор конкретного склада.

Алгоритм получения отчета может описываться с использованием встроенного языка или формироваться системой автоматически, в случае использования системы компоновки данных (см. [здесь](#)). Для вывода отчетов может быть использован как текстовый формат, так и специализированный табличный формат отчетов (макеты).

Система также поддерживает возможность разработки внешних обработок, хранящихся не в самой конфигурации, а в отдельных файлах.

1.3.2.2.7. Планы видов характеристик

В системе «1С:Предприятие» объекты **Планы видов характеристик** предназначены для описания множеств однотипных объектов аналитического учета.

1.3.2.2.8. Планы видов расчета

Объекты данного вида предназначены для создания видов расчетов, используемых в механизмах периодических расчетов.

1.3.2.2.9. Планы счетов

План счетов является одним из основных понятий бухгалтерского учета. Планом счетов называется совокупность синтетических счетов, предназначенных для группировки информации о хозяйственной деятельности предприятия. Информация, накапливаемая на таких синтетических счетах, позволяет получить полную картину состояния средств предприятия в денежном выражении.

1.3.2.2.10. Планы обмена

Объекты данного вида предназначены для организации обмена данными между различными информационными базами, а также информационными базами и внешними программными системами.

1.3.2.2.11. Бизнес-процессы и задачи

Позволяют создавать формализованные описания типичных последовательностей работ, выполняемых в организации, и на их основе формировать списки задач, которые необходимо выполнить тому или иному сотруднику организации в данный момент. Например, процесс продажи товара может быть представлен как последовательность выписки счета, его утверждения, получения наличной оплаты и отгрузки товара со склада. За выполнение каждого из этапов могут отвечать различные сотрудники. Таким образом, в любой момент времени можно определить, в каком состоянии находится процесс продажи товара и кто из сотрудников в данный момент должен выполнить какие-либо действия.

1.3.2.2.12. Регистры

Регистры предназначены для хранения и обработки различной информации, отражающей хозяйственную или организационную деятельность предприятия и не имеющей объектной природы.

В регистрах обычно хранится информация об изменении состояний объектов или другая информация, не отражающая непосредственно объекты предметной области. Например, в регистрах может храниться информация о курсах валют или информация о приходе и расходе товаров.

В системе «1С:Предприятие» существует 4 вида регистров:

- регистры сведений,
- регистры накопления,

- регистры расчетов,
- регистры бухгалтерии.

1.3.2.2.13. Специализированные объекты конфигурации (ветвь «Общие»)

Помимо объектов, описывающих предметную область учета, конфигурация содержит ряд вспомогательных объектов, не относящихся непосредственно к деятельности предприятия, однако тесно связанных с функционированием самой системы. Это механизмы взаимодействия пользователей с системой «1С:Предприятие» (командный интерфейс, критерии отбора, права доступа различных групп пользователей к различной информации); вспомогательные объекты оформительского назначения, позволяющие производить конфигурирование на основе сформированных стилей; библиотеки картинок с учетом национального языка; модуль приложения и общие модули, в которых располагаются процедуры и функции, доступные из прочих модулей конфигурации; общие макеты печатных форм и многое другое.

1.3.2.3. Подчиненные группы объектов

В зависимости от вида объекта конфигурации объект может иметь различные подчиненные группы объектов. Например, реквизиты, измерения, формы, табличные части и т. д. Состав подчиненных объектов зависит от типа объекта.

Реквизиты – дополнительная информация об объекте, доступная только в пределах этого объекта.

Табличные части – наборы дополнительной информации об объекте, представленной в виде таблицы.

ВНИМАНИЕ! Число строк одной табличной части не может быть более 100 000.

Реквизиты табличных частей – состав табличной части объекта, доступный только в пределах табличной части объекта.

Формы – форма используется для ввода, просмотра и редактирования информации, хранящейся в объекте конфигурации, содержит модуль формы – программу на встроенном языке системы «1С:Предприятие». Способность иметь визуальное представление позволяет объекту конфигурации организовать интерактивное взаимодействие с пользователем. Характер такого взаимодействия разрабатывается специалистом, осуществляющим конфигурирование системы «1С:Предприятие», и определяется в основном типом объекта конфигурации. Для разработки форм в конфигураторе применяется комплексный редактор форм, позволяющий редактировать все компоненты формы во взаимосвязи. Каждый объект может иметь несколько форм.

Команды – используются для выполнения различных операций с объектом. Команды бывают независимыми и параметризуемыми.

Макеты – табличные, HTML или текстовые документы (также могут использоваться двоичные и Active-документы), предназначенные для формирования печатных форм объекта.

Графы – графы журнала документов.

Измерения – для регистров это объекты конфигурации, данные о которых учитываются в регистре.

Ресурсы – данные, учитываемые в регистре.

Группы подчиненных объектов не удаляются и не имеют редактируемых свойств.

1.3.2.4. Типизированные и типобразующие объекты

Одним из основных свойств некоторых объектов конфигурации является **Тип данных**. Это свойство определяет, какого рода информацию может содержать объект конфигурации. Тип данных объекта конфигурации назначается при создании или редактировании свойств объекта в процессе настройки конфигурации.

Объекты конфигурации, для которых может быть указан тип информации, содержащейся в объекте, в системе «1С:Предприятие» называются типизированными объектами конфигурации.

Такие объекты конфигурации, как [Справочник](#), [Документ](#), [Обработка](#), не являются типизированными объектами, так как содержат «комплексную» информацию и, в свою очередь, включают в себя типизированные объекты конфигурации.

Типы данных, которые может принимать объект конфигурации, можно разделить на две группы.

Первую группу составляют примитивные типы данных: [Число](#), [Строка](#), [Дата](#) и [Булево](#). Соответственно, информация, хранящаяся в объекте конфигурации, может быть числом, произвольной строкой символов, датой или логической величиной. Кроме этих типов к примитивным типам относят [NULL](#), [Неопределено](#) и [Тип](#) (подробнее см. раздел «Примитивные типы данных» справки по встроенному языку).

Кроме этого, некоторые объекты конфигурации системы «1С:Предприятие» также могут образовывать типы **данных**. Например, константе может быть назначен тип данных [ДокументСсылка](#). В этом случае значение константы будет представлять собой ссылку на один из существующих в системе «1С:Предприятие» документов.

Объекты конфигурации, которые могут образовывать типы значений конфигурации, в системе «1С:Предприятие» называются типообразующими объектами конфигурации. Такими объектами в системе «1С:Предприятие» являются:

- справочники,
- документы,
- планы видов характеристик,
- планы счетов,
- планы видов расчета,
- планы обмена,
- бизнес-процессы,
- задачи,
- перечисления.

Необходимо обратить внимание, что типообразующие объекты конфигурации образуют тип данных сразу после создания в конфигураторе объекта любого из таких типов. При этом появляются сразу три новых вида типов: [Ссылка](#), [Объект](#) и [Список](#). Например, когда в конфигураторе создается новый справочник, то в списке типов данных появляются новые типы данных: [СправочникСсылка.<ИмяСправочника>](#), [СправочникОбъект.<ИмяСправочника>](#) и [СправочникСписок.<ИмяСправочника>](#). Такие типы данных могут быть присвоены любому из типизированных объектов конфигурации.

Некоторые данные могут иметь составной тип. Для этого в окне редактирования типа данных нужно установить флажок [Составной тип данных](#) и указать те типы, которые могут принимать данные. Кроме того, допускается выбор специального типа [ЛюбаяСсылка](#).

При выборе типа данных реквизита система, помимо выбора типов, определенных в конкретном прикладном решении, предоставляет разработчику возможность выбирать наборы типов. Наборами типов, например, являются [ЛюбаяСсылка](#), [СправочникСсылка](#), [Характеристика.<имя>](#) и др.

Наборы типов, так же как и составной тип данных, содержат некий перечень типов, определенных в данном прикладном решении, однако, в отличие от составного типа, этот перечень формируется системой автоматически, в результате анализа метаданных.

Например, в прикладном решении имеются справочники *Номенклатура* и *Контрагенты*. Если определен реквизит составного типа данных, в который входят типы *СправочникСсылка.Номенклатура* и *СправочникСсылка.Контрагенты*, то наряду с этим можно определить реквизит, содержащий набор типов *СправочникСсылка*. И в том и в другом случае можно хранить в реквизите ссылки как на справочник *Номенклатура*, так и на справочник *Контрагенты*.

После добавления нового справочника *Цены* в реквизите составного типа по-прежнему могут храниться только ссылки на справочники *Номенклатура* и *Контрагенты*, а в реквизите, описанном как набор типов, допускается хранение ссылки на любой из справочников, доступных в данной конфигурации, в том числе и на справочник *Цены*.

При запуске прикладного решения набор типов преобразуется системой, как правило, в составной тип, содержащий все типы, которые должны входить в этот набор. Поэтому во втором случае в набор типов попадет и новый справочник *Цены*.

Однако набор типов не всегда преобразуется системой в составной тип данных. Если оказывается, что в набор типов входит единственный тип значений, то набор типов будет преобразован в этот самый тип значений. Такая ситуация возможна, например, когда план видов характеристик (назовем его *Свойства*) имеет единственный тип значений в свойстве *ТипЗначенияХарактеристик*. Тогда набор типов *Характеристика.Свойства* будет преобразован системой не в составной тип данных, содержащий один тип значений, а в тот единственный тип значений, который указан для плана видов характеристик.

Эта особенность может быть важна, когда, например, выполняется проверка реквизита, тип которого описан как *Характеристика.Свойства*, на заполненность. Когда *Характеристика.Свойства* преобразуется системой в составной тип данных, проверять нужно на значение *Неопределено*, а если *Характеристика.Свойства* преобразуется в определенный тип значения, то проверять нужно на значение по умолчанию данного типа.

1.3.3. Командный интерфейс

Командный интерфейс – это основное средство навигации пользователя по функциональности конфигурации. Командный интерфейс строится на основе подсистем. Разработчик конфигурации включает прикладные объекты в соответствующие подсистемы.

На основе этой информации (структуры подсистем и привязки объектов к подсистемам) система автоматически строит командный интерфейс для пользователя. Пользователю отображается структура прикладного решения (иерархия подсистем) и предоставляются стандартные команды доступа к функциональности прикладных объектов (вызов списков справочников, документов, открытие отчетов, обработок и т. д.). Однако разработчик, разумеется, может отредактировать предлагаемое системой построение командного интерфейса (изменить порядок, видимость команд). Для этого предназначен редактор командного интерфейса, который вызывается как для конкретной подсистемы, так и для всех подсистем.

Сами команды, включаемые в командный интерфейс (открытие списков, ввод новых объектов, открытие отчетов и т. д.), предоставляются системой автоматически. Но разработчик может создать свои команды, которые будут включаться в командный интерфейс.

Цель создания интерфейса – обеспечить структурированный доступ пользователей к той информации, которая необходима им в соответствии с их обязанностями.

1.3.4. Форма

Совокупность экранного диалога, модуля, реквизитов и команд называется **формой**.

Большинство объектов конфигурации в системе «1С:Предприятие» могут иметь визуальную форму. В самом общем случае форма как объект конфигурации состоит из следующих частей:

- Экранный диалог, используемый для ввода и редактирования информации.

- Модуль формы – программа на встроенном языке системы «1С:Предприятие». Как правило, модуль формы выполняет обработку вводимой в диалог информации для целей входного контроля, выполнения расчетов и т. д.
- Список реквизитов.
- Команды, используемые в форме.

Любая из этих составных частей формы может отсутствовать, то есть не содержать информации.

С помощью формы можно реализовать интерактивное взаимодействие прикладного объекта с пользователем. Характер такого взаимодействия разрабатывается специалистом, осуществляющим конфигурирование системы «1С:Предприятие». Подробнее об устройстве формы см. [здесь](#).

Для разработки форм в конфигураторе применяется редактор форм, позволяющий редактировать все компоненты формы во взаимосвязи.

1.3.5. Модуль

Модулем называется программа на встроенном языке системы «1С:Предприятие». Модули располагаются в заданных точках структуры конфигурации и вызываются для выполнения в заранее известные моменты работы системы «1С:Предприятие». Специалист, выполняющий конфигурирование системы, может использовать модули для описания сложных алгоритмов взаимодействия объектов конфигурации, для которых недостаточно имеющихся в конфигураторе визуальных средств.

В конфигурации существует несколько видов модулей. Это модуль управляемого приложения, модуль обычного приложения, модуль внешнего соединения, модуль сеанса, общие модули, модули форм и модули объектов конфигурации (менеджеров значения констант, справочников, документов, планов видов характеристик, планов счетов, планов видов расчета, планов обмена, бизнес-процессов, задач, отчетов, обработок, наборов записей регистров), модули менеджеров объектов конфигурации (справочников, документов, планов видов характеристик, планов счетов, планов видов расчета, планов обмена, бизнес-процессов, задач, отчетов, обработок, регистров сведений, регистров накопления, регистров бухгалтерии, регистров расчета, перечислений, журналов документов, хранилищ настроек), модули наборов записей (регистров сведений, регистров накопления, регистров бухгалтерии, регистров расчета), модули команд.

Для доступа к модулю необходимо в контекстном меню объекта конфигурации выбрать пункт [Открыть модуль...](#) Для корневого объекта конфигурации выбирается модуль управляемого приложения, модуль сеанса, модуль внешнего соединения и модуль обычного приложения. Некоторые объекты (например, константы, журналы документов) не имеют модуля.

Подробное описание назначений модулей см. в разделе «Что такое программный модуль?» справки по встроенному языку.

В модулях объектов возможно объявление переменных, процедур и функций, которые будут доступны при работе с объектом извне во встроенном языке, дополняя контекст объекта. В этих модулях располагают процедуры обработки различных событий, например, ввода на основании. Также в них располагают различные процедуры, с помощью которых выполняются действия над объектом, инициированные вне данного объекта (например, выполнение печати).

Модуль менеджера позволяет расширить функциональность менеджеров, предоставляемых системой, за счет написания процедур и функций на встроенном языке. Фактически это позволяет описать методы для объекта конфигурации (например, справочника), которые относятся не к конкретному экземпляру объекта базы данных, а к самому объекту конфигурации. Модуль менеджера не может иметь переменных и тела модуля.

Если функции или процедуры модуля менеджера объявлены как экспортируемые, к ним можно будет получить доступ через менеджер объекта:

```
Функция ПолучитьСписокДебиторов ( )
```

```
...
```

```
КонецФункции
```

```
// Вызов из прикладного кода.
```

```
Дебиторы = Справочники.Контрагент.ПолучитьСписокДебиторов ( );
```

1.3.6. Макет

Макетом в системе программ «1С:Предприятие» называется объект конфигурации, предназначенный для формирования печатных форм.

Общие макеты печатных форм располагаются в ветви **Макеты** ветви **Общие** дерева конфигурации: печатные формы объектов конфигурации (справочников, документов, журналов документов, планов счетов, планов видов характеристик, планов видов расчетов, регистров, отчетов и обработок и других объектов) располагаются в подчиненных объектах **Макеты**, а также во внешних файлах (в этом случае должно быть установлено свойство табличного документа **Макет**).

Макеты могут быть следующего типа:

- **Табличный документ** – предполагает использование стандартной технологии создания и использования макетов. Подготовка макета производится с помощью табличного редактора.
- **Двоичные данные** – используются двоичные данные.
- **ActiveDocument** – предполагает использование технологии OLE Active document.
- **HTML-документ** – предполагает использование редактора HTML-документа.
- **Текстовый документ** – предполагает использование текстового документа в качестве макета. Подготовка текстового макета производится с помощью редактора текстовых макетов.
- **Географическая схема** – предполагает использование географической схемы, подготовленной в редакторе географических схем, в качестве макета.
- **Графическая схема** – предполагает использование подготовленной в редакторе графической схемы.
- **Схема компоновки данных** – предполагает использование схемы компоновки данных, подготовленной в конструкторе.
- **Макет оформления компоновки данных** – предполагает использование макета оформления системы компоновки данных.

1.4. Варианты работы

«1С:Предприятие» поддерживает два варианта работы:

- файловый,
- клиент-серверный.

И в том и в другом варианте все прикладные решения работают полностью идентично. Файловый вариант работы в основном предназначен для персонального использования, в то время как клиент-серверный вариант – для использования в рабочих группах или в масштабе предприятия.

1.4.1. Файловый вариант

Файловый вариант работы с информационной базой рассчитан на персональную работу одного

пользователя или работу небольшого количества пользователей в локальной сети. В этом варианте все данные информационной базы (конфигурация, база данных, административная информация) располагаются в одном файле.

Такой вариант работы обеспечивает легкость установки и эксплуатации автоматизированной системы. При этом для работы с информационной базой не требуются дополнительные программные средства, достаточно иметь операционную систему и «1С:Предприятие».

Файловый вариант «1С:Предприятия» обеспечивает высокую целостность информационной базы и простое создание резервных копий. Исключена ситуация, когда пользователь может по ошибке (например, при копировании информационной базы) перепутать различные файлы информационной базы и привести таким образом систему в неработоспособное состояние.

Кроме этого, резервное копирование может осуществляться на файловом уровне, путем простого копирования файла информационной базы.

Однако, несмотря на легкость и простоту использования, файловый вариант обладает некоторыми ограничениями (см. [здесь](#)).

1.4.2. Клиент-серверный вариант

Клиент-серверный вариант предназначен для использования в рабочих группах или в масштабе предприятия. Он реализован на основе трехуровневой архитектуры «клиент-сервер».

Программа, работающая у пользователя (клиентское приложение), взаимодействует с кластером серверов «1С:Предприятия», а кластер, при необходимости, обращается к серверу баз данных (Microsoft SQL Server, PostgreSQL, IBM DB2 или Oracle Database). При этом физически кластер серверов «1С:Предприятия» и сервер баз данных могут располагаться как на одном компьютере, так и на разных. Это позволяет администратору при необходимости распределять нагрузку между серверами.

Использование кластера серверов «1С:Предприятия» позволяет сосредоточить на нем выполнение наиболее объемных операций по обработке данных. Например, при выполнении даже весьма сложных запросов программа, работающая у пользователя, будет получать только необходимую ей выборку, а вся промежуточная обработка будет выполняться на сервере. Обычно увеличить мощность кластера серверов гораздо проще, чем обновить весь парк клиентских машин.

Другим важным аспектом использования 3-уровневой архитектуры является удобство администрирования и упорядочивание доступа пользователей к информационной базе. В этом варианте пользователь не должен знать о физическом расположении конфигурации или базы данных. Весь доступ осуществляется через кластер серверов «1С:Предприятия». При обращении к той или иной информационной базе пользователь должен указать только имя кластера и имя информационной базы, а система запрашивает соответственно имя и пароль пользователя. Подробнее с администрированием системы можно ознакомиться в книге «1С:Предприятие 8.3. Руководство администратора».

Несмотря на то, что система «1С:Предприятие» старается скрыть от пользователя особенности поведения различных серверов баз данных, это не всегда удается. Особенности работы системы с тем или иным сервером баз данных см. [здесь](#).

Важной особенностью работы в клиент-серверном варианте является возможность работы сервера «1С:Предприятия» и серверов баз данных на различных операционных системах (семейство Windows и различные дистрибутивы Linux).

1.5. Технологические средства разработки

Для описания специфических алгоритмов обработки информации и создания интерфейса, ориентированного на удобное представление описанных в конфигурации данных, в системе «1С:Предприятие» используются несколько технологических механизмов.

Встроенный программный язык. Необходимость наличия встроенного языка определена концепцией настраиваемости системы. Синтаксис встроенного языка вполне отвечает стандартам высокоуровневых

языков.

Язык является предметно-ориентированным. Он поддерживает специализированные типы данных предметной области, определяемые конфигурацией системы. Работа с этими типами данных в языке организована с использованием объектной техники. Язык ориентирован на пользователей различной квалификации. В частности, его отличает мягкая типизация данных, обеспечивающая быстрое написание программных модулей, и жесткий контроль синтаксических конструкций, уменьшающий вероятность ошибок.

Так как система сочетает в себе визуальные и языковые средства конфигурирования, использование встроенного языка в системе имеет событийно-зависимую ориентацию, то есть языковые модули используются в конкретных местах для отработки отдельных алгоритмов, настраиваемых в процессе конфигурации. Так, например, для документа можно описать алгоритм автоматического заполнения реквизитов при вводе нового документа. Данная процедура будет вызвана системой в нужный момент.

Механизм запросов. Для получения произвольных отчетов сложной структуры в системе предусмотрен предметно-ориентированный механизм запросов. Данное средство опирается на существующую условно-переменную структуру информационной базы системы, что позволяет сравнительно просто описывать достаточно сложные запросы.

Встроенный текстовый редактор используется для создания программных модулей на встроенном языке и для редактирования документов в текстовом виде.

Одной из особенностей редактора является возможность контекстного выделения цветом синтаксических конструкций встроенного языка, а также группировка различных синтаксических конструкций.

При наборе текстов на встроенном языке удобно пользоваться контекстной подсказкой и шаблонами.

Благодаря тому, что встроенный язык системы имеет мощные средства манипулирования текстами, текстовый формат может быть успешно использован для обмена с другими системами самой различной информацией.

Встроенный редактор форм. Работа с настраиваемыми структурами данных и работа в интерфейсе операционной системы Microsoft Windows вызывает необходимость произвольной настройки форм для ввода и редактирования информации. Для этого в системе «1С:Предприятие» существует встроенный редактор форм.

Редактор позволяет оформить большинство окон, которые используются в системе для ввода и просмотра предметной информации (формы документов, справочников, настройки отчетов).

Встроенный редактор табличных документов. Для всех выходных документов (первичных документов и отчетов) в системе предусмотрен единый формат – формат табличных документов.

Редактор табличных документов – это мощное средство, сочетающее в себе оформительские возможности табличной структуры и векторной графики. Он может быть использован как для создания небольших документов с очень сложной структурой линий (типа платежного поручения), так и для объемных ведомостей, журналов и других подобных документов.

Редактор табличных документов предоставляет пользователям богатый набор оформительских возможностей (шрифты, цвета, линии, узоры). Имеется возможность вывода информации в графическом виде (диаграммы).

Одной из главных особенностей редактора табличных документов является ориентация на формирование отчетов при помощи встроенного языка системы «1С:Предприятие». Гибкое построение отчетов с его помощью становится возможным благодаря наличию механизма манипулирования именованными областями документа. Редактор табличных документов позволяет манипулировать не только горизонтальными, но и вертикальными областями, что делает возможным создание отчетов, масштабируемых не только в высоту, но и в ширину. Сочетание возможностей редактора с таким объектом, как система компоновки данных, позволяет создавать универсальные отчеты, которые дают возможность обрабатывать и представлять информацию в различных разрезах и различной детализации без дополнительного вмешательства разработчика.

С другой стороны, табличный документ может выступать в качестве элемента управления формы и таким образом использоваться для ввода данных.

Встроенный редактор картинок. Редактор позволяет создавать картинки произвольных размеров для использования их в качестве пиктограмм панели инструментов, картинок кнопок и других оформительских целей.

Встроенный редактор HTML-документов. Редактор позволяет создавать пользовательские описания и имеет большие оформительские возможности (механизм гиперссылок, использование стилей, размещение картинок и т. д.).

Конструкторы – вспомогательные инструменты, облегчающие разработку стандартных элементов системы «1С:Предприятие». В системе имеются, например, конструкторы форм констант, справочников, документов, журналов документов, отчетов и других объектов, конструкторы печатных форм, конструкторы движений регистров и другие.

С помощью конструкторов производится не только формирование визуальных составляющих этих объектов, но и в некоторых случаях (ввод на основании, печать, выходная форма и др.) формируются программные модули.

Система настройки пользовательского интерфейса. Для того чтобы интерфейс конкретной конфигурации системы полностью отражал настроенные структуры данных и алгоритмы, в системе «1С:Предприятие», помимо редактора диалоговых форм и табличных документов, предусмотрена возможность настройки командного интерфейса системы.

При этом командный интерфейс автоматически учитывает права доступа пользователя, который вошел в систему. При этом пользователю будут показаны только те объекты системы, доступ к которым пользователю разрешен.

Подсистемы. Конфигуратор позволяет на этапе проектирования в рамках одной конфигурации выделить различные подсистемы (например, торговый учет и исследовательский комплекс). Для каждой подсистемы можно указать объекты конфигурации, которые в нее входят. Допускается указывать принадлежность одного объекта к нескольким подсистемам. Фактически подсистемы определяют основные разделы конфигурации, с которыми будет работать пользователь. В связи с тем, что структура подсистем определяет интерфейс конфигурации, следует уделять вопросам проектирования подсистем (и их иерархии) особое внимание.

Система настройки прав доступа (роли). Данная система позволяет описывать наборы прав, соответствующие должностям пользователей или виду деятельности. Структура прав определяется конкретной конфигурацией системы. Например, могут быть введены такие наборы прав, как [Главный бухгалтер](#), [Кладовщик](#), [Менеджер](#), [Начальник отдела](#).

Кроме того, для объектов, хранящихся в базе данных (справочник, документы, регистры и т. д.), могут быть определены права доступа к отдельным полям и записям.

Сам список пользователей создается уже для конкретной организации. Каждому пользователю назначается одна или несколько ролей, основной интерфейс и язык, используемые при работе с программой.

Отладчик. Для удобства разработки конфигурации в системе предусмотрен отладчик. Отладчик позволяет проследить исполнение программных модулей конфигурации, замерять сравнительное время исполнения, просматривать содержимое переменных.

Хранилище конфигураций. Для групповой разработки конфигурации разработчики используют механизм хранилища конфигураций. Он позволяет распределять права доступа по модификации объекта конфигурации и производить необходимые изменения одновременно, а не последовательно.

Поддержка конфигураций. Для удобства проведения обновлений конфигураций предусмотрен механизм формирования разработчиками типовых конфигураций файлов поставки и комплектов поставки (включают программу установки), а также механизм обновления типовых конфигураций, находящихся на поддержке.

Глава 10. Система компоновки данных

Система компоновки данных предназначена для создания отчетов «1С:Предприятия» на основе их декларативного описания. Использование декларативного описания отчетов позволяет реализовать следующие возможности:

- создание отчета без программирования;
- возможность создания различных вариантов отчета;
- возможность задания различных вариантов пользовательских настроек;
- использование автоматически генерируемых форм просмотра и настройки отчета;
- разбиение исполнения отчета на этапы;
- исполнение отдельных этапов построения отчета на различных компьютерах;
- независимое использование отдельных частей системы компоновки данных;
- программное влияние на процесс выполнения отчета;
- настройки структуры отчета;
- совмещение в отчете нескольких таблиц;
- создание вложенных отчетов и др.

Возможности системы компоновки данных используются:

- при работе динамических списков (см. [здесь](#));
- при подготовке данных для последующей обработки (например, в обработках).

10.1. Общие сведения о компоновке данных

Система компоновки данных представляет собой совокупность элементов, каждый из которых соответствует определенному этапу выполнения отчета. Таким образом, весь процесс выполнения отчета в системе компоновки данных сводится к последовательному переходу от одного элемента к другому, доходя в итоге до готового отчета.

Каждый элемент системы компоновки данных имеет собственное декларативное описание, возможность программного доступа и возможность сериализации в/из XML. Такой подход позволяет гибко управлять различными этапами выполнения отчета.

Основные элементы системы компоновки данных представлены на [рис. 249](#).

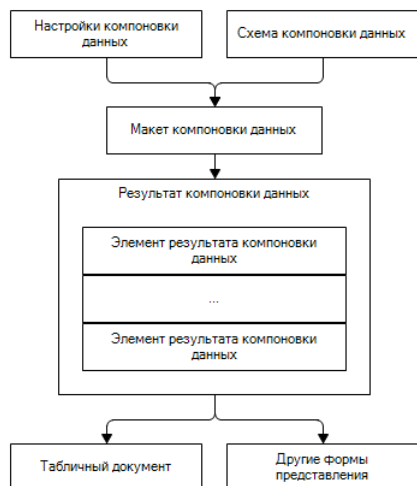


Рис. 249. Основные элементы системы компоновки данных

Схема компоновки данных – описывает суть данных, которые предоставляются отчету (откуда получать данные и как можно управлять компоновкой данных). Представляет собой базу, на основе которой могут быть сформированы всевозможные отчеты. Может содержать:

- текст запроса с инструкциями системы компоновки данных;
- описание нескольких наборов данных;
- описание доступных полей;
- описание связей между несколькими наборами данных;
- описание параметров получения данных;
- описание макетов полей и группировок и др.

Настройки компоновки данных – описывают все, что может настроить разработчик или пользователь в некоторой установленной схеме компоновки данных. Могут содержать:

- отбор;
- упорядочивание;
- условное оформление;
- структуру отчета (составные части будущего отчета);
- параметры получения данных;
- параметры вывода данных и др.

Макет компоновки данных – представляет собой уже готовое описание того, как должен быть сформирован отчет. В нем соединяется схема компоновки и настройки компоновки. Фактически представляет собой результат применения конкретных настроек к схеме компоновки и является готовым заданием процессору компоновки на формирование отчета нужной структуры с учетом конкретных настроек.

Элемент результата компоновки данных – результат компоновки данных представляется набором элементов результата компоновки данных. Как самостоятельная логическая сущность результат компоновки данных не существует, существуют только его элементы. Элементы результата компоновки данных можно вывести в табличный документ для представления их конечному пользователю или в другие виды документов.

Процесс компоновки данных состоит из нескольких этапов, которые представлены на [рис. 250](#).

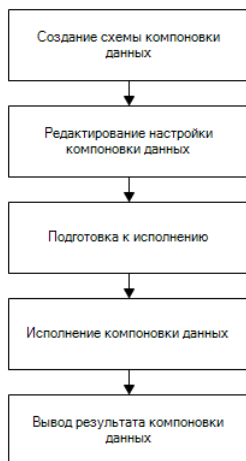


Рис. 250. Этапы процесса компоновки данных

Создание схемы компоновки данных может быть выполнено:

- визуально, при помощи конструктора схемы компоновки данных;
- визуально, при помощи любого редактора, позволяющего редактировать текст XML;
- программно, при помощи объектов встроенного языка системы «1С:Предприятие».

Редактирование настроек компоновки данных – для редактирования настроек компоновки в системе предусмотрен ряд объектов встроенного языка и расширений табличного поля.

Подготовка к исполнению – процесс формирования макета компоновки данных. В данном процессе формируются запросы, необходимые для получения данных, указанных в настройках, формируются макеты областей отчета.

Исполнение компоновки данных – это процесс получения, агрегации, оформления данных.

Вывод результата компоновки данных – полученный результат компоновки может быть выведен в документ, который будет показан пользователю. Отчет может быть выведен в различных форматах.

Схема на [рис. 251](#) в обобщенном виде представляет объекты системы компоновки данных, используемые на различных этапах создания отчета.

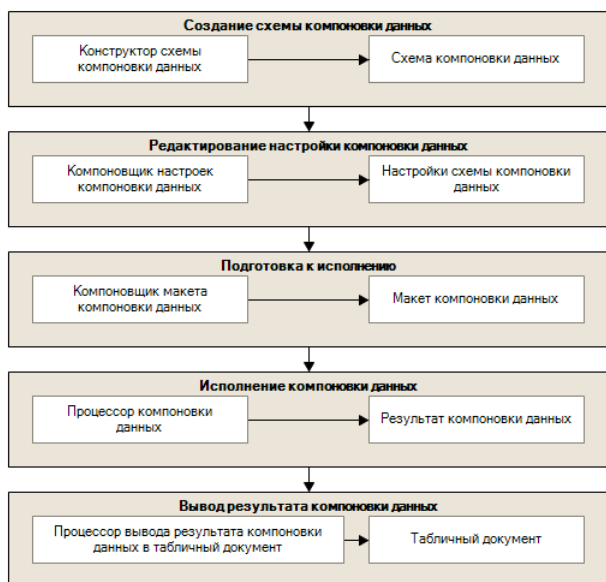


Рис. 251. Объекты системы компоновки данных

Конструктор схемы компоновки данных – может быть использован для создания схемы компоновки данных.

Компоновщик настроек компоновки данных – может быть использован для редактирования настроек системы компоновки данных.

Компоновщик макета компоновки данных – используется для подготовки к исполнению.

Процессор компоновки данных – осуществляет исполнение компоновки данных.

Процессор вывода результата компоновки данных в табличный документ – выводит элементы результата компоновки данных в табличный документ.

Отличительной особенностью отчета, получаемого с помощью системы компоновки данных, является то, что он может иметь сложную структуру, включающую в себя различное сочетание следующих элементов:

- группировка,
- таблица,
- диаграмма,
- вложенный отчет.

Таким образом, отчет, полученный с помощью системы компоновки данных, представляет собой не таблицу, а сложную иерархическую структуру, в которой участвуют перечисленные элементы.

Отчет по номенклатуре

5 товаров с наибольшим остатком

№ в группе	Номенклатура	Количество остаток
1	Клавиатура LK-601 KB-2000 PS/2	137,00
2	Мышь 2-кноп A4Tech PS/2	83,00
3	Клавиатура Apple Pro Keyboards	74,00
4	Мышь Ice Mouse MUS-2	73,00
5	Мышь GENIUS "EASY" (3 кнопки),	71,00

Остатки на складах

Склад	Количество остаток
Витрина в офисе	33,00
Основной склад	518,00
Склад отдела продаж	274,00
Итого	825,00

Процент остатков товаров на складах

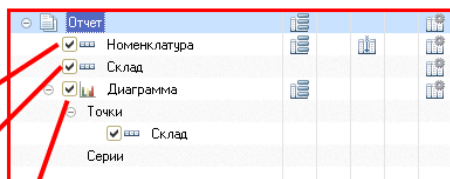


Рис. 252. Структура отчета

На данном рисунке приведен пример отчета, содержащего на первом уровне иерархии диаграмму и группировку по номенклатурным группам. А внутри группировки по номенклатурным группам расположена таблица, содержащая обороты номенклатуры, принадлежащей указанной номенклатурной группе.

10.2. Общие объекты системы компоновки данных

10.2.1. Свойство «Использование»

Многие объекты, в основном подсистемы настроек компоновки данных, имеют свойство булевого типа *Использование*. Это свойство позволяет отключать часть функциональности без ее физического удаления. Значение этого свойства по умолчанию – *Истина*, кроме отдельно описанных случаев.

10.2.2. Поле системы компоновки данных

Это объект, представляющий путь к данным поля. Поле реализовано в виде отдельного типа с целью устранить неоднозначность в свойствах некоторых объектов, которые могут принимать значения как строк, так и полей. Имеет конструктор с параметром *Строка*, описывающий путь к данным; свойств и методов не имеет. Если путь к данным содержит идентификатор с пробелами или специальными символами, то такие идентификаторы следует заключать в квадратные скобки.

10.2.3. Параметры системы компоновки данных

Механизм параметров был реализован для единообразного использования и редактирования коллекций некоторых значений, состав и тип элементов которых определен заранее и не может быть изменен. Механизм параметров состоит из двух частей:

- Доступные параметры – определяют состав коллекции и допустимые типы ее элементов. Параметр является аналогом поля системы компоновки данных.
- Значения параметров.

10.3. Схема компоновки данных

Схема компоновки данных представляется объектом встроенного языка системы «1С:Предприятие» *СхемаКомпоновкиДанных* и состоит из множества других вложенных объектов. Схема компоновки данных имеет представление в виде XML; таким образом, может быть создана любыми средствами, позволяющими генерировать XML, равно как и использована любыми средствами, которые могут читать XML.

Схема компоновки данных используется для предоставления информации о доступных настройках, а также при формировании макета компоновки данных (см. [здесь](#)), т. е. при исполнении компоновки данных.

Для визуального редактирования схемы компоновки данных предназначен конструктор схемы компоновки данных (см. [здесь](#)).

Загрузку схемы компоновки данных из XML можно осуществить стандартными средствами встроенного языка.

```
ЧтениеXML = Новый ЧтениеXML;
ЧтениеXML.УстановитьСтроку(ЭлементыФормы.ТекстСхемыКомпоновкиДанных.ПолучитьТекст());
СКД = СериализаторXDTO.ПрочитатьXML(ЧтениеXML, Тип("СхемаКомпоновкиДанных"));
```

Копировать в буфер обмена

Все выражения, описываемые в схеме компоновки данных, записываются на языке выражений системы компоновки данных (см. [здесь](#)).

10.3.1. Составные части схемы компоновки данных

Каждая схема компоновки данных содержит множество объектов, описывающих ту или иную часть. Рассмотрим эти составные части.

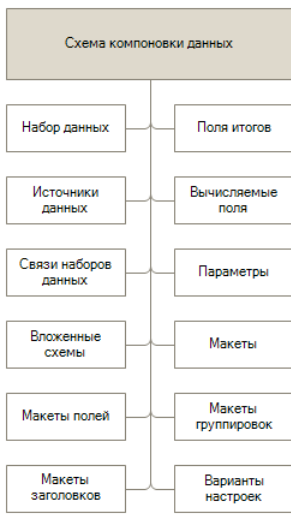


Рис. 253. Составные части схемы компоновки данных

10.3.1.1. Источники данных

Схема компоновки данных может содержать несколько источников данных.

Под источником данных подразумевается источник, из которого будут получаться данные. В качестве источника данных выступает информационная база системы «1С:Предприятие».

Источники данных описываются в свойстве `ИсточникиДанных` схемы, которое содержит коллекцию значений, состоящую из элементов `ИсточникДанныхСхемыКомпоновкиДанных`.

10.3.1.2. Наборы данных

Наборы данных в схеме компоновки данных содержат информацию о том, какие поля можно получать из данного набора, какие поля набора данных можно использовать в отборе и т. п.

В схеме компоновки данных допускается наличие нескольких наборов данных (см. [здесь](#)).

Наборы данных описываются в свойстве `НаборыДанных` схемы. Свойство содержит коллекцию значений, в которую могут входить следующие элементы:

- **запрос** (`НаборДанныхЗапросСхемыКомпоновкиДанных`) – получение данных описывается при помощи языка запросов;
- **объект** (`НаборДанныхОбъектСхемыКомпоновкиДанных`) – описывается имя внешнего набора данных, из которого будут получаться данные;
- **объединение** (`НаборДанныхОбъединениеСхемыКомпоновкиДанных`) – описываются наборы данных – составные части объединения.

Все три набора данных содержат ряд общих свойств:

- **Имя** – имя набора данных, под которым к этому набору данных можно будет обращаться из других объектов схемы компоновки данных. В рамках одной схемы компоновки данных имена наборов данных должны быть уникальными.
- **Поля** – описания полей, доступных для набора данных.

Кроме этого, наборы данных – запрос и объект содержат свойство `ИсточникДанных` – имя источника данных, из которого будут получаться данные. Оно должно содержать имя одного источника данных, присутствующего в схеме компоновки данных (см. [здесь](#)).

Набор данных – запрос содержит свойство `Запрос` – текст запроса, при помощи которого будут получаться данные из источника данных. В тексте запроса допускается использование специального расширения – расширения языка запросов для системы компоновки данных (см. [здесь](#)).

Набор данных – объединение содержит свойство `Элементы`, содержащее перечень наборов данных, входящих в объединение (см. [здесь](#)).

10.3.1.2.1. Набор данных – запрос

Содержит обычный запрос к данным, расположенным в информационной базе «1С:Предприятия».

Набор данных – запрос может содержать пакетный запрос. Результирующим запросом будет пакетный запрос. При этом состав полей, которые будут помещаться во временную таблицу, автоматически определяется по использованным в других запросах полям. В случае, если от временной таблицы не понадобилось ни одно поле, временная таблица не будет помещаться в результирующий запрос. Отбор, применяемый в настройках компоновки данных, применяется во всех запросах пакета.

Содержимым набора данных будет считаться результат последнего запроса пакета.

В рамках одного набора данных возможно обращение только к одному источнику данных (включая внешние источники данных).

10.3.1.2.2. Набор данных – объект

Набор данных – объект используется для вывода в отчет информации из некоторого объекта встроенного языка: таблицы значений, результата запроса, текущего документа и т. п.

Этот «источник» данных описывается в схеме компоновки, затем он заполняется программным образом, например, по нажатию какой-то кнопки, и в качестве внешнего набора данных передается в процессор компоновки.

10.3.1.2.3. Набор данных – объединение

Набор данных – объединение содержит свойство `Элементы`, описывающее наборы данных, которые необходимо объединить.

Заметим, что значения полей набора данных – объединение будут получаться из полей вложенных наборов данных по их пути к данным. Так, в приведенном примере у внешнего набора данных будет поле `СуммаРасх`. Данные для него будут получаться из вложенных наборов данных, у которых путь к данным – `СуммаРасход`.

10.3.1.2.4. Поле набора данных схемы компоновки данных

Набор данных может содержать описания полей, которые будут доступны для этого набора данных.

Поля набора данных описываются в свойстве `Поля наборов данных`, которое содержит коллекцию значений, состоящую из элементов `ПолеНабораДанныхСхемыКомпоновкиДанных`.

10.3.1.3. Связи наборов данных

Наборы данных, присутствующие в схеме компоновки данных, могут быть связаны друг с другом.

Связи наборов данных описываются в свойстве [СвязиНаборовДанных](#) схемы компоновки данных, которое содержит коллекцию значений, состоящую из элементов [СвязьНаборовДанныхСхемыКомпоновкиДанных](#).

10.3.1.4. Вычисляемые поля

В схеме компоновки данных существует возможность описать поля, которые будут вычисляться по некоторым выражениям с использованием полей наборов данных. Данные поля могут быть использованы в настройках точно так же, как поля набора данных.

Вычисляемые поля описываются в свойстве [ВычисляемыеПоляСхемыКомпоновкиДанных](#) схемы компоновки данных, которое содержит коллекцию значений, состоящую из элементов [ВычисляемоеПолеСхемыКомпоновкиДанных](#).

10.3.1.5. Поля ресурсов

В схеме компоновки данных возможно описание полей ресурсов, значения которых будут вычисляться для групповых записей. Осуществляется это при помощи описания поля итога.

Поля итога описываются в свойстве [ПоляИтога](#) схемы компоновки данных, которое содержит коллекцию значений, состоящую из элементов [ПолеИтогаСхемыКомпоновкиДанных](#). При расчете ресурсов каждая запись набора данных, для которого рассчитывается ресурс, участвует в расчете ресурса только один раз.

10.3.1.6. Параметры

Схема компоновки данных содержит описание параметров данных.

Параметры данных описываются в свойстве [Параметры схемы компоновки данных](#), которое содержит коллекцию значений, состоящую из элементов [ПараметрСхемыКомпоновкиДанных](#).

Для параметра можно указать признак обязательности использования параметров и запрет незаполненного параметра. Например, с помощью комбинации этих признаков можно реализовать параметр, без заполнения которого формирование отчета невозможно.

10.3.1.7. Вложенные схемы

Схема компоновки данных может содержать описания вложенных схем компоновки данных.

Вложенные схемы компоновки данных описываются в свойстве [ВложенныеСхемыКомпоновкиДанных](#) схемы компоновки данных, которое содержит коллекцию значений, состоящую из элементов [ВложеннаяСхемаКомпоновкиДанных](#).

10.3.1.8. Макеты

В схеме компоновки данных можно описать макеты, которые будут использоваться для вывода поля или группировки. При указании макета для поля или группировки указывается имя макета, описанного в данном свойстве.

Макеты описываются в свойстве [Макеты схемы компоновки данных](#). Свойство содержит коллекцию значений, состоящую из элементов [ОписаниеМакетаСхемыКомпоновкиДанных](#).

10.3.1.9. Макеты полей

Для каждого поля в схеме компоновки данных может быть указано имя макета, используемого для вывода данного поля в результат компоновки.

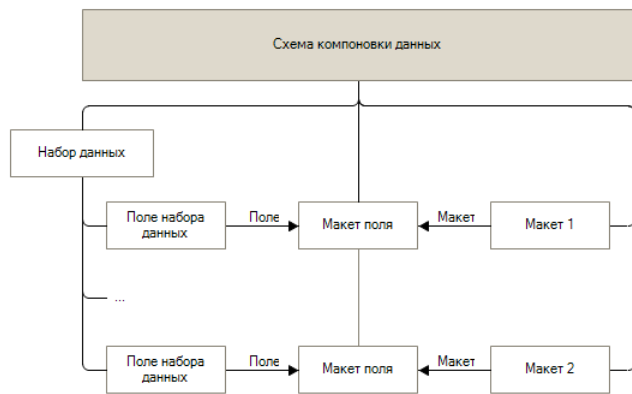


Рис. 254. Макеты полей

Связь поля с макетом описывается с помощью объекта [МакетПоляСхемыКомпоновкиДанных](#). Коллекция этих объектов содержится в свойстве [МакетыПолей](#) объекта [СхемаКомпоновкиДанных](#).

10.3.1.10. Макеты группировок

Для каждой группировки в схеме компоновки данных можно указать имя макета, который будет использоваться при выводе данной группировки.

Связь группировки с макетом описывается с помощью объекта [МакетГруппировкиСхемыКомпоновкиДанных](#). Коллекция этих объектов содержится в свойстве [МакетыГруппировок](#) объекта [СхемаКомпоновкиДанных](#).

10.3.1.11. Макеты заголовков группировок

Для каждой группировки могут описываться также макеты заголовков группировок.

Связь заголовка группировки с макетом описывается с помощью объекта [МакетГруппировкиСхемыКомпоновкиДанных](#) (см. предыдущий раздел). Коллекция этих объектов содержится в свойстве [МакетыЗаголовковГруппировок](#) объекта [СхемаКомпоновкиДанных](#).

10.3.1.12. Настройки по умолчанию

Каждый вариант отчета, который задан в схеме компоновки данных, содержит настройки компоновки данных по умолчанию, которые могут быть заданы разработчиком. При этом вариант по умолчанию будет считаться тот вариант, который стоит первым в списке вариантов настроек системы компоновки данных (вариант [Диаграмма по периодам](#) на [рис. 255](#)). Настройки варианта по умолчанию будут применяться при первом открытии отчета, при выборе команды [Еще – Стандартные настройки](#) ([Все действия – Стандартные настройки](#)), а также могут быть использованы для программной установки настроек по умолчанию.

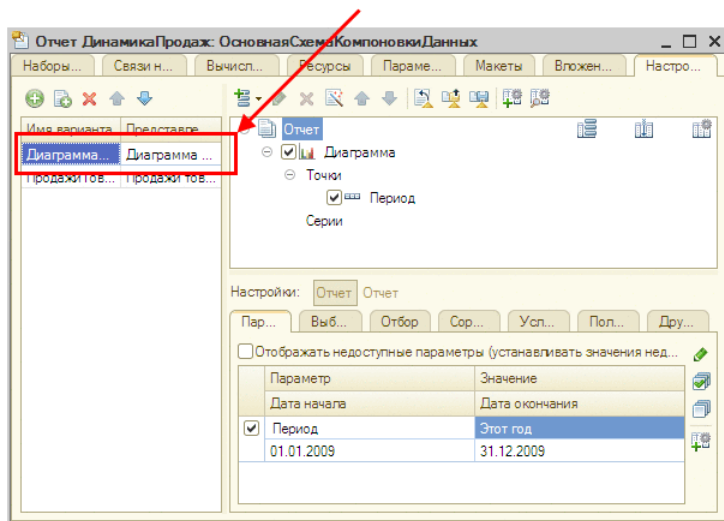


Рис. 255. Вариант по умолчанию

Подробнее о настройках компоновки данных см. [здесь](#).

10.3.2. Работа с несколькими наборами данных

Система компоновки данных позволяет использовать в одной компоновке несколько наборов данных.

Для того чтобы в одной компоновке использовать несколько наборов данных, необходимо внести в схему описания наборов данных, которые предполагается использовать, и указать связи между наборами данных.

Рассмотрим следующий пример.

Запишем три набора данных: **ПрайсЛист**, **Остатки** и **Продажи**.



Рис. 256. Пример нескольких наборов данных

Опишем связи между наборами данных. Создадим связь между наборами данных **ПрайсЛист** и **Остатки**, а также связь между наборами данных **ПрайсЛист** и **Продажи** (см. [рис. 257](#)).

Источник связи	Приемник свя...	Выражение и...	Выражение п...	Параметр	Список парам...	Усп
ПрайсЛист	Продажи	Ссылка	Товар	Номенклатура	<input checked="" type="checkbox"/>	
ПрайсЛист	Остатки	Ссылка	Товар	Номенклатура	<input checked="" type="checkbox"/>	

Рис. 257. Связи наборов данных

Дополнительно опишем ресурсы (см. [рис. 258](#)).

Доступные поля	Поле	Выражение	Рассчитыв...
Валюта	ВалютнаяСуммаКонечныйОстатокДт	Сумма(ВалютнаяСуммаКонечныйОста...	Валюта
ВалютнаяСуммаКонечныйОстатокДт	ВалютнаяСуммаКонечныйОстатокКт	Сумма(ВалютнаяСуммаКонечныйОста...	Валюта
ВалютнаяСуммаКонечныйОстатокКт	ВалютнаяСуммаНачальныйОстатокДт	Сумма(ВалютнаяСуммаНачальныйОс...	Валюта
ВалютнаяСуммаНачальныйОстатокДт	ВалютнаяСуммаНачальныйОстатокКт	Сумма(ВалютнаяСуммаНачальныйОс...	Валюта
ВалютнаяСуммаНачальныйОстатокКт	ВалютнаяСуммаОборотДт	Сумма(ВалютнаяСуммаОборотДт)	Валюта
ВалютнаяСуммаОборотДт	ВалютнаяСуммаОборотКт	Сумма(ВалютнаяСуммаОборотКт)	Валюта
ВалютнаяСуммаОборотКт	СуммаКонечныйОстатокДт	Сумма(СуммаКонечныйОстатокДт)	
СуммаКонечныйОстатокДт	СуммаКонечныйОстатокКт	Сумма(СуммаКонечныйОстатокКт)	
СуммаКонечныйОстатокКт	СуммаНачальныйОстатокДт	Сумма(СуммаНачальныйОстатокДт)	
СуммаНачальныйОстатокДт	СуммаНачальныйОстатокКт	Сумма(СуммаНачальныйОстатокКт)	
СуммаНачальныйОстатокКт	СуммаОборотДт	Сумма(СуммаОборотДт)	
СуммаОборотДт	СуммаОборотКт	Сумма(СуммаОборотКт)	
СуммаОборотКт			
Счет			

Рис. 258. Ресурсы схемы компоновки данных

Если в системе компоновки данных описывается связь между двумя наборами данных, то набор данных, к которому идет связь, будет считаться **зависимым**. Набор данных, от которого идет связь, будет считаться **родительским** по отношению к зависимому от него набору данных.

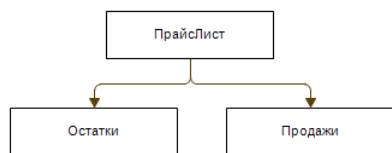


Рис. 259. Зависимые наборы данных

В приведенных примерах зависимыми наборами данных будут наборы данных *Остатки* и *Продажи*. А родительским по отношению к обоим этим наборам – *ПрайсЛист*.

В схеме компоновки данных нет указания типа связи. Все связи считаются левыми внешними соединениями. То есть запись родительского набора данных будет использоваться в компоновке даже в том случае, если для нее не найдены записи в зависимом наборе.

В макете компоновки данных (см. [здесь](#)) возможно указание типа связи. Тип связи генерируется компоновщиком макета в зависимости от накладываемых глобальных отборов. Если на поле набора данных, являющегося зависимым, накладывается глобальный отбор, генерируемые связи в макете компоновки данных от этого набора данных до всех его родительских наборов (вплоть до начала иерархии наборов данных) будут иметь тип *Внутренняя*. Это означает, что записи родительского набора данных будут участвовать в компоновке только в том случае, если будут найдены записи в зависимых наборах данных.

Например, если пользователь наложит глобальный отбор на поле *Склад*, наборы данных *ПрайсЛист* и *Остатки* будут связаны внутренней связью.

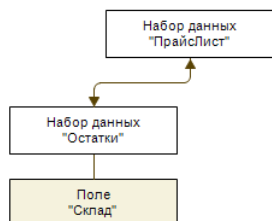


Рис. 260. Внутренняя связь

Если набор данных зависит от некоторого набора и в связи указана возможность использования списка параметров, данные из зависимого набора данных будут получаться порциями по 1000 записей. В случае, если использование списка параметров в связи не разрешено, записи будут получаться по одной.

Если и зависимый, и родительский набор данных содержит поле с одинаковым именем, данное поле будет получаться из родительского набора данных. В приведенных примерах поле *Номенклатура* всегда будет получаться из набора данных *ПрайсЛист*.

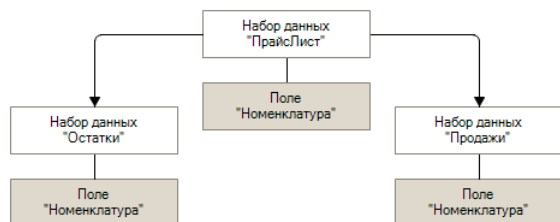


Рис. 261. Общее поле

Несвязанные наборы данных не могут содержать поля с одинаковыми именами, если у них нет общего родителя, в котором данное поле также присутствует. Так, наборы данных *Остатки* и *Обороты* могут содержать поле *Номенклатура*, но не могли бы оба содержать поле *Контрагент*.

В описании связи, в выражении источника, используются поля из набора данных источника (родительского набора); в выражении связи приемника используются поля набора данных приемника (зависимого набора данных). Так, при описании связи между наборами данных *ПрайсЛист* и *Остатки* выражение *Номенклатура* в свойстве *ВыражениеИсточник* использует поле набора данных *ПрайсЛист*, а выражение *Номенклатура* в свойстве *ВыражениеПриемник* использует поле набора данных *Остатки*.

В одной группировке не могут быть использованы поля из не связанных друг с другом наборов данных. При этом наборы данных, имеющие общие родительские наборы данных, связанными не считаются. Исключение сделано для полей-итогов, которые могут быть использованы в любой группировке. В приведенном примере мы не сможем в одной группировке задействовать поля *Склад* и *Контрагент*. Однако поля *КоличествоОстаток* и *СуммаОборот* сможем задействовать, т. к. они являются ресурсами.

Данные зависимого набора данных не могут быть получены без получения данных родительского набора. То есть при получении данных из зависимого набора автоматически будут получаться и данные из родительского набора (и всех родителей родителя). В нашем примере при получении набора данных *Остатки* будут получаться данные и из набора *ПрайсЛист*.

Если в группировке используются наборы данных из нескольких наборов данных, при исполнении компоновки будет осуществляться обход по последнему зависимому набору данных. Так, если в группировке будут использоваться поля наборов *ПрайсЛист* и *Остатки*, обход будет происходить по набору *Остатки*.

Если ни одного поля из связанного набора данных в настройках не задействовано, набор данных не будет включен в макет компоновки данных.

10.3.3. Расширение языка запросов для системы компоновки данных

Расширение языка запросов для системы компоновки данных осуществляется при помощи специальных синтаксических инструкций, заключаемых в фигурные скобки и помещаемых непосредственно в текст запроса.

10.3.3.1. Синтаксические элементы расширения языка запросов системы компоновки данных

ВЫБРАТЬ

Описание:

В этом предложении описываются поля, которые пользователь сможет выбирать для вывода. После данного ключевого слова через запятую перечисляются псевдонимы полей из основного списка выборки запроса, которые будут доступными для настройки.

После псевдонима поля может находиться комбинация символов «. *», что обозначает возможность использования дочерних полей от данного поля.

Например, запись *Номенклатура.** обозначает возможность использования дочерних полей поля *Номенклатура* (например, поля *Номенклатура.Код*). Элемент **ВЫБРАТЬ** может присутствовать только в первом запросе объединения.

Пример:

```
{ВЫБРАТЬ Номенклатура, Склад}
```

Копировать в буфер обмена

ГДЕ

Описание:

Описываются поля, на которые пользователь сможет накладывать отбор. В данном предложении используются поля таблиц. Использование псевдонимов полей списка выборки недопустимо. Каждая часть объединения может содержать собственный элемент **ГДЕ**.

Если значения параметров не заданы, то предложение **ГДЕ** в результирующий запрос не включается.

Пример:

```
{ГДЕ Номенклатура.*; Склад }
{ГДЕ Документ.Дата >= &амр;ДатаНачала, Документ.Дата <= &амр;ДатаКонца}
```

[Копировать в буфер обмена](#)

ХАРАКТЕРИСТИКИ

Описание:

Для обеспечения работы с характеристиками в расширение языка запросов для системы компоновки данных введен синтаксис описания характеристик.

В примере (см. далее) описываются характеристики для полей типа **Ссылка** на справочник **Номенклатура**.

В характеристиках описываются следующие свойства:

- **ТИП** – имя типа, для которого описываются характеристики.
- **ВИДЫХАРАКТЕРИСТИК** – имя таблицы или запрос для получения списка характеристик. В запросе допустимо использовать ключевое слово **РАЗРЕШЕННЫЕ**. В этом случае в списке характеристик будут присутствовать только те характеристики, которые допускаются для пользователя наложенными ограничениями доступа к данным.

Если список характеристик указан не запросом, а именем таблицы, то в запросе получения списка характеристик, который формирует система, также используется ключевое слово **РАЗРЕШЕННЫЕ**.

- **ПОЛЕКЛЮЧА** – имя поля, содержащего идентификатор характеристики.
- **ПОЛЕИМЕНИ** – имя поля, содержащего имя характеристики.
- **ПОЛЕТИПАЗНАЧЕНИЯ** – имя поля, содержащего тип значения характеристики. Если тип значения не указан, считается, что характеристика имеет тип **Булево**.
- **ЗНАЧЕНИЯХАРАКТЕРИСТИК** – имя таблицы или запрос для получения значений характеристик.
- **ПОЛЕОБЪЕКТА** – имя поля, содержащего идентификатор объекта (например, ссылка номенклатуры).
- **ПОЛЕВИДА** – имя поля, содержащего идентификатор характеристики.
- **ПОЛЕЗНАЧЕНИЯ** – имя поля, содержащего значение характеристики. Если не указано, значение будет равно **Истина** (если такая характеристика у объекта есть), **Ложь** – в противном случае.

Пример:

```
{ХАРАКТЕРИСТИКИ ТИП(Справочник.Номенклатура)
ВИДЫХАРАКТЕРИСТИК (ВЫБРАТЬ
    ВидыДопСвойств.Ссылка,
    ВидыДопСвойств.Наименование,
    ВидыДопСвойств.ТипЗначения
ИЗ
    ПланВидовХарактеристик.ВидыДопСвойств КАК ВидыДопСвойств )
ПОЛЕКЛЮЧА Ссылка
ПОЛЕИМЕНИ Наименование
ПОЛЕТИПАЗНАЧЕНИЯ ТипЗначения
ЗНАЧЕНИЯХАРАКТЕРИСТИК РегистрСведений.ДопСвойства
ПОЛЕОБЪЕКТА Номенклатура
ПОЛЕВИДА ВидСвойства
ПОЛЕЗНАЧЕНИЯ Свойство
}
```

[Копировать в буфер обмена](#)

Параметры

Описание:

Кроме основных элементов система компоновки данных принимает элементы, записанные в параметрах виртуальных таблиц. В таких случаях тип полей зависит от типа параметра, в котором располагаются элементы.

Поля **ДатаНачала**, **ДатаКонца**, **Номенклатура** и **Склад** (см. пример) станут доступными в отборе, т. е. пользователь сможет применять для них фильтры.

Пример:

```
ВЫБРАТЬ
    УчетНоменклатурыОбороты.Номенклатура КАК Номенклатура,
    УчетНоменклатурыОбороты.Склад КАК Склад,
    УчетНоменклатурыОбороты.КоличествоПриход КАК КоличествоПриход,
    УчетНоменклатурыОбороты.КоличествоРасход КАК КоличествоРасход
ИЗ
    РегистрНакопления.УчетНоменклатуры.Обороты( {&ДатаНачала},
        {&ДатаКонца},
        {Номенклатура.*},
        {Склад.*} ) КАК УчетНоменклатурыОбороты
```

[Копировать в буфер обмена](#)

10.3.3.2. Автоматическое заполнение доступных полей

При автоматическом заполнении доступных полей запроса выполняются следующие действия:

- все поля списка выборки и их дочерние поля становятся доступными для выбора, упорядочивания, группировки, отбора и др.;
- параметры виртуальных таблиц становятся доступными для отбора.

10.3.4. Язык выражений системы компоновки данных

Язык выражений системы компоновки данных предназначен для записи выражений, используемых в различных подсистемах. Выражения используются в следующих подсистемах:

- схема компоновки данных – для описания вычисляемых полей, полей итогов, выражений связи и т. д.;
- настройки компоновки данных – для описания выражений пользовательских полей;

- макет компоновки данных – для описания выражений связи наборов данных, описания параметров макета и т. д.

ПРИМЕЧАНИЕ. В языке выражений компоновки данных отсутствует возможность получения поля через точку от выражения.

ВНИМАНИЕ! Вычисление выражения системы компоновки данных происходит в безопасном режиме (см. [здесь](#)).

10.3.4.1. Литералы

В выражении могут присутствовать литералы типов, описанных далее.

Строка

Описание:

Строковый литерал записывается в символах «"».

При необходимости использования внутри строкового литерала символа «"» следует использовать два таких символа.

Пример:

```
"Литерал ""в кавычках"""
```

[Копировать в буфер обмена](#)

Число

Описание:

Число записывается без пробелов, в десятичном формате. Дробная часть отделяется при помощи символа «.».

Пример:

```
10.5
200
```

[Копировать в буфер обмена](#)

Дата

Описание:

Литерал типа *Дата* записывается при помощи ключевого слова *ДАТАВРЕМЯ* (*DATETIME*). После данного ключевого слова в скобках, через запятую, перечисляются год, месяц, день, часы, минуты, секунды. Указание времени необязательно.

Пример:

```
// Шестое января 1975 года
ДАТАВРЕМЯ(1975, 1, 06)
// Второе декабря 2006 года, 23 часа 56 минут 57 секунд
ДАТАВРЕМЯ(2006, 12, 2, 23, 56, 57)
```

[Копировать в буфер обмена](#)

Булево

Описание:

Булевы значения могут быть записаны при помощи литералов *Истина* (*True*), *Ложь* (*False*).

Значение

Описание:

Для указания литералов других типов (системных перечислений, предопределенных данных) используется ключевое слово *ЗНАЧЕНИЕ*, после которого в скобках идет указание имени литерала.

Пример:

```
ЗНАЧЕНИЕ(ВидСчета.Активный)
```

[Копировать в буфер обмена](#)

Поля

Описание:

В выражениях могут использоваться поля наборов данных. Поле идентифицируется путем к данным. Части пути к данным отделяются друг от друга символом «.» . Имя поля не является чувствительным к регистру. Если путь к данным содержит идентификатор с пробелами или специальными символами, то такие идентификаторы следует заключать в квадратные скобки.

Пример:

```
Номенклатура.Артикул
Продажи.СуммаОборот
Продажи.[Сумма оборот]
```

[Копировать в буфер обмена](#)

Параметры

Описание:

Выражения могут использовать параметры. Для использования в выражении параметра достаточно написать его имя, которому будет предшествовать символ «&».

Пример:

```
&Контрагент
&ДатаНачала
```

[Копировать в буфер обмена](#)

Тип

Описание:

Формирует значение типа *Тип*. Тип задается при помощи ключевого слова *Тип*.

Пример:

```
Тип("Строка")
```

[Копировать в буфер обмена](#)

10.3.4.2. Операции над числами**Унарный "-"**

Описание:

Данная операция предназначена для изменения знака числа на обратный.

Пример:

-Продажи.Количество

[Копировать в буфер обмена](#)

Унарный "+"

Описание:

Данная операция не выполняет над числом никаких действий.

Пример:

+Продажи.Количество

[Копировать в буфер обмена](#)

Бинарный "-"

Описание:

Данная операция предназначена для вычисления разности двух чисел.

Пример:

ОстиОбрт.НачальныйОстаток - ОстиОбрт.КонечныйОстаток
ОстиОбрт.НачальныйОстаток - 100
400 - 357

[Копировать в буфер обмена](#)

Бинарный "+"

Описание:

Данная операция предназначена для вычисления суммы двух чисел.

Пример:

ОстиОбрт.НачальныйОстаток + ОстиОбрт.Оборот
ОстиОбрт.НачальныйОстаток + 100
400 + 357

[Копировать в буфер обмена](#)

Произведение "*"

Описание:

Данная операция предназначена для вычисления произведения двух чисел.

Пример:

Номенклатура.Цена * 1.2
2 * 3.14

[Копировать в буфер обмена](#)

Деление "/"

Описание:

Данная операция предназначена для получения результата деления одного операнда на другой.

Пример:

Номенклатура.Цена / 1.2
2 / 3.14

[Копировать в буфер обмена](#)

Остаток от деления "%"

Описание:

Данная операция предназначена для получения остатка от деления одного операнда на другой.

Пример:

Номенклатура.Цена % 1.2

[Копировать в буфер обмена](#)

2 % 3.14

[Копировать в буфер обмена](#)

10.3.4.3. Операции над строками**Конкатенация (Бинарный "+")**

Описание:

Данная операция предназначена для конкатенации двух строк.

Пример:

Номенклатура.Артикул + " : "+ Номенклатура.Наименование

[Копировать в буфер обмена](#)

ПОДОБНО (LIKE)

Описание:

Данная операция проверяет соответствие строки переданному шаблону.

Значением оператора **ПОДОВНО** является *Истина*, если значение выражения удовлетворяет шаблону, и *Ложь* – в противном случае.

Следующие символы в строке шаблона имеют смысл, отличный от просто очередного символа строки:

- «%» – процент: последовательность, содержащая ноль и более произвольных символов.
- «_» – подчеркивание: один произвольный символ.
- «[...]» – один или несколько символов в квадратных скобках: один символ, любой из перечисленных внутри квадратных скобок. В перечислении могут встречаться диапазоны, например a-z, означающие произвольный символ, входящий в диапазон, включая концы диапазона.
- «[^...]» – в квадратных скобках значок отрицания, за которым следует один или несколько символов: любой символ, кроме тех, которые перечислены следом за значком отрицания.

Любой другой символ означает сам себя и не несет никакой дополнительной нагрузки. Если в качестве самого себя необходимо записать один из перечисленных символов, то ему должен предшествовать спецсимвол, указанный после ключевого слова **СПЕЦСИМВОЛ** (**ESCAPE**).

Например, приведенный ниже шаблон означает подстроку, состоящую из последовательности символов:

- буквы **A**,
- буквы **B**,
- буквы **V**,
- одной цифры,
- одной из букв **a**, **b**, **v** или **г**,
- символа подчеркивания,
- буквы **a**,
- буквы **b**,
- буквы **v**.

Причем эта последовательность может располагаться, начиная с произвольной позиции в строке.

Пример:

```
"%ABV[0-9][абвг]_абв%" СПЕЦСИМВОЛ "\"
```

[Копировать в буфер обмена](#)

10.3.4.4. Операции сравнения

Равно (=)

Описание:

Данная операция предназначена для сравнения двух операндов на равенство.

Пример:

```
Продажи.Контрагент = Продажи.НоменклатураОсновнойПоставщик
```

[Копировать в буфер обмена](#)

Не равно (<>)

Описание:

Данная операция предназначена для сравнения двух операндов на неравенство.

Пример:

```
Продажи.Контрагент <> Продажи.НоменклатураОсновнойПоставщик
```

[Копировать в буфер обмена](#)

Меньше (<)

Описание:

Данная операция предназначена для проверки того, что первый операнд меньше второго.

Пример:

```
ПродажиТекущие.Сумма < ПродажиПрошлые.Сумма
```

[Копировать в буфер обмена](#)

Больше (>)

Описание:

Данная операция предназначена для проверки того, что первый операнд больше второго.

Пример:

```
ПродажиТекущие.Сумма > ПродажиПрошлые.Сумма
```

[Копировать в буфер обмена](#)

Меньше или равно (<=)

Описание:

Данная операция предназначена для проверки того, что первый операнд меньше либо равен второму.

Пример:

```
ПродажиТекущие.Сумма <= ПродажиПрошлые.Сумма
```

[Копировать в буфер обмена](#)

Больше или равно (>=)

Описание:

Данная операция предназначена для проверки того, что первый операнд больше либо равен второму.

Пример:

`ПродажиТекущие.Сумма >= ПродажиПрошлые.Сумма`

[Копировать в буфер обмена](#)

Строка начинается с подстроки (*НачинаетсяC/BeginsWith*)

Описание:

Данная операция осуществляет проверку того, что проверяемое значение начинается на строку, указанную в условии. Условие может быть указано непосредственным значением и шаблоном. Следующие символы в строке шаблона имеют смысл, отличный от просто очередного символа строки:

- `</>` – следующий символ надо интерпретировать как обычный символ.
- `<%>` – процент: последовательность, содержащая ноль и более произвольных символов.
- `<_>` – подчеркивание: один произвольный символ.

Строка не начинается с подстроки (*НеНачинаетсяC/NotBeginsWith*)

Описание:

Данная операция осуществляет проверку того, что проверяемое значение не начинается на строку, указанную в условии. Условие может быть указано непосредственным значением и шаблоном. Следующие символы в строке шаблона имеют смысл, отличный от просто очередного символа строки:

- `</>` – следующий символ надо интерпретировать как обычный символ.
- `<%>` – процент: последовательность, содержащая ноль и более произвольных символов.
- `<_>` – подчеркивание: один произвольный символ.

Строка удовлетворяет шаблону (*Подобно/Like*)

Описание:

Данная операция осуществляет проверку того, что проверяемое значение удовлетворяет заданному шаблону. Следующие символы в строке шаблона имеют смысл, отличный от просто очередного символа строки:

- `</>` – следующий символ надо интерпретировать как обычный символ.
- `<%>` – процент: последовательность, содержащая ноль и более произвольных символов.
- `<_>` – подчеркивание: один произвольный символ.

Строка не удовлетворяет шаблону (*НеПодобно/NotLike*)

Описание:

Данная операция осуществляет проверку того, что проверяемое значение не удовлетворяет заданному шаблону. Следующие символы в строке шаблона имеют смысл, отличный от просто очередного символа строки:

- `</>` – следующий символ надо интерпретировать как обычный символ.
- `<%>` – процент: последовательность, содержащая ноль и более произвольных символов.
- `<_>` – подчеркивание: один произвольный символ.

Операция проверки вхождения в список (*B/IN*)

Описание:

Данная операция осуществляет проверку наличия значения в переданном списке значений. Результатом операции будет *Истина*, если значение найдено, или *Ложь* – в противном случае.

Пример:

`Номенклатура В (&Товар1, &Товар2)`

[Копировать в буфер обмена](#)

Операция проверки наличия значения в наборе данных (*B/IN*)

Описание:

Операция осуществляет проверку наличия значения в указанном наборе данных. Набор данных для проверки должен содержать одно поле.

Пример:

`Продажи.Контрагент В Контрагенты`

[Копировать в буфер обмена](#)

Операция проверки значения на NULL (*ЕСТЬ NULL/IS NULL*)

Описание:

Данная операция возвращает значение *Истина*, если оно является значением *NULL*.

Пример:

`Продажи.Контрагент ЕСТЬ NULL`

[Копировать в буфер обмена](#)

Операция проверки значения на неравенство NULL (*ЕСТЬ НЕ NULL/IS NOT NULL*)

Описание:

Данная операция возвращает значение *Истина*, если оно не является значением *NULL*.

Пример:

`Продажи.Контрагент ЕСТЬ НЕ NULL`

[Копировать в буфер обмена](#)

10.3.4.5. Логические операции

Логические операции принимают в качестве операндов выражения, имеющие тип `Булево`.

Операция НЕ (NOT)

Описание:

Операция **НЕ** возвращает значение `Истина`, если ее операнд имеет значение `Ложь`, и значение `Ложь`, если ее операнд имеет значение `Истина`.

Пример:

`НЕ Документ.Грузополучатель = Документ.Грузоотправитель` [Копировать в буфер обмена](#)

Операция И (AND)

Описание:

Операция **И** возвращает значение `Истина`, если оба операнда имеют значение `Истина`, и значение `Ложь`, если один из операндов имеет значение `Ложь`, например:

Пример:

`Документ.Грузополучатель = Документ.Грузоотправитель И Документ.Грузополучатель = &Контрагент` [Копировать в буфер обмена](#)

Операция ИЛИ (OR)

Описание:

Операция **ИЛИ** возвращает значение `Истина`, если один из операндов имеет значение `Истина`, и `Ложь`, если оба операнда имеют значение `Ложь`.

Пример:

`Документ.Грузополучатель = Документ.Грузоотправитель ИЛИ Документ.Грузополучатель = &Контрагент` [Копировать в буфер обмена](#)

10.3.4.6. Агрегатные функции

Агрегатные функции осуществляют некоторое действие над набором данных.

СУММА (SUM)

Синтаксис:

`Сумма (Выражение)`

Описание:

Агрегатная функция `Сумма` рассчитывает сумму значений выражений, переданных ей в качестве аргумента для всех детальных записей. В качестве параметра можно передавать `Массив`. В этом случае функция будет применена к содержимому массива.

Пример:

`СУММА (Продажи. СуммаОборот)` [Копировать в буфер обмена](#)

КОЛИЧЕСТВО (COUNT)

Синтаксис:

`Количество (Выражение)`

Описание:

Функция `Количество` рассчитывает количество значений, отличных от значения `NULL`. В качестве параметра можно передавать `Массив`. В этом случае функция будет применена к содержимому массива.

Пример:

`КОЛИЧЕСТВО (Продажи. Контрагент)` [Копировать в буфер обмена](#)

КОЛИЧЕСТВО (РАЗЛИЧНЫЕ) (COUNT (DISTINCT))

Синтаксис:

`Количество (Различные Выражение)`

Описание:

Эта функция рассчитывает количество различных значений. Для получения различных значений следует перед параметром метода `Количество` указать `Различные (Distinct)`. В качестве параметра можно передавать `Массив`. В этом случае функция будет применена к содержимому массива.

Пример:

`КОЛИЧЕСТВО (Различные Продажи. Контрагент)` [Копировать в буфер обмена](#)

МАКСИМУМ (MAX)

Синтаксис:

`Максимум (Выражение)`

Описание:

Функция получает максимальное значение. В качестве параметра можно передавать `Массив`. В этом случае функция будет применена к содержимому массива.

Пример:

`МАКСИМУМ (Остатки. Количество)` [Копировать в буфер обмена](#)

Синтаксис:

Минимум (Выражение)

Описание:

Функция получает минимальное значение. В качестве параметра можно передавать **Массив**. В этом случае функция будет применена к содержимому массива.

Пример:

МИНИМУМ(Остатки.Количество)
[Копировать в буфер обмена](#)**СРЕДНЕЕ (AVG)****Синтаксис:**

Среднее (Выражение)

Описание:

Функция получает среднее значение для значений, отличных от **NULL**. В качестве параметра можно передавать **Массив**. В этом случае функция будет применена к содержимому массива.

Пример:

СРЕДНЕЕ(Остатки.Количество)
[Копировать в буфер обмена](#)**МАССИВ (ARRAY)****Синтаксис:**

Массив ([Различные] Выражение)

Описание:

Функция формирует массив, содержащий значение выражения для каждой детальной записи.

В качестве параметра можно использовать таблицу значений. При этом результатом работы функции будет массив, содержащий значения первой колонки таблицы значений, переданной в качестве параметра.

Если выражение содержит функцию **Массив**, то считается, что данное выражение является агрегатным.

Если указано ключевое слово **Различные**, то получаемый массив не будет содержать дублирующихся значений.

Пример:

ФункцииОтчетов.СтандартноеОтклонение(Массив(Сумма))
[Копировать в буфер обмена](#)**ТАБЛИЦАЗНАЧЕНИЙ (VALUETABLE)****Синтаксис:**

ТаблицаЗначений([Различные] Выражение1 [КАК ИмяКолонки1][, Выражение2 [КАК ИмяКолонки2]], ...)

Описание:

Функция формирует таблицу значений, содержащую столько колонок, сколько параметров у функции. Детальные записи получают из наборов данных, которые нужны для вычисления всех выражений, указанных в параметрах функции.

Если параметрами функции выступают поля-остатки, то в результирующую таблицу значений попадут значения для записей по уникальным комбинациям измерений из других периодов. При этом значения получают только для полей-остатков, измерений, счетов, полей периодов и их реквизитов. Значения остальных полей в записях из других периодов считаются равными **NULL**.

Если выражение содержит функцию **ТаблицаЗначений**, то считается, что данное выражение является агрегатным.

Если указано ключевое слово **Различные**, то в получаемой таблице значений не будет строк, содержащих одинаковые данные.

Для каждой колонки можно задать имя с помощью ключевого слова **КАК**, следующего после выражения, формирующего значение колонки.

Пример:

ФункцииОтчетов.ТаблицуЗначенийВСтроку(ТаблицаЗначений(Склад **КАК** Склад, КоличествоОстаток **КАК** Остаток))
[Копировать в буфер обмена](#)**СВЕРНУТЬ (GROUPBY)****Синтаксис:**

Свернуть (<Выражение>, <НомераКолонок>)

Описание:

Функция предназначена для удаления дубликатов из массива.

Параметры:

<Выражение>

Тип: **Массив** или **ТаблицаЗначений**. Для значения, расположенного в данном формальном параметре, выполняется удаление дубликатов.

<НомераКолонок>

Тип строка. Используется, если параметр **Выражение** имеет тип **ТаблицаЗначений**. Номера или имена (через запятую) колонок таблицы значений, среди которых нужно искать дубликаты. По умолчанию все колонки.
Пример:

Свернуть(ТаблицаЗначений(НомерТелефона, Адрес), "НомерТелефона");
[Копировать в буфер обмена](#)

ПОЛУЧИТЬЧАСТЬ (GETPART)**Синтаксис:**

ПолучитьЧасть (<Выражение>, <НомерКолонок>)

Описание:

Функция получает таблицу значений, которая содержит указанные колонки исходной таблицы значений.

Параметры:

<Выражение>

Тип: **ТаблицаЗначений**. Таблица значений, из которой нужно получить колонки.

<НомераКолонок>

Тип: **Строка**. Номера или имена (через запятую) колонок таблицы значений, которые нужно получить.

Возвращаемое значение:

Таблица значений, содержащая только те колонки, которые указаны в параметре **НомераКолонок**.

Пример:

[ПолучитьЧасть\(Свернуть\(ТаблицаЗначений\(НомерТелефона, Адрес\), "НомерТелефона"\), "НомерТелефона"\);](#)

[Копировать в буфер обмена](#)

УПОРЯДОЧИТЬ (ORDER)**Синтаксис:**

Упорядочить (<Выражение>, <НомераКолонок>)

Описание:

Предназначена для упорядочивания элементов массива и таблицы значений.

Параметры:

<Выражение>

Тип: **Массив** или **ТаблицаЗначений**. Объект, который требуется упорядочить.

<НомераКолонок>

Используется, если параметр **Выражение** имеет тип **ТаблицаЗначений**. Номера или имена (через запятую) колонок таблицы значений, которые требуется упорядочить. По умолчанию все колонки.

После каждой колонки может содержаться направление упорядочивания или признак автоматического упорядочивания.

Возвращаемое значение:

Массив или таблица значений, которые упорядочены в соответствии с переданными параметрами.

Пример:

[Упорядочить\(ТаблицаЗначений\(НомерТелефона, Адрес, ДатаЗвонка\), "ДатаЗвонка Убыв"\);](#)

[Копировать в буфер обмена](#)

СОЕДИНИТЬСТРОКИ (JOINSTRINGS)**Синтаксис:**

СоединитьСтроки(<Значения>, <РазделительЭлементов>, <РазделительКолонок>)

Описание:

Предназначена для объединения строк в одну строку.

Параметры:

<Значения>

Значения, строковые представления которых нужно объединить в одну строку. Если типом значения параметра является **Массив**, то в строку будут объединяться элементы массива. Если типом значения параметра является **ТаблицаЗначений**, то в строку будут объединяться все колонки и строки таблицы.

<РазделительЭлементов>

Тип: **Строка**. Содержит текст, который нужно использовать в качестве разделителя между элементами массива и строками таблицы значений. По умолчанию символ перевода строк.

<РазделительКолонок>

Тип: **Строка**. Содержит текст, который нужно использовать в качестве разделителя между колонками таблицы значений. По умолчанию "; ".

Возвращаемое значение:

Объединенная строка.

Пример:

[СоединитьСтроки\(ТаблицаЗначений\(НомерТелефона, Адрес\)\);](#)

[Копировать в буфер обмена](#)

ГРУППОВАЯОБРАБОТКА (GROUPPROCESSING)**Синтаксис:**

ГрупповаяОбработка(<Выражения>, <ВыраженияИерархии>, <ИмяГруппировки>)

Описание:

Формируется таблица значений, содержащая значения параметров (в колонках) для каждой групповой записи (в строках). В случае использования иерархической группировки каждый уровень иерархии обрабатывается отдельно. Значения для иерархических записей также помещаются в таблицу данных. В возвращаемом объекте эта таблица значений будет размещена в свойстве [Данные](#).

В свойство [ТекущийЭлемент](#) помещается строка передаваемой таблицы значений, для которой в настоящий момент вычисляется функция.

При реализации функции, которая в качестве параметра может принимать данные групповой обработки, следует учитывать, что в качестве значения в функцию может быть передано значение `NULL`. Например, такое может произойти в случае расчета функции [ГрупповаяОбработка\(\)](#) с указанием имени группировки, которая в настоящий момент недоступна.

Параметры:

<Выражения>

Строка с перечисленными через запятую выражениями, которые нужно вычислить. После каждого выражения возможно наличие необязательного ключевого слова **КАК** и имени колонки результирующей таблицы значений. Каждое выражение описывает колонку формируемой таблицы значений.

<ВыраженияИерархии>

Выражения, которые нужно вычислить для иерархических записей. Аналогично параметру [Выражения](#) с тем отличием, что параметр [Выражения](#) используется для неиерархических записей, а параметр [ВыраженияИерархии](#) используется для иерархических записей. Если параметр не указан, то для вычисления значений для иерархических записей используется выражения, указанные в параметре [Выражение](#).

<ИмяГруппировки>

Имя группировки, в которой нужно вычислять группировку обработки. Если не указано, то вычисление происходит в текущей группировке. Если вычисление идет в таблице и параметр содержит пустую строку или не указан, то значение вычисляется для группировки – строки. Компонщик макета при генерации макета при генерации макета компонки данных заменяет данное имя именем группировки в результирующем макете. Если группировка недоступна, то функция будет заменена на значение `NULL`.

Возвращаемое значение:

Объект [ДанныеГрупповойОбработкиКомпоновкиДанных](#).

Пример:

Пример реализации функции [АВСКлассификация\(\)](#), которая возвращает значение 1, если значение составляет 75 % от общей суммы, 2, если процент между 75 % и 95 %, и 3 в других случаях.

[Копировать в буфер обмена](#)

```
// Рассчитать АВС классификацию.
Функция АВСКлассификация(Данные) Экспорт
Перем ТаблицаЗначений;
Если Данные = Null Тогда
    Возврат Null;
КонецЕсли;
Если ТипЗнч(Данные) <> Тип("ДанныеГрупповойОбработкиКомпоновкиДанных") Тогда
    Сообщить(ТипЗнч(Данные));
    ВызватьИсключение "В функции АВСКлассификация() можно передавать только объект типа ДанныеГрупповойОбработкиКомпоновкиДанных";
КонецЕсли;
Если Не Данные.ВременныеДанныеОбработки.Свойство("АВСКлассификацияТаблицаЗначений", ТаблицаЗначений) Тогда
    // выполним расчет классификации один раз при первом вызове
    // а потом запоним выполненный расчет и дальше будем этот
    // расчет использовать
    ТаблицаЗначений = Данные.Данные.Скопировать();
    ТаблицаЗначений.Колонки.Добавить("Номер", Новый ОписаниеТипов("Число"));
    Номер = 0;
    ОбщаяСумма = 0;
    Для Каждого СтрокаТаблицыЗначений Из ТаблицаЗначений Цикл
        СтрокаТаблицыЗначений.Номер = Номер;
        Номер = Номер + 1;
        Если СтрокаТаблицыЗначений[0] <> NULL Тогда
            ОбщаяСумма = ОбщаяСумма + СтрокаТаблицыЗначений[0];
        КонецЕсли;
    КонецЦикла;
    ТаблицаЗначений.Сортировать(ТаблицаЗначений.Колонки[0].Имя + " Убыв");
    ТаблицаЗначений.Индексы.Добавить("Номер");
    НакопленнаяСумма = 0;
    ИндексКлассаА = Неопределено;
    ИндексКлассаВ = Неопределено;
    Для Каждого СтрокаТаблицыЗначений Из ТаблицаЗначений Цикл
        Если СтрокаТаблицыЗначений[0] <> NULL Тогда
            НакопленнаяСумма = НакопленнаяСумма + СтрокаТаблицыЗначений[0];
        КонецЕсли;
        Если ОбщаяСумма = 0 Тогда
            Процент = 1;
        Иначе
            Процент = НакопленнаяСумма / ОбщаяСумма;
        КонецЕсли;
        Если Процент > 0.75 Тогда
            Если ИндексКлассаА = Неопределено Тогда
                ИндексКлассаА = ТаблицаЗначений.Индекс(СтрокаТаблицыЗначений);
            ИначеЕсли Процент > 0.90 Тогда
                Если ИндексКлассаВ = Неопределено Тогда
                    ИндексКлассаВ = ТаблицаЗначений.Индекс(СтрокаТаблицыЗначений);
                КонецЕсли;
                Прервать;
            КонецЕсли;
        КонецЕсли;
    КонецЦикла;
    Данные.ВременныеДанныеОбработки.Вставить("АВСКлассификацияТаблицаЗначений", ТаблицаЗначений);
    Данные.ВременныеДанныеОбработки.Вставить("АВСКлассификацияИндексКлассаА", ИндексКлассаА);
    Данные.ВременныеДанныеОбработки.Вставить("АВСКлассификацияИндексКлассаВ", ИндексКлассаВ);
КонецЕсли;
Если Данные.ТекущийЭлемент = Неопределено Тогда
    // Итог по группировке.
    Возврат Null;
Иначе
    Строка = ТаблицаЗначений.Найти(Данные.Данные.Индекс(Данные.ТекущийЭлемент), "Номер");
    Если Строка = Неопределено Тогда
        Возврат Null;
    Иначе
        Индекс = ТаблицаЗначений.Индекс(Строка);
        Если Индекс <= Данные.ВременныеДанныеОбработки.АВСКлассификацияИндексКлассаА Тогда
            Возврат 1;
        ИначеЕсли Индекс <= Данные.ВременныеДанныеОбработки.АВСКлассификацияИндексКлассаВ Тогда
            Возврат 2;
        Иначе
            Возврат 3;
        КонецЕсли;
    КонецЕсли;
КонецЕсли;
КонецФункции
```

В выражении компоновки для получения класса можно использовать следующее выражение (например, в ресурсе или пользовательском поле):

[Копировать в буфер обмена](#)

[АВСКлассификация](#)([ГрупповаяОбработка](#)("Сумма(СуммаОборот)"))

КАЖДЫЙ (EVERY)

Синтаксис:

Каждый(<X>)

Описание:

Агрегатная функция **Каждый** определяет наличие в переданном наборе хотя бы одного значения **Ложь**.

Возвращаемое значение:

Истина – если в переданном наборе не содержится значения **Ложь**.

Ложь – если в переданном наборе содержится хотя бы одно значение **Ложь**.

ЛЮБОЙ (ANY)

Синтаксис:

Любой(<X>)

Описание:

Агрегатная функция **Любой** определяет наличие в переданном наборе хотя бы одного значения **Истина**.

Возвращаемое значение:

Истина – если в переданном наборе содержится хотя бы одно значение **Истина**.

Ложь – если в переданном наборе не содержится значения **Истина**.

СТАНДАРТНОЕ ОТКЛОНЕНИЕ ГЕНЕРАЛЬНОЙ СОВОКУПНОСТИ (STDDEV_POP)

Синтаксис:

СтандартноеОтклонениеГенеральнойСовокупности(<X>)

Описание:

Вычисляет стандартное отклонение генеральной совокупности переданного набора.

Вычисление выполняется по формуле: $\text{SQRT}(\text{ДисперсияГенеральнойСовокупности}(X))$.

Возвращаемое значение:

Результат вычисления функции.

СТАНДАРТНОЕ ОТКЛОНЕНИЕ ВЫБОРКИ (STDDEV_SAMP)

Синтаксис:

СтандартноеОтклонениеВыборки(<X>)

Описание:

Вычисляет стандартное отклонение выборки переданного набора.

Вычисление выполняется по формуле: $\text{SQRT}(\text{ДисперсияВыборки}(X))$.

Возвращаемое значение:

Результат вычисления функции.

ДИСПЕРСИЯ ВЫБОРКИ (VAR_SAMP)

Синтаксис:

ДисперсияВыборки(<X>)

Описание:

Вычисляет выборочную дисперсию для переданного набора.

Вычисление выполняется по формуле: $(\text{SUM}(X^2) - \text{SUM}(X)^2 / \text{COUNT}(X)) / (\text{COUNT}(X) - 1)$. Если количество записей в наборе равно 1 ($\text{COUNT}(X) = 1$), возвращаемое значение равно **NULL**.

Возвращаемое значение:

Результат вычисления функции.

ДИСПЕРСИЯ ГЕНЕРАЛЬНОЙ СОВОКУПНОСТИ (VAR_POP)

Синтаксис:

ДисперсияГенеральнойСовокупности(<X>)

Описание:

Вычисляет дисперсию генеральной совокупности для переданного набора. Значения **NULL** игнорируются.

Вычисление выполняется по формуле: $(\text{SUM}(X^2) - \text{SUM}(X)^2 / \text{COUNT}(X)) / \text{COUNT}(X)$.

Возвращаемое значение:

Результат вычисления функции.

КОВАРИАЦИЯ ГЕНЕРАЛЬНОЙ СОВОКУПНОСТИ (COVAR_POP)

Синтаксис:

КовариацияГенеральнойСовокупности(<Y>, <X>)

Описание:

Вычисляет ковариацию совокупности множества пар переданных наборов.

Вычисление выполняется по формуле: $(\text{SUM}(Y * X) - \text{SUM}(X) * \text{SUM}(Y) / N) / N$. N – число пар значений X и Y из переданных наборов, где ни значение X, ни значение Y не равны NULL. Пары, где есть хотя бы одно значение NULL, игнорируются.

Возвращаемое значение:

Результат вычисления функции.

КОВАРИАЦИЯ ВЫБОРКИ (COVAR_SAMP)

Синтаксис:

КовариацияВыборки(<Y>, <X>)

Описание:

Вычисляет образец ковариации множества пар переданных наборов.

Вычисление выполняется по формуле: $(\text{SUM}(Y * X) - \text{SUM}(Y) * \text{SUM}(X) / N) / (N - 1)$. N – число пар значений X и Y из переданных наборов, где ни значение X, ни значение Y не равны NULL. Пары, где есть хотя бы одно значение NULL, игнорируются.

Возвращаемое значение:

Результат вычисления функции или NULL – если функция применяется к пустым множествам.

КОРРЕЛЯЦИЯ (CORR)

Синтаксис:

Корреляция(<Y>, <X>)

Описание:

Вычисляет коэффициент корреляции множества пар переданных наборов. Вычисление выполняется по формуле:

$\text{КовариацияГенеральнойСовокупности}(Y, X) / (\text{СтандартноеОтклонениеГенеральнойСовокупности}(Y) * \text{СтандартноеОтклонениеГенеральнойСовокупности}(X))$. Пары, где есть хотя бы одно значение NULL, игнорируются.

Возвращаемое значение:

Результат вычисления функции или NULL – если функция применяется к пустым множествам.

РЕГРЕССИЯ НАКЛОН (REGR_SLOPE)

Синтаксис:

РегрессияНаклон(<Y>, <X>)

Описание:

Вычисляет наклон линии.

Вычисление выполняется по формуле: $\text{КовариацияГенеральнойСовокупности}(Y, X) / \text{ДисперсияГенеральнойСовокупности}(X)$. Пары, где есть хотя бы одно значение NULL, игнорируются.

Возвращаемое значение:

Результат вычисления функции.

РЕГРЕССИЯ ОТРЕЗОК (REGR_INTERCEPT)

Синтаксис:

РегрессияОтрезок(<Y>, <X>)

Описание:

Вычисляет Y-точку пересечения линии регресса.

Вычисление выполняется по формуле: $\text{Среднее}(Y) - \text{РегрессияНаклон}(Y, X) * \text{Среднее}(X)$. Пары, где есть хотя бы одно значение NULL, игнорируются.

Возвращаемое значение:

Результат вычисления функции.

РЕГРЕССИЯ КОЛИЧЕСТВО (REGR_COUNT)

Синтаксис:

РегрессияКоличество(<Y>, <X>)

Описание:

Вычисляет количество пар, не содержащих NULL.

Возвращаемое значение:

Результат вычисления функции.

РЕГРЕССИЯ R2 (REGR_R2)

Синтаксис:

Регрессия_R2(<Y>, <X>)

Описание:

Вычисляет коэффициент детерминации для регрессии. Функция вычисляется без учета пар значений, содержащих значение `NULL`.

Возвращаемое значение:

- `NULL`, если ДисперсияГенеральнойСовокупности(`X`) равно 0.
- 1, если ДисперсияГенеральнойСовокупности(`Y`) равно 0 и ДисперсияГенеральнойСовокупности(`X`) не равно 0.
- $\text{Pow}(\text{Корреляция}(\text{Y}, \text{X}), 2)$, если ДисперсияГенеральнойСовокупности(`Y`) больше 0 и ДисперсияГенеральнойСовокупности(`X`) не равно 0.

РЕГРЕССИЯСРЕДНЕЕХ (REGR_AVGX)**Синтаксис:**

РегрессияСреднееХ(<Y>, <X>)

Описание:

Вычисляет среднее из независимых переменных <X> для линии регрессии, после исключения пар, где хотя бы одно значение равно `NULL`.

Вычисление выполняется по формуле: `Среднее(X)`.

Возвращаемое значение:

Результат вычисления функции.

РЕГРЕССИЯСРЕДНЕЕY (REGR_AVGY)**Синтаксис:**

РегрессияСреднееY(<Y>, <X>)

Описание:

Вычисляет среднее из зависимых переменных <Y> для линии регрессии, после исключения пар, где хотя бы одно значение равно `NULL`.

Вычисление выполняется по формуле: `Среднее(Y)`.

Возвращаемое значение:

Результат вычисления функции.

РЕГРЕССИЯSXX (REGR_SXX)**Синтаксис:**

РегрессияSXX(<Y>, <X>)

Описание:

Выполняет расчет по следующей формуле:

$\text{РегрессияКоличество}(\text{Y}, \text{X}) * \text{ДисперсияГенеральнойСовокупности}(\text{X})$. Пары, где есть хотя бы одно значение `NULL`, игнорируются.

Возвращаемое значение:

Результат вычисления функции.

РЕГРЕССИЯSYX (REGR_SYX)**Синтаксис:**

РегрессияSYX(<Y>, <X>)

Описание:

Выполняет расчет по следующей формуле:

$\text{РегрессияКоличество}(\text{Y}, \text{X}) * \text{ДисперсияГенеральнойСовокупности}(\text{Y})$. Пары, где есть хотя бы одно значение `NULL`, игнорируются.

Возвращаемое значение:

Результат вычисления функции.

РЕГРЕССИЯSXY (REGR_SXY)**Синтаксис:**

РегрессияSXY(<Y>, <X>)

Описание:

Выполняет расчет по следующей формуле:

$\text{РегрессияКоличество}(\text{Y}, \text{X}) * \text{КовариацияГенеральнойСовокупности}(\text{Y}, \text{X})$. Пары, где есть хотя бы одно значение `NULL`, игнорируются.

Возвращаемое значение:

Результат вычисления функции.

МЕСТОВПОРЯДКЕ (RANK)**Синтаксис:**

МестоВПорядке(<Порядок>, <ПорядокИерархии>, <ИмяГруппировки>)

Описание:

Функция вычисляет, какое место имеет текущая запись среди записей текущей группировки, если ее упорядочить в порядке, указанном в параметрах функции. Нумерация начинается с 1.

Параметры:

<Порядок>

Тип: **Строка**. Содержит выражения, в последовательности которых нужно расположить групповые записи, разделенные через запятую. Направление упорядочивания управляется при помощи слов **Возр**, **Убыв**. После поля также можно указать строку **Автоупорядочивание**, что обозначает, что при упорядочивании ссылок нужно использовать поля упорядочивания, определенные для объекта, на который указывает ссылка. Если последовательность не указана, то значение рассчитывается в последовательности группировки.

<ПорядокИерархии>

Тип: **Строка**. Строка, содержащая выражения упорядочивания для иерархических записей.

<ИмяГруппировки>

Тип: **Строка**. Имя группировки, в которой нужно вычислять функцию. Если не указано, то вычисление происходит в текущей группировке. Если вычисление выполняется в таблице и параметр содержит пустую строку или не указан, то значение вычисляется для группировки-строки. Компоновщик макета при генерации макета компоновки данных заменяет данное имя на имя группировки в результирующем макете. Если группировка не доступна, то функция будет заменена на значение **NULL**.

Возвращаемое значение:

Номер по порядку. Если в последовательности имеются две или более записей с одинаковым значением полей упорядочивания, то для всех этих записей функция возвращает одинаковое значение.

КЛАССИФИКАЦИЯABC (CLASSIFICATIONABC)

Синтаксис:

КлассификацияABC(<Значение>, <КоличествоГрупп>, <ПроцентыДляГрупп>, <ИмяГруппировки>)

Описание:

Выполняет ABC-классификацию каждой записи в заданной группировке.

Параметры:

<Значение>

Тип: **Строка**. Указывает выражение, для которого следует рассчитывать классификацию.

<КоличествоГрупп>

Тип: **Число**. На сколько групп необходимо разбить набор значений.

<ПроцентыДляГрупп>

Тип: **Строка**. Объем, в %, каждой группы разбиения, кроме последней группы. Перечисляются в строке через запятую.

<ИмяГруппировки>

Тип: **Строка**. Имя группировки, в которой нужно вычислять функцию. Если не указано, то вычисление происходит в текущей группировке. Если вычисление идет в таблице и параметр содержит пустую строку или не указан, то значение вычисляется для группировки-строки. Компоновщик макета при генерации макета компоновки данных заменяет данное имя на имя группировки в результирующем макете. Если группировка не доступна, то функция будет заменена на значение **NULL**.

Возвращаемое значение:

Номер класса. **1** соответствует классу **A**, **2** – классу **B**, **3** – классу **C** и т. д.

Пример:

КлассификацияABC(СуммаОборот, 3, "15, 25")

[Копировать в буфер обмена](#)

10.3.4.7. Другие операции

Операция ВЫБОР (CASE)

Описание:

Операция **ВЫБОР** предназначена для осуществления выбора одного из нескольких значений при выполнении некоторых условий.

Пример:

ВЫБОР Когда Сумма > 1000 Тогда Сумма Иначе 0 Конец

[Копировать в буфер обмена](#)

10.3.4.8. Правила сравнения двух значений

Если типы сравниваемых значений отличаются друг от друга, то отношения между значениями определяются на основании приоритета типов:

- **NULL** (самый низший);
- Булево;
- Число;
- Дата;
- Строка;
- ссылочные типы.

Отношения между различными ссылочными типами определяются на основе ссылочных номеров таблиц, соответствующих тому или иному типу.

Если типы данных совпадают, то производится сравнение значений по следующим правилам:

- у типа **Булево** значение **Истина** больше значения **Ложь**;
- у типа **Число** обычные правила сравнения для чисел;
- у типа **Дата** более ранние даты меньше более поздних;

- у типа *Строка* сравнения строк в соответствии с установленными национальными особенностями базы данных;
- ссылочные типы сравниваются на основе своих значений (номера записи и т. п.).

10.3.4.9. Работа со значением NULL

Любая операция, в которой значение одного из операндов *NULL*, будет давать результат *NULL*.

Есть исключения:

- операция *И* будет возвращать *NULL* только в случае, если ни один из операндов не имеет значение *Ложь*;
- операция *ИЛИ* будет возвращать *NULL* только в случае, если ни один из операндов не имеет значение *Истина*.

10.3.4.10. Приоритеты операций

Операции имеют следующие приоритеты (первая строка имеет низший приоритет):

- *ИЛИ*;
- *И*;
- *НЕ*;
- *В, ЕСТЬ NULL, ЕСТЬ НЕ NULL*;
- *=, <>, <=, <, >=, >*;
- Бинарный *+*, Бинарный *-*;
- ***, */*, *%*;
- Унарный *+*, Унарный *-*.

10.3.4.11. Функции

ВЫЧИСЛИТЬ (EVAL)

Синтаксис:

Вычислить (<Выражение>, <Группировка>, <ТипРасчета>)

Описание:

Функция **ВЫЧИСЛИТЬ** предназначена для вычисления выражения в контексте некоторой группировки.

Функция используется для совместимости с предыдущими версиями «1С:Предприятия». Вместо данной функции рекомендуется использовать функцию **ВычислитьВыражение**.

Параметры:

<Выражение>

Строка, содержащая вычисляемое выражение.

<Группировка>

Строка, содержащая имя группировки, в контексте которой необходимо вычислить выражение. Если в качестве имени группировки используется пустая строка, вычисление будет выполнено в контексте текущей группировки. Если в качестве имени группировки будет использована строка **ОбщийИтог**, вычисление будет выполнено в контексте общего итога. В остальных случаях вычисление будет выполняться в контексте родительской группировки с таким именем.

<ТипРасчета>

Строка, содержащая тип расчета. Если данный параметр имеет значение **ОбщийИтог**, выражение будет вычисляться для всех записей группировки. Если значение параметра – **Группировка**, значения будут вычисляться для текущей групповой записи группировки.

Пример:

[Сумма\(Продажи.СуммаОборот\) / ВЫЧИСЛИТЬ\("Сумма\(Продажи.СуммаОборот\)", "ОбщийИтог"\)](#)

[Копировать в буфер обмена](#)

В данном примере в результате получится отношение суммы по полю **Продажи.СуммаОборот** записи группировки к сумме того же поля во всей компоновке.

ВЫЧИСЛИТЬВЫРАЖЕНИЕ (EVALEXPRESSION)

Синтаксис:

ВычислитьВыражение (<Выражение>, <Группировка>, <ОбластьВычисления>, <Начало>, <Конец>, <Сортировка>, <ИерархическаяСортировка>, <ОбработкаОдинаковыхЗначени

Описание:

Функция предназначена для вычисления выражения в контексте некоторой группировки.

Функция учитывает отбор группировок, но не учитывает иерархические отборы.

Функция не может применяться к группировке в групповом отборе этой группировки. Например, в отборе группировки **Номенклатура** нельзя использовать выражение **ВычислитьВыражение("Сумма(СуммаОборот)", , "ОбщийИтог") > 1000**. Но такое выражение можно использовать в иерархическом отборе.

Если конечная запись предшествует начальной, то считается, что записи для расчета детальных данных и расчета агрегатных функций отсутствуют.

При расчете интервальных выражений для общего итога (параметр **Группировка** имеет значение **ОбщийИтог**) считается, что записи для расчета детальных данных и расчета агрегатных функций отсутствуют.

Если выражение упорядочивания содержит поля, которые не могут быть использованы в группировке, то компоновщик макета при генерации выражения функции **ВычислитьВыражение** заменяет функцию **ВычислитьВыражение** на *NULL*.

Параметры:

<Выражение>

Тип: *Строка*. Выражение, которое нужно вычислить.

<Группировка>

Тип: **Строка**. Содержит имя группировки, в контексте которой необходимо вычислить выражение. В случае, если в качестве имени группировки используется пустая строка, вычисление будет выполнено в контексте текущей группировки. Если в качестве имени группировки будет использована строка **ОбщийИтог**, вычисление будет выполнено в контексте общего итога. В остальных случаях вычисление будет выполняться в контексте родительской группировки с таким именем.

Например:

`Сумма(Продажи.СуммаОборот)/Вычислить("Сумма(Продажи.СуммаОборот)", "ОбщийИтог")`

[Копировать в буфер обмена](#)

В данном примере в результате получится отношение суммы по полю `Продажи.СуммаОборот` записи группировки к сумме того же поля во всей компоновке.

<ОбластьВычисления>

Тип: **Строка**. Параметр может принимать следующие значения:

- **ОбщийИтог** – выражение будет вычисляться для всех записей группировки.
- **Иерархия** – выражение будет вычисляться для родительской иерархической записи, если таковая имеется, и для всей группировки, если родительской иерархической записи не имеется.
- **Группировка** – выражение будет вычисляться для текущей групповой записи группировки.
- **ГруппировкаНеРесурса** – при вычислении функции для групповой записи по ресурсам выражение будет вычислено для первой групповой записи исходной группировки.

При вычислении функции `ВычислитьВыражение()` со значением `ГруппировкаНеРесурса` для групповых записей, не являющихся группировками по ресурсам, функция вычисляется, как в случае, если значение параметра было равно значению `Группировка`.

Компоновщик макета компоновки данных при генерации макета компоновки данных при выводе в макет поля-ресурса, по которому выполняется группировка, помещает в макет выражение, вычисляемое при помощи функции `ВычислитьВыражение()`, с указанием параметра `ГруппировкаНеРесурса`. Для остальных ресурсов в группировке по ресурсу помещаются обычные выражения ресурсов.

<Начало>

Тип: **Строка**. Указывает, с какой записи нужно начинать фрагмент, в котором рассчитывать агрегатные функции выражения, и из какой записи получать значения полей вне агрегатных функций. Значение может быть одним из следующих:

- **Первая (First)**. Необходимо получать первую запись группировки. После слова в скобках можно указывать выражение, результат которого будет использоваться как смещение от начала группировки. Получаемое значение должно целым числом больше нуля. Например, `Первая(3)` – получение третьей записи от начала группировки.

Если первая запись выходит за пределы группировки, то считается, что записей нет. Например, если записей 3, а требуется получить `Первая(4)`, то считается, что записей нет.

- **Последняя (Last)**. Необходимо получить последнюю запись группировки. После слова в скобках можно указывать выражение, результат которого будет использоваться как смещение от конца группировки. Получаемое значение должно целым числом больше нуля. Например, `Последняя(3)` – получение третьей записи от конца группировки.

Если последняя запись выходит за пределы группировки, то считается, что записей нет. Например, если записей 3, а требуется получить `Последняя(4)`, то считается, что записей нет.

- **Предыдущая (Previous)**. Необходимо получить предыдущую запись группировки. После слова в скобках можно указывать выражение, результат которого будет использоваться как смещение назад от текущей записи группировки. Например, `Предыдущая(2)` – получение предыдущей от предыдущей записи.

Если предыдущая запись выходит за пределы группировки (например, для второй записи группировки требуется получить `Предыдущая(3)`, то получается первая запись группировки.

При получении предыдущей записи для итога по группировке считается, что получается первая запись.

- **Следующая (Next)**. Необходимо получить следующую запись группировки. После слова в скобках можно указывать выражение, результат которого будет использоваться как смещение вперед от текущей записи группировки. Например, `Следующая(2)` – получение следующей от следующей записи.

Если следующая запись выходит за пределы группировки, то считается, что записей нет. Например, если записей 3 и для третьей записи получают `Следующая()`, то считается, что записей нет.

При получении следующей записи для итога по группировке считается, что записи нет.

- **Текущая (Current)**. Необходимо получить текущую запись.

При получении для итога по группировке получается первая запись.

- **ОграничивающееЗначение (BoundaryValue)**. Необходимо получить запись по указанному значению. После слова `ОграничивающееЗначение` в скобках следует указать выражение первого поля упорядочивания, со значения которого нужно начинать фрагмент.

В качестве записи будет получаться первая запись, значение поля упорядочивания у которой больше или равно указанному значению. Например, если в качестве поля упорядочивания используется поле `Период`, оно имеет значения `01.01.2010`, `01.02.2010`, `01.03.2010` и требуется получить `ОграничивающееЗначение(ДатаВремя(2010, 1, 15))`, то будет получена запись с датой `01.02.2010`.

<Конец>

Тип: **Строка**. Указывает, до какой записи нужно продолжать фрагмент, в котором рассчитывать агрегатные функции выражения. Значение может быть одним из следующих:

- **Первая (First)**. Необходимо получать первую запись группировки. После слова в скобках можно указывать выражение, результат которого будет использоваться как смещение от начала группировки. Получаемое значение должно целым числом больше нуля. Например, `Первая(3)` – получение третьей записи от начала группировки.

Если первая запись выходит за пределы группировки, то считается, что записей нет. Например, если записей 3, а требуется получить `Первая(4)`, то считается, что записей нет.

- **Последняя (Last)**. Необходимо получить последнюю запись группировки. После слова в скобках можно указывать выражение, результат которого будет использоваться как смещение от конца группировки. Получаемое значение должно целым числом больше нуля. Например, `Последняя(3)` – получение третьей записи от конца группировки.

Если последняя запись выходит за пределы группировки, то считается, что записей нет. Например, если записей 3, а требуется получить `Последняя(4)`, то считается, что записей нет.

- **Предыдущая (Previous)**. Необходимо получить предыдущую запись группировки. После слова в скобках можно указывать выражение, результат которого будет использоваться как смещение назад от текущей записи группировки. Например, `Предыдущая(2)` – получение предыдущей от предыдущей записи.

Если предыдущая запись выходит за пределы группировки (например, для второй записи группировки требуется получить `Предыдущая(3)`, то получается первая запись группировки.

При получении предыдущей записи для итога по группировке считается, что получается первая запись.

- **Следующая (Next)**. Необходимо получить следующую запись группировки. После слова в скобках можно указывать выражение, результат которого будет использоваться

как смещение вперед от текущей записи группировки. Например, *Следующая(2)* – получение следующей от следующей записи.

Если следующая запись выходит за пределы группировки, то считается, что записей нет. Например, если записей 3 и для третьей записи получают *Следующая()*, то считается, что записей нет.

При получении следующей записи для итога по группировке считается, что записи нет.

- **Текущая (Current)**. Необходимо получить текущую запись.

При получении для итога по группировке получается первая запись.

- **ОграничивающееЗначение (BoundaryValue)**. Необходимо получить запись по указанному значению. После слова *ОграничивающееЗначение* в скобках следует указать выражение первого поля упорядочивания, со значения которого необходимо завершить фрагмент.

В качестве записи будет получаться первая запись, значение поля упорядочивания у которой больше или равно указанному значению. Например, если в качестве поля упорядочивания используется поле *Период*, оно имеет значения **01.01.2010**, **01.02.2010**, **01.03.2010** и требуется получить *ОграничивающееЗначение(ДатаВремя(2010, 1, 15))*, то будет получена запись с датой **01.02.2010**.

<Сортировка>

Тип: *Строка*. Перечисляются выражения, разделенные запятыми, которые описывают правила упорядочивания. Если не указана, то упорядочивание выполняется так же, как и у группировки, для которой вычисляется выражение. После каждого выражения можно указать ключевое слово *Возр* (для упорядочивания по возрастанию), *Убыв* (для упорядочивания по убыванию) и *Автоупорядочивание* (для упорядочивания ссылочных полей по полям, по которым нужно упорядочивать объект, на который выполняется ссылка). Слово *Автоупорядочивание* может использоваться как со словом *Возр*, так и со словом *Убыв*.

<ИерархическаяСортировка>

Тип: *Строка*. Аналогично параметру *Сортировка*. Применяется для упорядочивания иерархических записей. Если не указано, компоновщик макета генерирует упорядочивание в соответствии с упорядочиванием, указанным в параметре *Сортировка*.

<ОбработкаОдинаковыхЗначенийПорядка>

Тип: *Строка*. Указывает правило определения предыдущей или следующей записи, если есть несколько записей с одинаковым значением упорядочивания:

- **Отдельно (Separately)** обозначает, что для определения предыдущей и следующей записей используется последовательность упорядоченных записей. Значение по умолчанию.
- **Вместе (Together)** обозначает, что предыдущая и следующие записи определяются на основании значений выражений упорядочивания.

Например, если полученная последовательность упорядочена по дате:

№	Дата	ФИО	Значение
1	01 января 2001 года	Иванов М.	10
2	02 января 2001 года	Петров С.	20
3	02 января 2001 года	Сидоров Р.	30
4	03 января 2001 года	Петров С.	40

Если значение параметра равно *Отдельно*, то:

- Предыдущей записью к записи 3 будет запись 2.
- Если расчетный фрагмент определяется как *Текущая*, *Текущая* (соответственно, параметры *Начало* и *Конец*), то для записи 2 этот фрагмент будет состоять из одной записи 2. Выражение *ВычислитьВыражение("Сумма(Значение)", , , Текущая, Текущая)* будет равно 20.

Если значение параметра равно *Вместе*, то:

- Предыдущей записью к записи 3 будет запись 1.
- Если расчетный фрагмент определяется как *Текущая*, *Текущая* (соответственно, параметры *Начало* и *Конец*), то для записи 2 этот фрагмент будет состоять из записей 2 и 3. Выражение *ВычислитьВыражение("Сумма(Значение)", , , Текущая, Текущая)* будет равно 50.

При указании значения параметра, равного *Вместе*, в параметрах *Начало* и *Конец* нельзя указывать смещение для позиций *Первая*, *Последняя*, *Предыдущая*, *Следующая*.

Пример:

Если требуется рассчитать сумму с накоплением, то можно использовать следующее выражение:

ВычислитьВыражение("Сумма(СуммаОборот)", , , "Первая", "Текущая")

[Копировать в буфер обмена](#)

Если требуется получить значение группировки в предыдущей строке, то можно использовать следующее выражение:

ВычислитьВыражение("Курс", , , "Предыдущая")

[Копировать в буфер обмена](#)

ВЫЧИСЛИТЬВЫРАЖЕНИЕСГРУППИРОВКОЙМАССИВ (EVALEXPRESSIONWITHGROUPARRAY)

Синтаксис:

ВычислитьВыражениеСГруппировкойМассив(<Выражение>, <ВыражениеПолейГруппировки>, <ОтборЗаписей>, <ОтборГруппировок>)

Описание:

Функция возвращает массив, каждый элемент которого содержит результат вычисления выражения для группировки по указанному полю.

Компоновщик макета при генерации макета преобразовывает параметры функции в термины полей макета компоновки данных. Например, поле *Контрагент* будет преобразовано в *НаборДанных.Контрагент*.

Компоновщик макета при генерации выражений для вывода пользовательского поля, в выражении которого присутствует только функция *ВычислитьМассивСГруппировкойМассив()*, генерирует выводимое выражение таким образом, чтобы выводимая информация была упорядочена. Например, для пользовательского поля с выражением:

ВычислитьВыражениеСГруппировкойМассив("Сумма(СуммаОборот)", "Контрагент")

[Копировать в буфер обмена](#)

Компоновщик макета сгенерирует для вывода следующее выражение:

СоединитьСтроки(Массив(Упорядочить(ВычислитьВыражениеСГруппировкойТаблицаЗначений("Представление(Сумма(НаборДанных.СуммаОборот))", Сумма(НаборДанных.СуммаОбо

[Копировать в буфер обмена](#)

Параметры:

<Выражение>

Тип: **Строка**. Выражение, которое нужно вычислить. Строка, например, `Сумма(СуммаОборот)`.

<ВыражениеПолейГруппировки>

Тип: **Строка**. Выражения полей группировки – выражения полей группировки, перечисленные через запятую. Например, `Контрагент, Партия`.

<ОтборЗаписей>

Тип: **Строка**. Выражение, описывающее отбор, применяемый к детальным записям. В выражении не поддерживается использование агрегатных функций. Например, `ПометкаУдаления = Ложь`.

<ОтборГруппировок>

Тип: **Строка**. Выражение, описывающее отбор, применяемый к групповым записям. Например, `Сумма(СуммаОборот) > &Параметр1`.

Пример:

[Максимум\(ВычислитьВыражениеСГруппировкойМассив \("Сумма\(СуммаОборот\)", "Контрагент"\)\)](#); [Копировать в буфер обмена](#)

ВЫЧИСЛИТЬВЫРАЖЕНИЕСГРУППИРОВКОЙТАБЛИЦАЗНАЧЕНИЙ (EVALEXPRESSIONWITHGROUPVALUETABLE)

Синтаксис:

ВычислитьВыражениеСГруппировкойТаблицаЗначений(<Выражения>, <ВыражениеПолейГруппировки>, <ОтборЗаписей>, <ОтборГруппировок>)

Описание:

Функция возвращает таблицу значений, каждая строка которой содержит результат вычисления выражений для группировки по указанному полю.

Компоновщик макета при генерации макета преобразовывает параметры функции в термины полей макета компоновки данных. Например, поле `Контрагент` будет конвертировано в `НаборДанных.Контрагент`.

Компоновщик макета при генерации выражений для вывода пользовательского поля, в выражении которого присутствует только функция `ВычислитьМассивСГруппировкойТаблицаЗначений()`, генерирует выводимое выражение таким образом, чтобы выводимая информация была упорядочена. Например, для пользовательского поля с выражением:

[ВычислитьВыражениеСГруппировкойТаблицаЗначений \("Контрагент, Сумма\(СуммаОборот\)", "Контрагент"\)](#) [Копировать в буфер обмена](#)

Компоновщик макета сгенерирует для вывода следующее выражение:

[СоединитьСтроки\(ПолучитьЧасть\(Упорядочить\(ВычислитьВыражениеСГруппировкойТаблицаЗначений \("НаборДанных.Контрагент, НаборДанных.КонтрагентПредставление, Сум"\)\)](#) [Копировать в буфер обмена](#)

Параметры:

<Выражения>

Тип: **Строка**. Выражения, которые нужно вычислить. В строке может быть перечислено несколько выражений через запятую. После каждого выражения может быть необязательное ключевое слово `КАК` и имя колонки таблицы значений. Например: `Контрагент, Сумма(СуммаОборот) Как ОбъемПродаж`.

<ВыражениеПолейГруппировки>

Тип: **Строка**. Выражения полей группировки – выражения полей группировки, перечисленные через запятую. Например, `Контрагент, Партия`.

<ОтборЗаписей>

Тип: **Строка**. Выражение, описывающее отбор, применяемый к детальным записям. В выражении не поддерживается использование агрегатных функций. Например, `ПометкаУдаления = Ложь`.

<ОтборГруппировок>

Тип: **Строка**. Выражение, описывающее отбор, применяемый к групповым записям. Например, `Сумма(СуммаОборот) > &Параметр1`.

Пример:

[ВычислитьВыражениеСГруппировкойТаблицаЗначений \("Контрагент КАК Контрагент, Сумма\(СуммаОборот\) Как ОбъемПродаж", "Контрагент"\)](#) [Копировать в буфер обмена](#)

Результатом работы данной функции будет таблица значений с колонками `Контрагент` и `ОбъемПродаж`, в которых будут находиться контрагенты с их объемами продаж.

УРОВЕНЬ (LEVEL)

Описание:

Функция предназначена для получения текущего уровня записи.

Пример:

[УРОВЕНЬ\(\)](#) [Копировать в буфер обмена](#)

УРОВЕНЬВГРУППИРОВКЕ (LEVELINGROUP)

Описание:

Функция предназначена для получения уровня записи относительно корня группировки.

Пример:

[УРОВЕНЬВГРУППИРОВКЕ\(\)](#) [Копировать в буфер обмена](#)

ЗНАЧЕНИЕЗАПОЛНЕНО (VALUEISFILLED)

Описание:

Возвращает **Истина**, если значение отлично от значения данного типа по умолчанию, отлично от значения `NULL`, отлично от пустой ссылки, отлично от значения **Неопределено**. Для логических значений осуществляется проверка на значение `NULL`. Для строк осуществляется проверка на отсутствие непробельных символов.

НОМЕРПОПОРЯДКУ (SERIALNUMBER)

Описание:

Получить следующий порядковый номер.

Пример:

НОМЕРПОПОРЯДКУ ()

[Копировать в буфер обмена](#)

НОМЕРПОПОРЯДКУВГРУППИРОВКЕ (GROUPSERIALNUMBER)

Описание:

Возвращает следующий порядковый номер в текущей группировке.

Пример:

НОМЕРПОПОРЯДКУВГРУППИРОВКЕ ()

[Копировать в буфер обмена](#)

ФОРМАТ (FORMAT)

Описание:

Получить отформатированную строку переданного значения. Форматная строка задается в соответствии с форматной строкой системы «1С:Предприятие».

Параметры:

- Значение,
- Форматная строка.

Пример:

ФОРМАТ(РасходныеНакладные.СуммаДок, "ЧДЦ=2")

[Копировать в буфер обмена](#)

НАЧАЛОПЕРИОДА (BEGINOFFPERIOD)

Описание:

Функция предназначена для выделения определенной даты из заданной даты.

Параметры:

- Выражение типа *Дата*.
- Тип периода – строка, содержащая одно из значений: *Минута*, *Час*, *День*, *Неделя*, *Месяц*, *Квартал*, *Год*, *Декада*, *Полугодие*.

Пример:

НАЧАЛОПЕРИОДА(ДатаВремя(2002, 10, 12, 10, 15, 34), "Месяц")

[Копировать в буфер обмена](#)

Результат:

01.10.2002 0:00:00

[Копировать в буфер обмена](#)

КОНЕЦПЕРИОДА (ENDOFFPERIOD)

Описание:

Функция предназначена для выделения определенной даты из заданной даты.

Параметры:

- Выражение типа *Дата*.
- Тип периода – строка, содержащая одно из значений: *Минута*, *Час*, *День*, *Неделя*, *Месяц*, *Квартал*, *Год*, *Декада*, *Полугодие*.

Пример:

КОНЕЦПЕРИОДА(ДатаВремя(2002, 10, 12, 10, 15, 34), "Неделя")

[Копировать в буфер обмена](#)

Результат:

13.10.2002 23:59:59

[Копировать в буфер обмена](#)

ДОБАВИТЬКДАТЕ (DATEADD)

Описание:

Функция предназначена для прибавления к дате некоторой величины.

Параметры:

- Выражение типа *Дата*.
- Тип *увеличения* – строка, содержащая одно из значений: *Секунда*, *Минута*, *Час*, *День*, *Неделя*, *Месяц*, *Квартал*, *Год*, *Декада*, *Полугодие*.
- *Величина* – на сколько необходимо увеличить дату. Тип *Число*. Дробная часть игнорируется.

Пример:

ДОБАВИТЬКДАТЕ(ДатаВремя(2002, 10, 12, 10, 15, 34), "Месяц", 1)

[Копировать в буфер обмена](#)

Результат:

РАЗНОСТЬДАТ (DATEDIFF)**Описание:**

Функция предназначена для получения разницы между двумя датами.

Параметры:

- Выражение типа *Дата*.
- Выражение типа *Дата*.
- Тип разности – одно из значений: *Секунда*, *Минута*, *Час*, *День*, *Месяц*, *Квартал*, *Год*.

Пример:

```
РАЗНОСТЬДАТ(ДАТАВРЕМЯ(2002, 10, 12, 10, 15, 34),  
ДАТАВРЕМЯ(2002, 10, 14, 9, 18, 06),  
"ДЕНЬ")
```

[Копировать в буфер обмена](#)**Результат:**

2

[Копировать в буфер обмена](#)**ТЕКУЩАЯДАТА (CURRENTDATE)****Описание:**

Возвращает системную дату. При компоновке макета компоновки во всех выражениях, которые присутствуют в компоновке, функция *ТекущаяДата()* заменяется значением текущей даты.

Пример:

```
ТЕКУЩАЯДАТА()
```

[Копировать в буфер обмена](#)**ПОДСТРОКА (SUBSTRING)****Описание:**

Данная функция предназначена для выделения подстроки из строки.

Параметры:

- выражение, имеющее строковый тип;
- позиция символа, с которого начинается выделяемая из строки подстрока;
- длина выделяемой подстроки.

Пример:

```
ПОДСТРОКА(Контрагенты.Адрес, 1, 4)
```

[Копировать в буфер обмена](#)**ДЛИНАСТРОКИ (STRINGLENGTH)****Описание:**

Функция предназначена для определения длины строки. Параметр – выражение строкового типа.

Пример:

```
ДЛИНАСТРОКИ(Контрагенты.Адрес)
```

[Копировать в буфер обмена](#)**ГОД (YEAR)****Описание:**

Данная функция предназначена для выделения года из значения типа *Дата*. Единственный параметр – это выражение, имеющее тип *Дата*.

Пример:

```
ГОД(РасхНакл.Дата)
```

[Копировать в буфер обмена](#)**КВАРТАЛ (QUARTER)****Описание:**

Данная функция предназначена для выделения номера квартала из значения типа *Дата*. Номер квартала в норме находится в диапазоне от 1 до 4. Единственный параметр функции – это выражение, имеющее тип *Дата*.

Пример:

```
КВАРТАЛ(РасхНакл.Дата)
```

[Копировать в буфер обмена](#)**МЕСЯЦ (MONTH)****Описание:**

Данная функция предназначена для выделения номера месяца из значения типа *Дата*. Номер месяца в норме находится в диапазоне от 1 до 12. Единственный параметр функции – это выражение, имеющее тип *Дата*.

Пример:

```
МЕСЯЦ(РасхНакл.Дата)
```

[Копировать в буфер обмена](#)

ДЕНЬГОДА (DAYOFYEAR)

Описание:

Данная функция предназначена для получения дня года из значения типа `Дата`. День года в норме находится в диапазоне от 1 до 365 (366). Единственный параметр функции – это выражение, имеющее тип `Дата`.

Пример:

`ДЕНЬГОДА(РасхНакл.Дата)`

[Копировать в буфер обмена](#)

ДЕНЬ (DAY)

Описание:

Данная функция предназначена для получения дня месяца из значения типа `Дата`. День месяца в норме находится в диапазоне от 1 до 31. Единственный параметр функции – это выражение, имеющее тип `Дата`.

Пример:

`ДЕНЬ(РасхНакл.Дата)`

[Копировать в буфер обмена](#)

НЕДЕЛЯ (WEEK)

Описание:

Данная функция предназначена для получения номера недели года из значения типа `Дата`. Недели года нумеруются, начиная с 1. Единственный параметр функции – это выражение, имеющее тип `Дата`.

Пример:

`НЕДЕЛЯ(РасхНакл.Дата)`

[Копировать в буфер обмена](#)

ДЕНЬНЕДЕЛИ (WEEKDAY)

Описание:

Данная функция предназначена для получения дня недели из значения типа `Дата`. День недели в норме находится в диапазоне от 1 (понедельник) до 7 (воскресенье). Единственный параметр функции – это выражение, имеющее тип `Дата`.

Пример:

`ДЕНЬНЕДЕЛИ(РасхНакл.Дата)`

[Копировать в буфер обмена](#)

ЧАС (HOUR)

Описание:

Данная функция предназначена для получения часа суток из значения типа `Дата`. Час суток находится в диапазоне от 0 до 23. Единственный параметр функции – это выражение, имеющее тип `Дата`.

Пример:

`ЧАС(РасхНакл.Дата)`

[Копировать в буфер обмена](#)

МИНУТА (MINUTE)

Описание:

Данная функция предназначена для получения минуты часа из значения типа `Дата`. Минута часа находится в диапазоне от 0 до 59. Единственный параметр функции – это выражение, имеющее тип `Дата`.

Пример:

`МИНУТА(РасхНакл.Дата)`

[Копировать в буфер обмена](#)

СЕКУНДА (SECOND)

Описание:

Данная функция предназначена для получения секунды минуты из значения типа `Дата`. Секунда минуты находится в диапазоне от 0 до 59. Единственный параметр функции – это выражение, имеющее тип `Дата`.

Пример:

`СЕКУНДА(РасхНакл.Дата)`

[Копировать в буфер обмена](#)

ВЫРАЗИТЬ (CAST)

Описание:

Данная функция предназначена для выделения типа из выражения, которое может содержать составной тип. Если выражение будет содержать тип, отличный от требуемого, будет возвращено значение `NULL`.

Параметры:

- Преобразуемое выражение.
- Тип – строка, содержащая строку типа. Например, `Число`, `Строка` и т. п. Кроме примитивных типов данная строка может содержать имя таблицы. В таком случае будет осуществлена попытка выразить к ссылке на указанную таблицу.

Пример:

`ВЫРАЗИТЬ(Данные.Реквизит1, "Число(10,3)")`

[Копировать в буфер обмена](#)

ЕСТЬNULL (ISNULL)

Данная функция возвращает значение второго параметра, если значение первого параметра `NULL`. В противном случае будет возвращено значение первого параметра.

Пример:

```
ЕСТЬNULL(Сумма(Продажи.СуммаОборот), 0)
```

[Копировать в буфер обмена](#)

ПРЕДСТАВЛЕНИЕ (PRESENTATION)

Синтаксис:

Представление (<Выражение>)

Описание:

Данная функция возвращает строковое представление переданного значения непримитивного типа. Для значений примитивного типа возвращает само значение.

Если в качестве параметра передан массив, то функция возвращает строку, содержащую строковые представления всех элементов массива, разделенных символами "; ". Если в качестве параметра передана таблица значений, то функция возвращает строку, содержащую строковые представления всех строк таблицы значений, причем представления ячеек каждой строки разделяются символами "; ", а строки – символом перевода строки. Если у какого-либо элемента строковое представление пустое, то вместо его представления выводится строка <Пустое значение>.

Пример:

```
Представление(Контрагент)
```

[Копировать в буфер обмена](#)

СТРОКА (STRING)

Синтаксис:

Строка (<Выражение>)

Описание:

Если в качестве параметра передан массив, то функция возвращает строку, содержащую строковые представления всех элементов массива, разделенных символами "; ". Если в качестве параметра передана таблица значений, то функция возвращает строку, содержащую строковые представления всех строк таблицы значений, причем представления ячеек каждой строки разделяются символами "; ", а строки – символом перевода строки. Если у какого-либо элемента строковое представление пустое, то вместо его представления выводится строка <Пустое значение>.

Пример:

```
Строка(ДатаПродажи)
```

[Копировать в буфер обмена](#)

ACOS

Синтаксис:

ACos(<X>)

Описание:

Вычисляет значение арккосинуса.

ASIN

Синтаксис:

ASin(<X>)

Описание:

Вычисляет значение арксинуса.

ATAN

Синтаксис:

ATan(<X>)

Описание:

Вычисляет значение арктангенса.

COS

Синтаксис:

Cos(<X>)

Описание:

Вычисляет значение косинуса.

SIN

Синтаксис:

Sin(<X>)

Описание:

Вычисляет значение синуса.

TAN

Синтаксис:

Tan(<X>)

Описание:

Вычисляет значение тангенса.

EXP

Синтаксис:

`Exp(<X>)`

Описание:

Вычисляет e в степени X .

LOG

Синтаксис:

`Log(<X>)`

Описание:

Вычисляет значение натурального логарифма.

LOG10

Синтаксис:

`Log10(<X>)`

Описание:

Вычисляет значение десятичного логарифма.

POW

Синтаксис:

`Pow(<X>, <Y>)`

Описание:

Вычисляет значение X в степени Y .

SQRT

Синтаксис:

`Sqrt(<X>)`

Описание:

Вычисляет значение квадратного корня.

OKP (ROUND)

Синтаксис:

`Окр(<Выражение>, <КоличествоЗнаков>)`

Описание:

Округляет значение `Выражение` до `КоличествоЗнаков` после запятой.

ЦЕЛ (INT)

Синтаксис:

`Цел(<Выражение>)`

Описание:

Вычисляет целую часть переданного числа, полностью отсекая дробную часть.

ТИПЗНАЧЕНИЯ (VALUETYPE)

Синтаксис:

`ТипЗначения(<Выражение>)`

Описание:

Вычисляет тип выражения, переданного в качестве параметра.

Возвращаемое значение:

Значение типа `Тип`.

10.3.4.12. Функции общих модулей

Выражение механизма компоновки данных может содержать вызовы функций глобальных общих модулей конфигурации и неглобальных общих модулей с установленным свойством `Клиент (обычное приложение)` (при использовании обычного приложения) или `Сервер` (при использовании управляемого приложения). Никакого дополнительного синтаксиса для вызова таких функций не требуется.

[Копировать в буфер обмена](#)

`СокращенноеНаименование` (`Докум.Ссылка`, `Докум.Дата`, `Докум.Номер`)

В данном примере будет осуществлен вызов функции `СокращенноеНаименование()` из общего модуля конфигурации.

Отметим, что использование функций общих модулей разрешено только при указании соответствующего параметра процессора компоновки данных.

Кроме того, функции общих модулей не могут быть использованы в выражениях пользовательских полей.

10.3.5. Конструктор схемы компоновки данных

Конструктор схемы компоновки данных представляет собой объект встроенного языка `КонструкторСхемыКомпоновкиДанных`, предназначенный для визуального конструирования схемы компоновки данных. Кроме того, конструктор схемы компоновки данных используется в конфигураторе при редактировании схемы компоновки данных.

Для наборов данных – запроса конструктор автоматически получает вложенные наборы данных из текста запроса и оформляет их как поля набора данных – вложенный набор данных.

Для наборов данных – объекта конструктор позволяет добавлять поля – вложенные наборы данных.

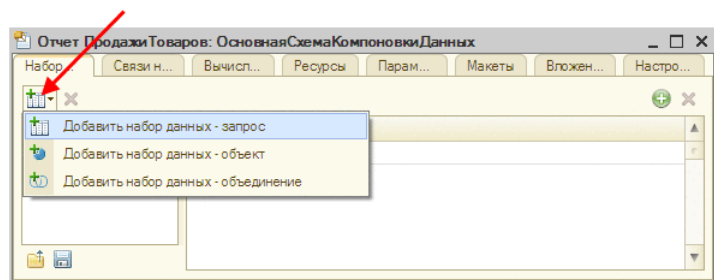


Рис. 262. Добавление нового набора данных

Ниже приведен пример открытия окна конструктора схемы компоновки данных и последующей сериализации полученной схемы компоновки в XML.

[Копировать в буфер обмена](#)

```
Процедура КоманднаяПанельРедактораОтчета (Кнопка)
    Конструктор = Новый КонструкторСхемыКомпоновкиДанных;
    Конструктор.УстановитьСхему(ПолучитьСхемуКомпоновкиДанных());
    Конструктор.Редактировать(ЭтотОбъект);
КонецПроцедуры
Процедура ОбработкаВыбора(ЗначениеВыбора, Источник)
    Если ТипЗнч(Источник) = Тип("КонструкторСхемыКомпоновкиДанных") Тогда
        СхемаКомпоновкиДанных = Источник.ПолучитьСхему();
        ЗаписьXML = Новый ЗаписьXML;
        ЗаписьXML.УстановитьСтроку();
        СериализаторXDTO.ЗаписатьXML(ЗаписьXML, СхемаКомпоновкиДанных, "dataCompositionScheme", "http://v8.1c.ru/8/data-composition-system/scheme");
        ЭлементыФормы.ТекстСхемыКомпоновкиДанных.УстановитьТекст(ЗаписьXML.Закреть());
    КонецЕсли;
КонецПроцедуры
```

Работа со схемой компоновки данных подразделяется на следующие этапы:

- редактирование наборов данных,
- редактирование полей наборов данных,
- редактирование связей наборов данных,
- редактирование вычисляемых полей,
- редактирование ресурсов,
- редактирование параметров,
- редактирование макетов,
- редактирование вложенных настроек,
- редактирование настроек системы компоновки данных.

10.3.5.1. Редактирование наборов данных

Система компоновки данных поддерживает редактирование следующих объектов:

- набор данных – запрос;
- набор данных – объект;
- набор данных – объединение;
- полей запроса.

При добавлении набора данных ему автоматически генерируется имя и источник данных (если источника данных не существует).

Поле набора данных схемы компоновки данных – вложенный набор данных может быть описано в запросе и описано в наборе данных – объекте.

В тексте запроса набора данных – объекта в предложениях `{ВЫБРАТЬ}` и `{ГДЕ}` можно использовать вложенные таблицы.

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    ПриходТовара.Дата,
    ПриходТовара.Номер,
    ПриходТовара.Поставщик,
    ПриходТовара.Склад,
    ПриходТовара.Товары.(
        НомерСтроки,
        Товар,
        Цена,
        Количество,
        Сумма
    )
{ВЫБРАТЬ
    Дата,
    Номер,
    Поставщик.*,
    Склад.*,
    Товары.(
        НомерСтроки,
        Товар.*,
        Цена,
        Количество,
        Сумма
    )
}
```



```

Документ.ПриходТовара КАК ПриходТовара
{ГДЕ
ПриходТовара.Дата,
ПриходТовара.Номер,
ПриходТовара.Поставщик.*,
ПриходТовара.Склад.*,
ПриходТовара.Товары.(
НомерСтроки,
Товар.*,
Цена,
Количество,
Сумма
)}

```

В приведенном примере жирным выделены фрагменты запроса, описывающие поля вложенных таблиц, доступные для настроек.

10.3.5.1.1. Редактирование набора данных – запроса

Редактирование набора данных – запроса заключается в составлении запроса к данным. Для этого можно воспользоваться конструктором запроса или редактировать текст запроса непосредственно в окне конструктора схемы компоновки данных.

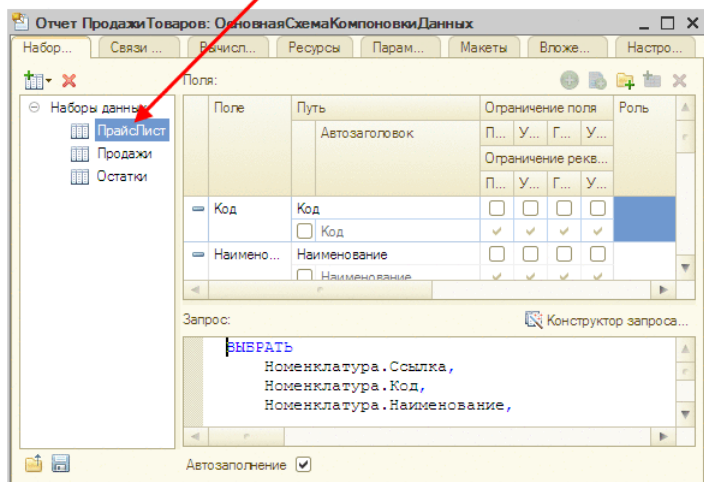


Рис. 263. Набор данных – запрос

Если флажок **Автозаполнение** установлен, то система компоновки данных автоматически заполняет поля схемы компоновки данных на основании созданного запроса.

10.3.5.1.2. Редактирование набора данных – объекта

Редактирование набора данных – объекта заключается в добавлении/редактировании имени объекта, содержащего данные; в добавлении/редактировании полей и группировок.

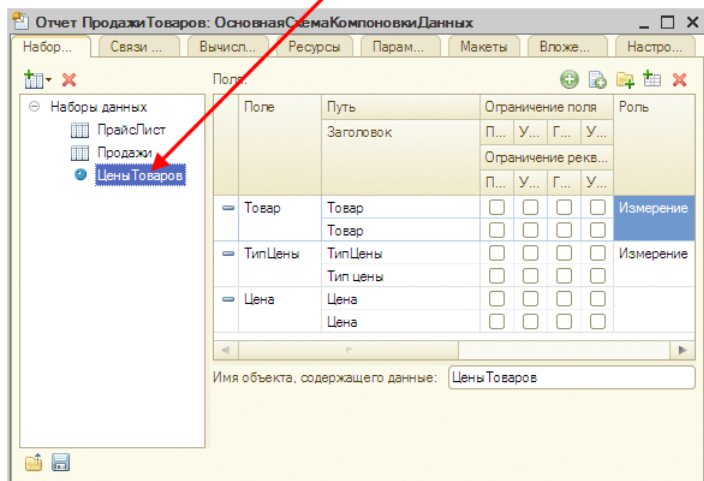


Рис. 264. Набор данных – объект

10.3.5.1.3. Редактирование набора данных – объединения

Редактирование набора данных – объединения заключается в редактировании объединенных полей из состава полей, подчиненных данному набору данных. Созданные наборы данных – запросы и объекты можно добавлять в набор данных – объединение, перемещая их мышкой.

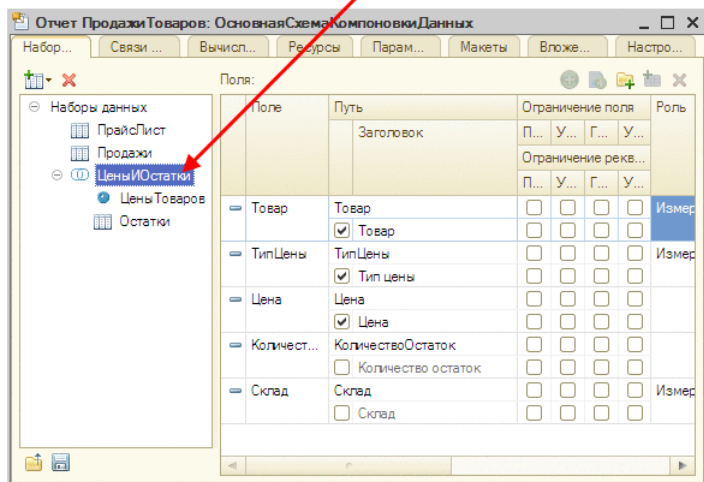


Рис. 265. Набор данных – объединение

10.3.5.2. Редактирование полей набора данных

При редактировании полей существует возможность задать:

- заголовок поля;
- ограничение доступности поля;
- ограничение доступности полей-реквизитов;
- роль для поля;
- представление поля;
- выражения упорядочивания;
- способ проверки иерархии (набор данных и параметр);
- тип значения поля;
- оформление поля.

Следует заметить, что роль поля, представление, выражение упорядочивания, параметр, тип значения и оформление поля редактируются только на верхнем уровне иерархии наборов данных.

Если у поля набора данных не установлен заголовок, система компоновки данных пытается сформировать заголовок поля на основе синонима поля запроса. Если синоним для поля не получен, то в качестве заголовка поля используется путь к данным поля, дополненный пробелами при помощи алгоритма преобразования имен в синонимы. Синонимы для полей запроса получаются только для тех полей, у которых либо не установлен псевдоним, либо псевдоним не отличается от псевдонима, которой был бы у данного поля по умолчанию.

Синонимы для поля запроса получаются следующим образом.

- Если запрос не содержит объединений:
 - Если выражение поля запроса состоит из одного поля, то синоним получается из этого поля.
 - Если выражение поля является агрегатной функцией над одним полем, то синоним поля получается из этого поля. Для других выражений считается, что синоним получить невозможно.
- Если запрос содержит объединения:
 - Во всех объединениях ищется поле, для которого можно получить синоним, использующийся в качестве синонима поля. Если такого поля не найдено, то считается, что синоним получить невозможно.

В конструкторе схемы компоновки данных существует дополнительная колонка с флажком, указывающим, что заголовок поля установлен вручную. Флажок автоматически снимается, если у поля не установлен заголовок.

При заполнении полей на основании запроса конструктор автоматически заполняет заголовки для тех полей, у которых невозможно получение синонима, и для тех полей, у которых в запросе псевдоним отличается от псевдонима данного поля по умолчанию. Таким образом, для таких полей флажок будет автоматически установлен.

Если данный флажок снят, в колонке **Заголовок** отображается заголовок, который будет выводиться пользователю. При этом сам заголовок в поле не заполнен. Колонка **Заголовок** для полей, у которых снят флажок, не редактируется и отображается недоступным цветом текста.

Если разработчику требуется изменить заголовок, он устанавливает флажок, при этом текст заголовка поля заполняется на основании текста, который ранее отображался в колонке **Заголовок**. Если флажок снят, конструктор очищает текст заголовка и начинает отображать в заголовке автоматически генерируемый заголовок.

Роль поля изначально определяется в запросе, но есть возможность ее изменения в отдельном диалоге.

Рис. 266. Роль поля

Для поля счета в параметре **Вид** необходимо явно указать ссылку на вид счета.

Для поля измерения в параметре **Измерение** можно указать путь к данным родительского измерения.

Установленный признак **Игнорировать значения NULL** означает, что в результат не будут включены групповые записи по данному полю, если оно содержит значение **NULL**.

В результирующем наборе данных поля с установленным флажком **Обязательное** будут присутствовать всегда, если в настройках задействовано хотя бы одно поле из его набора данных. Например, необходимо получать развернутые остатки по субконто. При этом если поле **Субконто** не будет использоваться в запросе, остатки будут получаться свернутыми.

Выражения упорядочивания поля, тип значения и оформление также могут быть отредактированы в отдельном диалоге.

10.3.5.3. Редактирование связей наборов данных

При наличии нескольких наборов данных верхнего уровня существует возможность настроить связь между ними по одному или нескольким полям.

Источник связи	Приемник свя...	Выражение и...	Выражение п...	Параметр	Список парам...	Усрд
ПрайсЛист	Продажи	Ссылка	Товар	Номенклатура	<input checked="" type="checkbox"/>	
ПрайсЛист	Остатки	Ссылка	Товар	Номенклатура	<input checked="" type="checkbox"/>	

Рис. 267. Связи наборов данных

Источником и приемником связи являются наборы данных. Выражения источника и приемника – поля наборов данных.

10.3.5.4. Редактирование вычисляемых полей

Закладка **Вычисляемые поля** позволяет создавать и редактировать следующие свойства/характеристики вычисляемых полей:

- путь к данным;
- выражение;
- заголовков;
- ограничение доступности;
- выражение представления;
- выражения упорядочивания;
- тип значения;
- оформление;
- доступные значения.

Выражения упорядочивания редактируются в отдельном диалоге.

10.3.5.5. Редактирование ресурсов

Вычисление ресурсов возможно по всем полям всех наборов данных и по вычисляемым полям. В левом табличном поле отображается список доступных и неиспользованных полей. В правом табличном поле отображаются поля, по которым будут формироваться итоги и выражения их вычисления.

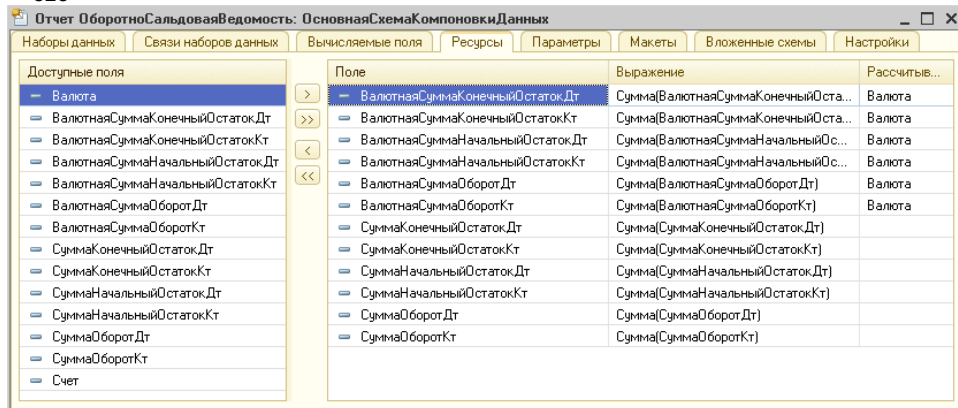


Рис. 268. Редактирование ресурсов

По умолчанию для числовых полей устанавливается функция *Сумма*, для нечисловых полей – *Количество*. Нажатием кнопки ">>" можно добавить в ресурсы все поля типа *Число*. При этом допускается ввод нескольких строк для одного ресурса. Компоновщик макета, получая выражение для ресурса, использует информацию о том, для какой группировки оно получается, и выдает соответствующее выражение.

В диалоге редактирования колонки *Рассчитывать по...* для полей, по которым возможна иерархическая группировка, добавляются строки с именем поля и ключевым словом *Иерархия*. Выражение будет использоваться для иерархических записей группировки по полю, указанному перед словом. При выборе группировки, в разрезе которой может рассчитываться ресурс, можно одновременно выбирать и обычное, и иерархическое поле.

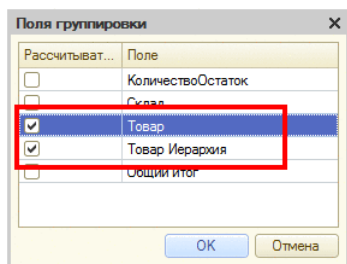


Рис. 269. Расчет ресурсов по полям группировки

Если для ресурса было указано, что его можно рассчитывать только в разрезе некоторой группировки (то есть в колонке *Рассчитывать по...* было выбрано хотя бы одно поле группировки), то данный ресурс будет выводиться в результат только для этой группировки и группировок, вложенных в нее.

10.3.5.6. Редактирование параметров

Редактирование параметров включает в себя:

- редактирование имени параметра;
- редактирование заголовка;
- редактирование доступных типов и значений параметра;
- определение значения и доступности списка значений параметра;
- определение выражения;
- определение параметра в качестве доступного поля настройки компоновки данных;
- ограничение доступности;
- определение обязательности заполнения параметра;
- признак использования параметра;
- задание параметров редактирования.

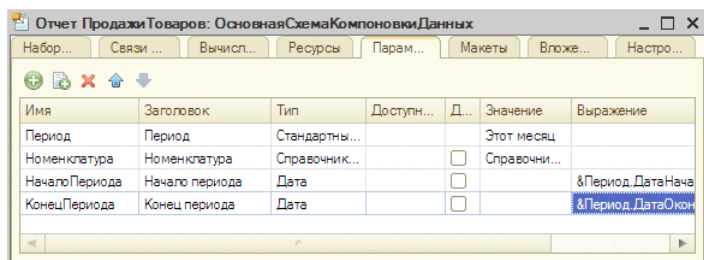


Рис. 270. Редактирование параметров

При неопределенном значении параметра считается, что значение – нулевая ссылка на заданный тип. В параметрах могут использоваться предопределенные данные и перечисления (режим Конфигуратор).

В колонке *Доступные значения* редактируются значения, которые могут быть выбраны пользователем в качестве параметров схемы компоновки данных. Если флажок в колонке *Доступен список значений* установлен, то это означает, что можно будет использовать несколько значений параметра.

ПРИМЕЧАНИЕ. При использовании типа параметра *СтандартныйПериод* следует учитывать, что даты начала и конца стандартного периода также содержат и время. Причем начальная дата имеет время **00:00:00**, а конечная дата **23:59:59**. Таким образом, в запросе не обязательно использовать функции *НАЧАЛОПЕРИОДА* и *КОНЕЦПЕРИОДА*.

Приведем пример использования обязательного параметра схемы компоновки данных: допустим, в отчете требуется обязательно указывать параметр *Организация*. В схеме компоновки данных для параметра *Организация* свойство *Использование* устанавливается в значение *Всегда*, а свойство *Запрещать незаполненные значения* – в значение *Истина*.

Во время работы с отчетом пользователю недоступен флажок *Использование* (он отсутствует), а незаполненное значение отображается обычным способом (волнистая красная

черта). Если пользователь не заполнит значение параметра и запустит отчет на исполнение, он получит сообщение о том, что нужно ввести значение параметра.

10.3.5.7. Редактирование макетов

Добавление макетов осуществляется нажатием кнопки **Добавить макет** на командной панели.

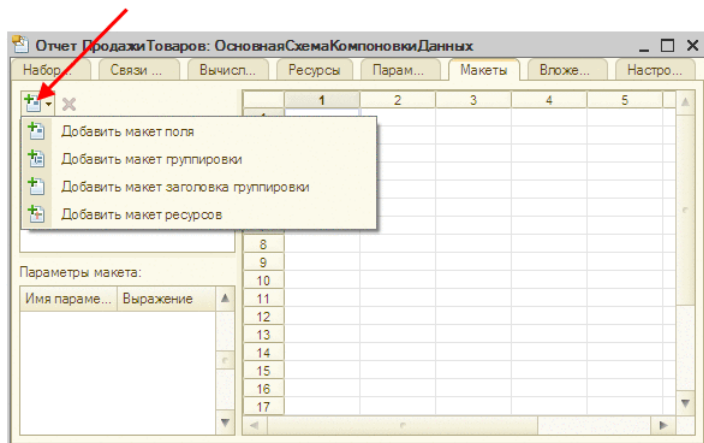


Рис. 271. Редактор макета

Предлагается выбрать один из вариантов макетов:

- макет поля,
- макет группировки,
- макет заголовка группировки,
- макет ресурсов таблицы.

Для макета группировки и заголовка группировки можно задать имя группировки или поля группировки, тип макета.

Для макета ресурса таблицы можно задать макеты для двух группировок, на пересечении которых он находится.

Колонка табличного поля **Область** указывает координаты области макета в табличном документе.

Редактирование областей табличного документа возможно с помощью панели свойств, вызываемой **Alt + Enter**. Редактируется как внешний вид, так и содержимое ячейки и параметр расшифровки для нее.

Макет может редактироваться на всех языках, поддерживаемых системой. При задании параметра или шаблона ячейке табличного документа параметры добавляются в макет и отображаются в колонке **Имя параметра** табличного поля **Параметры макета**. Возможно редактирование выражения параметра макета. При загрузке области макета разделяются пустыми строками.

10.3.5.8. Вложенные схемы

Закладка **Вложенные схемы** позволяет создавать и редактировать вложенные схемы компоновки данных. В роли схемы могут выступать вложенные схемы, редактируемые конструктором схемы компоновки данных.

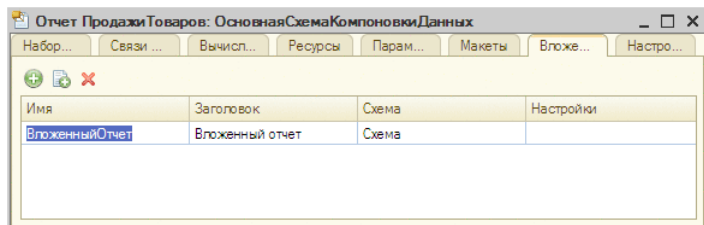


Рис. 272. Вложенные схемы

10.3.5.9. Настройки

Схема компоновки данных содержит настройки компоновки данных по умолчанию, которые могут быть заданы разработчиком.

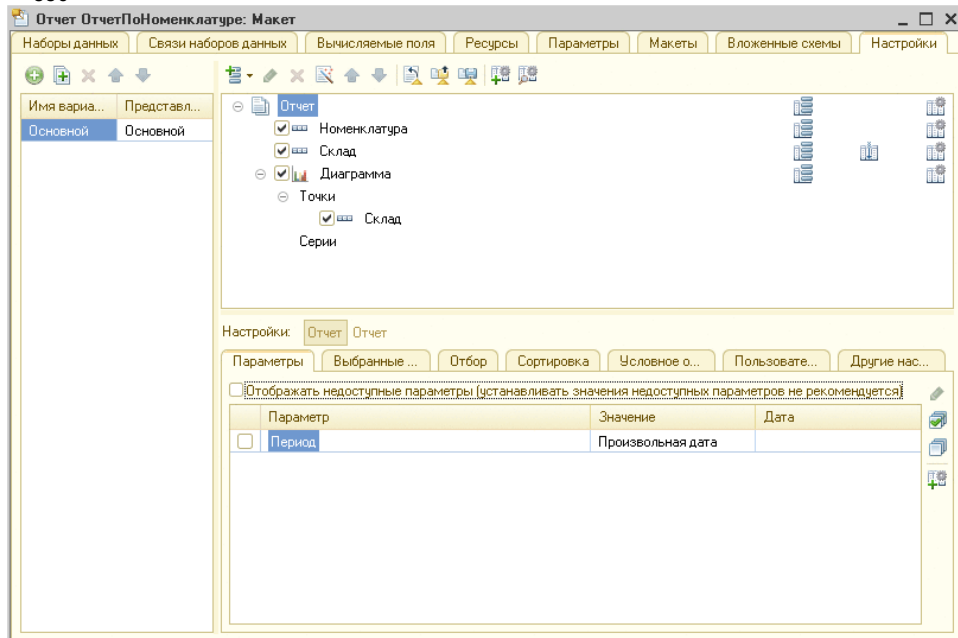


Рис. 273. Редактор настроек схемы компоновки данных

Настройка отчета **Авто позиция ресурсов** управляет тем, как будут выводиться поля ресурсов:

- **Не использовать** – в этом случае ресурсы выводятся в том порядке, в котором перечислены поля на закладке **Выбранные поля**.
- **После всех полей** – в этом случае ресурсы выводятся после всех полей.

Для группировки (или группировки таблицы) существует параметр **Вариант использования группировки**, который управляет выводом дополнительной информации после полей ресурсов. Если данный параметр не установлен либо имеет значение **Авто**, то группировка отвечает за вывод детальных записей.

Если параметр имеет значение **Дополнительная информация** и указан у группировки-колонки, то в результирующую таблицу в этой колонке будет выдана одна колонка, в ней будут выводиться поля, указанные на закладке **Выбранные поля** данной группировки и доступные для вывода в группировке-строке. Если при этом у группировки-колонки указаны поля группировки, то при компоновке макета будет выдана ошибка. Для группировки-строки – аналогично, только выводится одна строка, а не колонка.

Например, в таблицу в строках выдается **Номенклатура**. В колонках таблицы выдаются склады. Формируется отчет об остатках товаров на складах. Необходимо вывести артикул номенклатуры после колонок с остатками. Для этого следует добавить в колонки таблицы группировку без групповых полей и в настройках группировки указать параметр **Вариант использования группировки** в значение **Дополнительная информация**. На закладке **Выбранные поля** следует указать поле **Номенклатура.Артикул**.

При выводе такой группировки вне таблицы в группировке выдаются все поля, которые доступны в данной группировке.

Например, если в отчете имеется группировка, в которую вложено несколько группировок. При помощи группировки с параметром **Вариант использования группировки**, установленным в значение **Дополнительная информация**, можно вывести между группировками данные родительской группировки.

Для группировок с вариантом использования **Дополнительная информация** игнорируются следующие параметры:

- **Расположение полей группировки** – заголовки полей, выводимых в группировки, всегда выдаются в начале строки/колонки.
- **Количество записей**.
- **Процент записей**.
- **Расположение итогов**.
- **Расположение полей группировок**.
- **Расположение группировок**.
- **Расположение общих итогов**.

При выполнении компоновки макета, если в группировке с вариантом использования **Дополнительная информация** используется поле, которое недоступно в группировке или в таблице, недоступно ни в одной группировке, выдается исключение.

10.3.5.9.1. Вложенные поля

Для числовых полей

Для числовых ресурсов система автоматически формирует вложенные поля. Эти поля автоматически рассчитываются системой и позволяют упростить получение различных показателей, например, для получения процентного отношения значения ресурса к сумме значений ресурса во всех данных отчета. Перечислим эти поля.

§ в группе иерархии

Идентификатор:

ПроцентВиерархии

PercentInHierarchy

Описание:

Данное поле содержит процент значения ресурса в текущей иерархической группе. При выводе вне таблицы равно **100 %**.

§ в группе иерархии колонки или точки

Идентификатор:

ПроцентВиерархииВКолонкеИлиТочке

PercentInColumnOrPointHierarchy

Описание:

Данное поле содержит отношение значения ресурса в текущей ячейке к значению итога ресурса на текущем уровне иерархии текущей группировки по колонке или точке (в процентах). При выводе вне таблицы равно полю **% в группе иерархии**.

% в группе иерархии строки или серии

Идентификатор:

ПроцентВИерархииВСтрокеИлиСерии

PercentInRowOrSeriesHierarchy

Описание:

Данное поле содержит отношение значения ресурса в текущей ячейке к значению итога ресурса на текущем уровне иерархии текущей группировки по строке или серии (в процентах). При выводе вне таблицы равно **100 %**.

% в группировке

Идентификатор:

ПроцентВГруппе

GroupPercent

Описание:

Данное поле содержит отношение значения ресурса в текущей ячейке к значению итога ресурса в текущей группировке. При выводе вне таблицы содержит **100 %**.

% в группировке колонки или точки

Идентификатор:

ПроцентВГруппеВКолонкеИлиТочке

ColumnOrPointGroupPercent

Описание:

Данное поле содержит отношение значения ресурса в текущей ячейке к значению итога ресурса в текущей группировке по колонке или точке (в процентах). При выводе вне таблицы содержит **100 %**.

% в группировке строки или серии

Идентификатор:

ПроцентВГруппеВСтрокеИлиСерии

RowOrSeriesGroupPercent

Описание:

Данное поле содержит отношение значения ресурса в текущей ячейке к значению итога ресурса в текущей группировке по строке или серии (в процентах). При выводе вне таблицы содержит поле **% в группировке**.

% в колонке или точке

Идентификатор:

ПроцентВКолонкеИлиТочке

ColumnOrPointPercent

Описание:

Данное поле содержит отношение значения ресурса в текущей ячейке к значению итога ресурса по колонке или точке (в процентах). При выводе вне таблицы содержит поле **% общий**.

% в строке или серии

Идентификатор:

ПроцентВСтрокеИлиСерии

RowOrSeriesPercent

Описание:

Данное поле содержит отношение значения ресурса в текущей ячейке к значению итога ресурса по строке или серии (в процентах). При выводе вне таблицы содержит **100 %**.

% общий

Идентификатор:

ПроцентОбщий

OverallPercent

Описание:

Данное поле содержит отношение значения ресурса в текущей ячейке к значению общего итога ресурса в таблице (в процентах). При выводе вне таблицы содержит **100%**.

Для полей, содержащих в составе своих типов тип Дата

Для полей, содержащих в составе своих типов тип *Дата*, система автоматически формирует вложенные поля. Эти поля автоматически рассчитываются системой и позволяют упростить отображение различной информации, связанной с датой и временем. Для удобства работы, эти поля сгруппированы в три группы: **Даты начала** (*ДатыНачала*, *BeginDates*), **Даты конца** (*ДатыКонца*, *EndDates*), **Части дат** (*ЧастиДат*, *DateParts*). Перечислим эти поля.

Группа «Даты начала»

Начало дня

Идентификатор:

НачалоДня

Описание:

Возвращает дату и время начала дня.

Начало недели

Идентификатор:

НачалоНедели

BegOfWeek

Описание:

Возвращает дату и время начала недели.

Начало декады

Идентификатор:

НачалоДекады

BegOfTenDays

Описание:

Возвращает дату и время начала декады.

Начало месяца

Идентификатор:

НачалоМесяца

BegOfMonth

Описание:

Возвращает дату и время начала месяца.

Начало квартала

Идентификатор:

НачалоКвартала

BegOfQuarter

Описание:

Возвращает дату и время начала квартала.

Начало полугодия

Идентификатор:

НачалоПолугодия

BegOfHalfYear

Описание:

Возвращает дату и время начала полугодия.

Начало года

Идентификатор:

НачалоГода

BegOfYear

Описание:

Возвращает дату и время начала полугодия.

Начало минуты

Идентификатор:

НачалоМинуты

BegOfMinute

Описание:

Возвращает дату и время начала минуты (если в составе даты есть время).

Начало часа

Идентификатор:

НачалоЧаса

BegOfHour

Описание:

Возвращает дату и время начала часа (если в составе даты есть время).

Группа «Даты конца»

Конец дня

Идентификатор:

КонецДня

EndOfDay

Описание:

Возвращает дату и время конца дня.

Конец недели

Идентификатор:

КонецНедели

EndOfWeek

Описание:

Возвращает дату и время конца недели.

Конец декады

Идентификатор:

КонецДекады

EndOfTenDays

Описание:

Возвращает дату и время конца декады.

Конец месяца

Идентификатор:

КонецМесяца

EndOfMonth

Описание:

Возвращает дату и время конца месяца.

Конец квартала

Идентификатор:

КонецКвартала

EndOfQuarter

Описание:

Возвращает дату и время конца квартала.

Конец полугодия

Идентификатор:

КонецПолугодия

EndOfHalfYear

Описание:

Возвращает дату и время конца полугодия.

Конец года

Идентификатор:

КонецГода

EndOfYear

Описание:

Возвращает дату и время конца года.

Конец минуты

Идентификатор:

КонецМинуты

EndOfMinute

Описание:

Возвращает дату и время конца минуты (если в составе даты есть время).

Конец часа

Идентификатор:

КонецЧаса

EndOfHour

Описание:

Возвращает дату и время конца часа (если в составе даты есть время).

Группа «Части дат»

День

Идентификатор:

День

Day

Описание:

День месяца (число).

День недели

Идентификатор:

ДеньНедели

WeekDay

Описание:

Номер дня недели (как возвращает функция `ДеньНедели()`, неделя всегда начинается с понедельника).

Название дня недели

Идентификатор:

НазваниеДняНедели

WeekDayName

Описание:

Представление дня недели (1 – понедельник, 2 – вторник и т. д.). Номер дня получается функцией `ДеньНедели()`.

День года

Идентификатор:

ДеньГода

DayOfYear

Описание:

Номер для дня года (число).

Неделя года

Идентификатор:

НеделяГода

WeekOfYear

Описание:

Номер недели в году (число).

Месяц

Идентификатор:

Месяц

Month

Описание:

Номер месяца года (число).

Название месяца

Идентификатор:

НазваниеМесяца

MonthName

Описание:

Представление названия месяца (1 – январь, 2 – февраль и т. д.).

Квартал

Идентификатор:

Квартал

Quarter

Описание:

Номер квартала.

Год

Идентификатор:

Год

Year

Описание:

Значение года в 4-разрядном виде (число).

Минута

Идентификатор:

Минута

Minute

Описание:

Номер минуты.

Час

Идентификатор:

Час

Hour

Описание:

Номер часа.

10.3.5.9.2. Системные поля

В списке выбранных полей система формирует специальную группу системных полей (**СистемныеПоля**, **SystemFields**), которые предназначены для определения порядкового номера записи, как во всем отчете, так и в рамках группировок. Следует помнить, что системные поля не попадают в список полей, получающихся при развороте автополя, поэтому их необходимо добавлять вручную.

№ п/п

Идентификатор:

СистемныеПоля.НомерПоПорядку

SystemFields.SerialNumber

Описание:

Содержит порядковый номер строки в данном отчете. Начинается с 1.

№ в группе

Идентификатор:

СистемныеПоля.НомерПоПорядкуВГруппировке

SystemFields.GroupSerialNumber

Описание:

Содержит порядковый номер строки в текущей группировке. Начинается с 1.

Уровень

Идентификатор:

СистемныеПоля.Уровень

SystemFields.Level

Описание:

Содержит текущий уровень записи. Начинается с 1.

Уровень в группе

Идентификатор:

СистемныеПоля.УровеньВГруппировке

SystemFields.LevelInGroup

Описание:

Содержит текущий уровень записи относительно группировки. Начинается с 1.

10.3.5.9.3. Поля параметров

В выбранных полях присутствует специальная группа **Параметры**, которая предоставляет возможность поместить в отчет параметры схемы компоновки данных, у которых установлен флажок **Включать в доступные поля**.

10.3.6. Настройки варианта компоновки данных

В схеме компоновки данных имеется возможность определения нескольких вариантов настроек. **Вариант настройки** – это набор настроек отчета, которые разработчик посчитал нужным выделить отдельно. Такие варианты настроек хранятся в схеме компоновки данных.

Например, для отчета **Динамика продаж** одним вариантом отчета может служить диаграмма, показывающая продажи товаров по периодам, а другим – табличный отчет, показывающий продажи товаров в разрезе покупателей. При этом каждый вариант отчета обладает своим набором пользовательских настроек.

При использовании схемы компоновки данных для отчета варианты настроек, описанные в схеме, предоставляются пользователю как стандартные варианты отчета.

Система предоставляет возможность создания нового варианта отчета непосредственно в режиме 1С:Предприятие. Такое действие рекомендуется выполнять опытным пользователям. В этом случае новый вариант отчета сохраняется в хранилище вариантов отчетов, из которого другие пользователи могут загрузить необходимый вариант. Описание хранилища вариантов отчетов см. [здесь](#).

При помощи механизма сравнения и объединения конфигураций (см. [здесь](#)) части настроек можно сравнивать и объединять.

Загрузку схемы компоновки данных из XML можно осуществить стандартными средствами встроенного языка.

10.3.6.1. Структура варианта настроек компоновки данных

Структура – это некоторый скелет настроек. Она определяет взаимное расположение их основных элементов.

Структура настроек доступна через свойство `Структура` объекта `НастройкиКомпоновкиДанных`. Элементами структуры настроек могут быть:

- группировки;
- таблицы (`ТаблицаКомпоновкиДанных`);
- диаграммы (`ДиаграммаКомпоновкиДанных`);
- вложенные объекты настройки (`НастройкиВложенногоОбъектаСистемыКомпоновкиДанных`).

Группировка

Для реализации группировки в структуре настроек предусмотрено три разных типа данных:

- группировки (`ГруппировкаКомпоновкиДанных`);
- группировки таблиц (`ГруппировкаТаблицыКомпоновкиДанных`);
- группировки диаграмм (`ГруппировкаДиаграммыКомпоновкиДанных`).

Наличие трех типов связано с необходимостью реализовать ограничения, наложенные на взаимное расположение элементов в дереве структуры: таблицы и диаграммы не могут включать в себя ничего, кроме группировок. Соответственно, все объекты группировок имеют идентичную объектную модель, они различаются типом вложенной коллекции значений и составом параметров вывода.

Поля группировки

Набор полей, по которым осуществляется группировка, описывается с помощью объекта `ПоляГруппировкиСистемыКомпоновкиДанных`. В свойстве `Элементы` этого объекта содержится коллекция полей группировки, состоящая из объектов `ПолеГруппировкиСистемыКомпоновкиДанных`.

ПРИМЕЧАНИЕ. При выполнении группировки по полю-периоду в группировку автоматически добавляется родительское поле-период, не являющееся дополнительным периодом, в том случае, если в родительских группировках не осуществлялась группировка по этому родительскому полю-периоду.

Например, если группировка осуществляется по полю `Регистратор`, то в группировку автоматически будет добавлено поле `ПериодСекунда`.

При этом создание группировок по реквизитам полей-периодов запрещено.

Автополе группировки

Перед использованием автополе будет преобразовано в набор полей группировки.

Формирование набора происходит следующим образом. Берутся используемые выбранные поля со следующими условиями:

- они доступны для использования в полях группировки;
- не являются ресурсами;
- не зависят от других выбранных полей;
- не зависят от уже существующих полей группировки.

Если поле уже включено в данные поля группировки, повторно оно не добавляется.

Таблица

Описание таблицы в структуре настроек выполняется с помощью объекта `ТаблицаСистемыКомпоновкиДанных`.

Диаграмма

Описание диаграммы в структуре настроек выполняется с помощью объекта `ДиаграммаСистемыКомпоновкиДанных`.

Вложенный объект

Описание вложенного объекта в структуре настроек выполняется с помощью объекта `НастройкиВложенногоОбъектаСистемыКомпоновкиДанных`.

Для объекта реализовано свойство `Имя`, предназначенное для идентификации вложенного отчета в сгенерированном макете компоновки данных.

10.3.6.2. Свойства настроек компоновки данных

Выбор

Набор полей, выводимых в результат компоновки. Описывается с помощью объекта `ВыбранныеПоляКомпоновкиДанных`. В свойстве `Элементы` этого объекта содержится коллекция выбранных полей, состоящая из объектов `ВыбранноеПолеКомпоновкиДанных`.

Группа выбранных полей

Используется для группировки полей. Описывается с помощью объекта `ГруппаВыбранныхПолейКомпоновкиДанных`.

Автовыбранное поле

Перед использованием автополе будет преобразовано в набор выбранных полей. Состав набора полей зависит от того, какому элементу структуры принадлежит разворачиваемое автополе и в какой части структуры этот элемент располагается. Для каждого элемента система обходит все родительские элементы структуры отчета и из выбранных полей этих элементов отбирает ресурсы и поля по следующим правилам:

- Для группировки и группировки таблицы на место автополя подставляются:
 - все используемые поля этой группировки, которые доступны для использования в выбранных полях;
 - поля, которые являются реквизитами ее полей группировки,

- ресурсы родительских элементов.

ВНИМАНИЕ! Система при обходе учитывает только те группировки, тип которых [Без иерархии](#) или [Иерархия](#).

- Для группировки диаграммы ресурсы не выбираются, а обходятся все родительские элементы структуры настроек, и из выбранных полей этих элементов выбираются поля группировок, если по данному полю была задана группировка типа [Только иерархия](#).
- Для группировок типа [Детальные записи](#) (группировка, группировка таблицы, группировка диаграммы) из основных выбранных полей настроек, которым принадлежит группировка, выбираются все используемые поля, кроме полей, участвовавших в вышестоящих группировках, и реквизитов этих полей. Если же такая группировка имеет тип [Только иерархия](#), то ее поля и реквизиты будут использоваться системой при формировании набора полей выбора. Для группировки диаграммы ресурсы также не выбираются.
- Для группировок типа [Детальная информация](#) в выбранные поля помещаются поля – не ресурсы, которые доступны в группировке или родительской группировке, а в таблице – еще и в какой-либо группировке с противоположной оси.
- Для диаграммы автополе выбора заменяется всеми вышестоящими ресурсами.
- Для таблицы автополе выбора преобразовывается в набор используемых родительскими элементами ресурсов.

ПРИМЕЧАНИЕ. Если поле уже включено в выбранные поля, повторно оно не добавляется.

При этом поля добавляются в набор в следующем порядке: вначале поля собственных полей группировки (для группировок), потом поля из глобальных настроек (для группировок типа [Детальные записи](#)) и самыми последними – ресурсы и поля из родительских элементов.

Отбор

Используется для фильтрации записей, попадающих в результат компоновки. Кроме того, может использоваться для отбора записей, к которым применяется некоторое оформление (условное оформление), и для создания пользовательских полей выбора.

Описывается с помощью объекта [ОтборКомпоновкиДанных](#). В свойстве [Элементы](#) этого объекта содержится коллекция элементов отбора, состоящая из объектов [ЭлементОтбораКомпоновкиДанных](#).

Группа элементов отбора

Используется для группировки элементов отбора, которая упорядочивает данные результата. Описывается с помощью объекта [ГруппаЭлементовОтбораКомпоновкиДанных](#).

Порядок

Описывает, каким образом нужно упорядочивать записи, выводимые в результат. Представляет собой объект [ПорядокКомпоновкиДанных](#). В свойстве [Элементы](#) этого объекта содержится коллекция элементов порядка, состоящая из объектов [ЭлементПорядкаКомпоновкиДанных](#).

Автоэлемент порядка

Перед использованием автоматический элемент порядка будет преобразован в набор элементов порядка.

Формирование набора происходит по следующим правилам: ресурсы добавляются безусловно, а из полей не ресурсов в порядок добавляются поля, являющиеся реквизитами поля группировки, и само поле группировки (для детальных записей будут занесены все поля). Поля группировки, которые не были указаны в глобальном упорядочивании, попадут в конец порядка. Если поле уже включено в данный порядок, повторно оно не добавляется.

Условное оформление

Описывает, каким образом оформлять различные поля результата. Представляет собой объект [УсловноеОформлениеКомпоновкиДанных](#). В свойстве [Элементы](#) этого объекта содержится коллекция элементов порядка, состоящая из объектов [ЭлементУсловногоОформленияКомпоновкиДанных](#).

В оформлении компоновки данных конструктора компоновки данных параметры [Формат](#), [Текст](#) редактируются как многоязычные.

Для элемента условного оформления имеется возможность указать, для каких областей необходимо применять условное оформление:

- К полю, выводимому в группировке;
- К полю, выводимому в иерархической группировке;
- К полю, выводимому в общем итоге;
- К заголовку полей;
- К области заголовка отчета;
- К области, в которой выводятся параметры отчета;
- К области, в которой выводятся значения отборов.

Оформляемые поля

Поля, к которым применяется оформление. Описываются с помощью объекта [ОформляемыеПоляКомпоновкиДанных](#). В свойстве [Элементы](#) этого объекта содержится коллекция оформляемых полей, состоящая из объектов [ОформляемоеПолеКомпоновкиДанных](#). Если поля не указаны, оформление будет применено ко всей области.

Параметры вывода

Значения параметров вывода определяют внешний вид соответствующих объектов. Для части параметров поддерживается наследование. В связи с этим коллекция параметров вывода для элемента может содержать параметры, не относящиеся к нему самому, но используемые в элементах, которые могут быть вставлены в подчиненную часть структуры настроек.

Параметры данных

Значения параметров данных, как правило, используются в запросах для фильтрации выборки.

Пользовательские поля

Пользовательские поля позволяют расширить множество используемых доступных полей, определяя собственные выражения либо наборы вариантов с условиями использования конкретного варианта.

Пользовательские поля описываются с помощью объекта [ПользовательскиеПоляКомпоновкиДанных](#). В свойстве [Элементы](#) этого объекта содержится коллекция пользовательских полей, состоящая из объектов двух видов:

- поле-выражение (объект [ПользовательскоеПолеВыражениеКомпоновкиДанных](#));
- поле-выбор (объект [ПользовательскоеПолеВыборКомпоновкиДанных](#)).

Тип поля определяется системой автоматически, на основе его свойств.

Варианты пользовательского поля

Описание набора альтернатив, определяющих значение поля выбора. Описываются с помощью объекта `ВариантыПользовательскогоПоляВыборКомпоновкиДанных`. В свойстве `Элементы` этого объекта содержится коллекция вариантов пользовательского поля-выбора, состоящая из объектов `ВариантПользовательскогоПоляВыборКомпоновкиДанных`.

Работа с автополями

Если один элемент структуры настроек содержит автополя `АвтоПолеГруппировкиСистемыКомпоновкиДанных`, `АвтоВыбранноеПолеСистемыКомпоновкиДанных` и `АвтоЭлементПорядкаСистемыКомпоновкиДанных`, они преобразовываются в следующем порядке:

- `АвтоПолеГруппировкиСистемыКомпоновкиДанных`,
- `АвтоВыбранноеПолеСистемыКомпоновкиДанных`,
- `АвтоЭлементПорядкаСистемыКомпоновкиДанных`.

10.3.6.3. Доступные объекты

Доступные объекты – набор, определяющий объекты, которые могут быть использованы в компоновке как вложенные. Например, вложенные отчеты.

10.3.6.4. Доступные поля

Доступные поля – это множество полей, которые могут быть использованы при настройке компоновки данных и будут распознаны и правильно обработаны на последующих этапах компоновки. Доступные поля различаются по применению. Выделены следующие коллекции полей:

- поля для выбора (свойство `ДоступныеПоляВыбора`);
- поля группировок (свойство `ДоступныеПоляГруппировок`);
- поля порядка (свойство `ДоступныеПоляПорядка`);
- поля параметров данных (свойство `ДоступныеПоляПараметровДанных`);
- поля отбора (свойство `ДоступныеПоляОтбора`);
- поля отбора элементов структуры – используются во всех элементах структуры, кроме верхнего (свойство `ДоступныеПоляОтбораЭлементовСтруктуры`);
- поля дополнительных отборов – используются в условном оформлении (свойство `ДоступныеПоляДополнительныхОтборов`).

Все перечисленные свойства содержат коллекции значений, элементами которых являются объекты `ДоступноеПолеКомпоновкиДанных`.

В конструкторе схемы компоновки данных и в настройках схемы компоновки данных отчета доступные поля располагаются в следующей последовательности: вначале поля, не являющиеся ресурсами, в алфавитном порядке по заголовкам, после них поля-ресурсы в алфавитном порядке по заголовком. Последними в списке отражаются системные папки.

Доступное поле отбора

Для использования в отборах реализован специальный тип доступных полей. Он обладает всеми свойствами обычного доступного поля, а также предоставляет наборы доступных видов сравнения и доступных значений поля, необходимых для корректного построения элементов отбора.

10.3.6.5. Компоновщик настроек компоновки данных

Компоновщик настроек представляется объектом встроенного языка системы «1С:Предприятие» `КомпоновщикНастроекКомпоновкиДанных`. Объект предназначен для связи настроек компоновки данных и схемы компоновки данных. На основе схемы компоновки данных строится источник доступных настроек для работы конструктора настроек.

При необходимости использовать источник доступных настроек одного компоновщика в другом, имеется возможность получить источник доступных настроек с помощью метода `ПолучитьИсточникДоступныхНастроек()` объекта `КомпоновщикНастроекКомпоновкиДанных`.

Пример передачи источника доступных настроек компоновки данных:

```
Перем ПараметрыФормы;
ПараметрыФормы = Новый Структура;
ПараметрыФормы.Вставить("ИсточникДоступныхНастроек", Список.КомпоновщикНастроек.ПолучитьИсточникДоступныхНастроек());
ПараметрыФормы.Вставить("Настройки", Список.КомпоновщикНастроек.Настройки);
ПараметрыФормы.Вставить("ФиксированныеНастройки", Список.КомпоновщикНастроек.ФиксированныеНастройки);
ПараметрыФормы.Вставить("ПользовательскиеНастройки", Список.КомпоновщикНастроек.ПользовательскиеНастройки);
ОткрытьФорму("ОбщаяФорма.ФормаКомпоновщикаНастроек", ПараметрыФормы);
```

[Копировать в буфер обмена](#)

10.3.7. Пользовательские настройки системы компоновки данных

Существует возможность отметки некоторых настроек для того, чтобы пользователь мог редактировать их в отдельной форме. Данный механизм называется **пользовательские настройки**.

При выполнении компоновки применяются как пользовательские, так и полные настройки. При этом пользовательские настройки «накладываются» на полные, формируя реально исполняемые настройки.

10.3.7.1. Объектная модель пользовательских настроек

При помощи пользовательских настроек редактируются следующие объекты:

- `ОтборКомпоновкиДанных`,
- `ЭлементОтбораКомпоновкиДанных`,
- `ГруппаЭлементовОтбораКомпоновкиДанных`,
- `ПорядокКомпоновкиДанных`,
- `ВыбранныеПоляКомпоновкиДанных`,
- `УсловноеОформлениеКомпоновкиДанных`,
- `ЭлементУсловногоОформленияКомпоновкиДанных`,
- `ЗначениеПараметраНастроекКомпоновкиДанных`,
- `ГруппировкаКомпоновкиДанных`,
- `ГруппировкаТаблицыКомпоновкиДанных`,
- `ГруппировкаДиаграммыКомпоновкиДанных`,
- `ТаблицаКомпоновкиДанных`,

- ДиаграммаКомпоновкиДанных,
- ВложеннаяСхемаКомпоновкиДанных,
- КоллекцияЭлементовСтруктурыКомпоновкиДанных,
- КоллекцияЭлементовСтруктурыТаблицыКомпоновкиДанных,
- КоллекцияЭлементовСтруктурыДиаграммыКомпоновкиДанных.

У данных объектов существуют следующие свойства:

- ИдентификаторПользовательскойНастройки – предназначено для идентификации объекта пользовательской настройки. Если данное свойство установлено, то объект считается пользовательским и подлежит редактированию в пользовательских настройках.

При интерактивной отметке настройки как пользовательской система автоматически генерирует уникальный идентификатор и заполняет данное свойство строковым представлением этого идентификатора.

- ПредставлениеПользовательскойНастройки – строка, используемая для отображения представления в пользовательских настройках. В конструкторе схемы возможен ввод представления на нескольких языках.
- РежимОтображения – используется для определения быстрых настроек. Подробнее о данной возможности будет рассказано ниже.

Пользовательские настройки в объектной модели представляются отдельным объектом **ПользовательскиеНастройкиКомпоновкиДанных**. Данный объект имеет свойство **Элементы**. В этой коллекции находятся элементы пользовательских настроек. Возможны объекты следующих типов:

- ОтборКомпоновкиДанных,
- ЭлементОтбораКомпоновкиДанных,
- ГруппаЭлементовОтбораКомпоновкиДанных,
- ПорядокКомпоновкиДанных,
- ВыбранныеПоляКомпоновкиДанных,
- УсловноеОформлениеКомпоновкиДанных,
- ЭлементУсловногоОформленияКомпоновкиДанных,
- ЗначениеПараметраКомпоновкиДанных,
- ГруппировкаКомпоновкиДанных,
- ГруппировкаТаблицыКомпоновкиДанных,
- ГруппировкаДиаграммыКомпоновкиДанных,
- ТаблицаКомпоновкиДанных,
- ДиаграммаКомпоновкиДанных,
- НастройкиВложенногоОбъектаКомпоновкиДанных,
- СтруктураНастроекКомпоновкиДанных.

10.3.7.2. Настройка пользовательских элементов настроек

Установить пользовательский признак элемента настройки можно в форме настройки пользовательского элемента, который вызывается при помощи команды **Свойства элемента пользовательских настроек**.

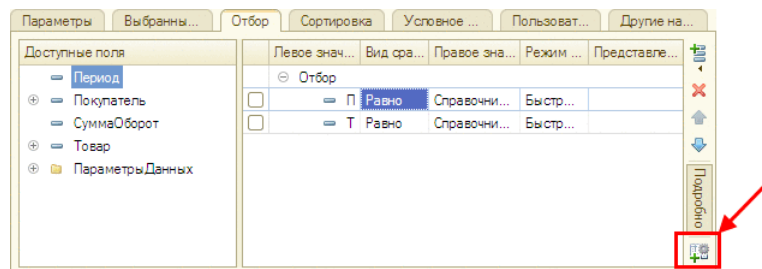


Рис. 274. Открыть пользовательскую настройку

В форме настройки пользовательского элемента также можно указать представление, которое будет использоваться для элемента, и режим его редактирования.

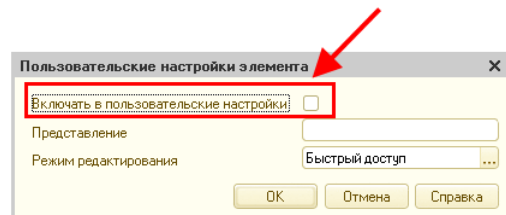


Рис. 275. Свойство пользовательской настройки

В списке структуры настроек команда **Свойства элемента пользовательских настроек** позволяет настраивать пользовательские настройки для текущего элемента структуры. Для каждого элемента структуры определен свой состав настраиваемых элементов.

Объект	Настраиваемые элементы
Отчет	<ul style="list-style-type: none"> • выбранные поля, • порядок,

	<ul style="list-style-type: none"> • отбор, • условное оформление, • состав группировок
Группировка/ группировка таблицы/ группировка диаграммы	<ul style="list-style-type: none"> • группировка, • выбранные поля, • отбор, • порядок, • условное оформление, • состав вложенных группировок
Диаграмма	<ul style="list-style-type: none"> • диаграмма, • выбранные поля, • условное оформление, • состав группировок серий, • состав группировок точек
Таблица	<ul style="list-style-type: none"> • таблица, • выбранные поля, • условное оформление, • состав группировок строк, • состав группировок колонок
Вложенная схема	<ul style="list-style-type: none"> • вложенный отчет, • выбранные поля, • отбор, • порядок, • условное оформление, • состав группировок

В зависимости от того, где происходит вызов, команда **Свойства** элемента пользовательских настроек позволяет изменять различные настройки:

- список отбора – пользовательские настройки для текущего элемента/групп отбора;
- список параметров вывода и параметров данных – пользовательские настройки для текущего параметра;
- список условного оформления – пользовательские настройки для текущего элемента условного оформления.

Кроме того, в таблице структуры команда **Пользовательские настройки** позволяет открыть модальную форму, в которой будут отображаться пользовательские настройки со своими значениями по умолчанию.

10.3.7.3. Редактирование пользовательских настроек

Редактирование объекта **Пользовательские настройки** осуществляется в таблице (см. [рис. 276](#)).

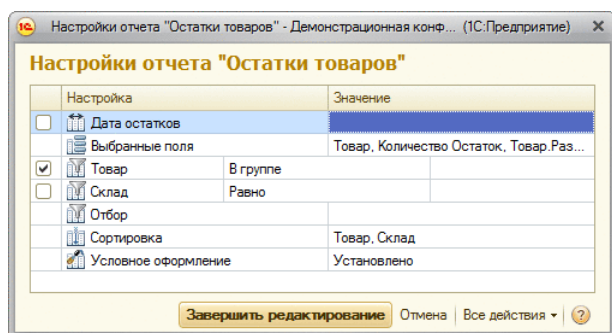


Рис. 276. Редактор пользовательских настроек

СОВЕТ. Имеет смысл выносить в пользовательские те настройки (из всего многообразия, предлагаемого системой компоновки данных), которые предназначены для управления отчетом конечным пользователем. Предполагается, что пользователь будет оперировать только этими настройками.

Например, в запросе, используемом в качестве набора данных для отчета, существует большое количество полей, по которым можно выполнять отбор. Однако разработчик отчета считает, что есть ряд элементов отбора, которые используются наиболее часто и их логично вынести в пользовательские настройки. Тогда пользователь будет иметь возможность отредактировать как «особые» элементы отбора (строка **Товары** на [рис. 276](#)), так и отбор целиком (строка **Отбор** на [рис. 276](#)).

Также логично выносить в пользовательские такие настройки, как:

- **Период или дата** – практически для всех отчетов;
- **Группировки и условное оформление** – для табличных отчетов;
- **Значение счета бухгалтерского учета** – для бухгалтерских отчетов и т. д.

При этом для каждого отчета разработчик будет самостоятельно принимать решение о том, какие настройки станут пользовательскими в его отчете (или варианте отчета).

10.3.7.4. Быстрые пользовательские настройки

Среди пользовательских настроек разработчик отчета может выделить такие настройки, которые пользователь редактирует наиболее часто (например, фильтр по товару в отчете **Остатки товаров** или фильтр по организации в бухгалтерском отчете). Тогда можно указать, что пользовательская настройка является быстрой (свойство **РежимОтображения** установлено в значение **Быстрый доступ**). Такие настройки будут редактироваться прямо в форме отчета.

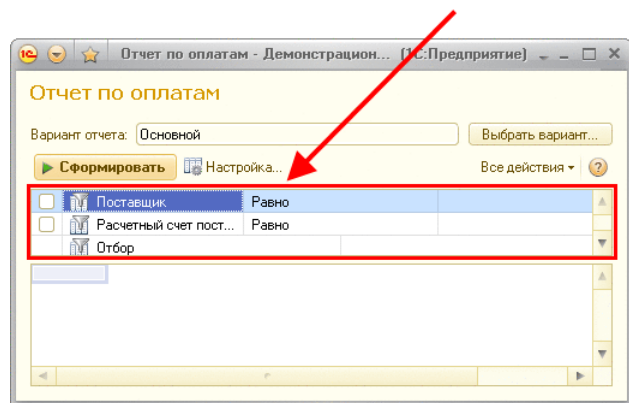


Рис. 277. Быстрые пользовательские настройки

У таблицы, предназначенной для редактирования пользовательских настроек, также имеется свойство **РежимОтображения**, которое определяет, показывать все пользовательские настройки или только быстрые.

Если пользователя не устраивает текущий состав быстрых пользовательских настроек, то он может самостоятельно изменить их состав, например, исключить из состава быстрых пользовательских настроек те, которые он не планирует изменять часто.

Для такой настройки пользователю необходимо выбрать команду **Еще – Изменить состав настроек...** (**Все действия – Изменить состав настроек...**) в окне редактирования настроек отчета.

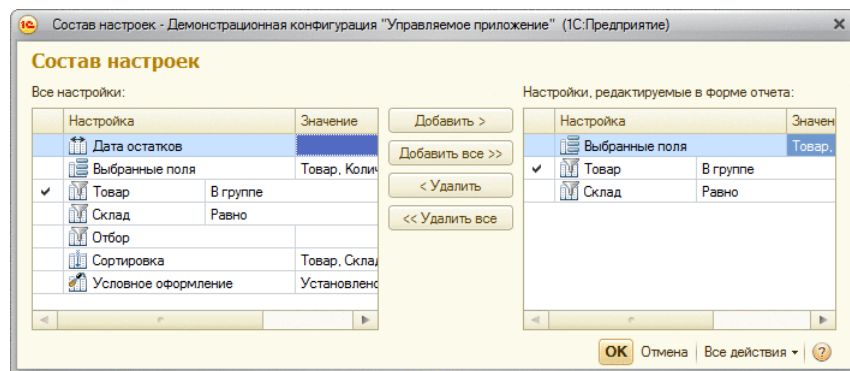


Рис. 278. Редактирование состава быстрых пользовательских настроек

В левой части формы представлены все пользовательские настройки, которые могут быть выбраны быстрыми, а в правой – настройки, которые в данный момент редактируются в форме отчета.

В формах редактирования отбора или условного оформления присутствуют команды редактирования свойств элемента пользовательских настроек. Таким образом, пользователь может перенести в состав быстрых пользовательских настроек те элементы отбора или условного оформления, которые он изменяет постоянно в своей работе.

Управлять составом быстрых пользовательских настроек можно также из формы редактирования настроек с помощью колонки **Редактирование в форме отчета**. Эта колонка по умолчанию не отображается в форме, и ее необходимо включить с помощью команды **Еще – Изменить форму...** (**Все действия – Изменить форму...**) окна редактирования настроек.

В этом случае нельзя добавлять новые настройки, но можно оперативно менять состав существующих настроек.

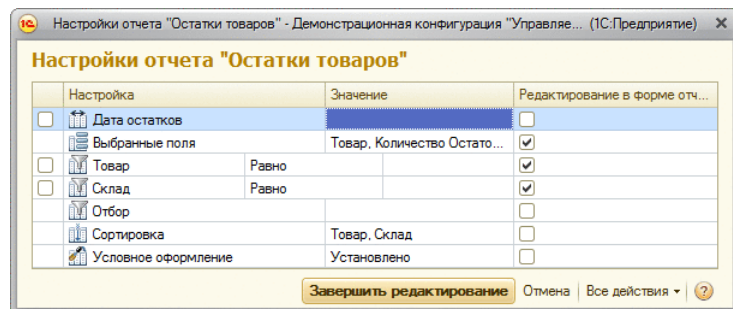


Рис. 279. Колонка «Редактирование в форме отчета»

10.3.7.5. Компоновщик настроек

У компоновщика настроек имеется свойство **ПользовательскиеНастройки**. В данном свойстве находятся значения редактируемых пользовательских настроек. Свойство недоступно для записи с помощью встроенного языка. Кроме того, у компоновщика настроек имеется метод **ЗагрузитьПользовательскиеНастройки()**, который загружает значения пользовательских настроек, переданные в качестве параметра метода.

Метод **ПолучитьНастройки()** позволяет получить копию текущих настроек (с учетом пользовательских настроек).

Метод **ЗагрузитьНастройки()** загружает переданные настройки в компоновщик настроек (пользовательские настройки также перезаполняются на основании переданных данных).

10.3.7.6. Заполнение значений пользовательских настроек

При заполнении значений пользовательских настроек для различных элементов настроек в пользовательские настройки добавляются соответствующим образом заполненные элементы.

Для типов **ЭлементОтбораКомпоновкиДанных**, **ГруппаЭлементовОтбораКомпоновкиДанных**, **ЭлементУсловногоОформленияКомпоновкиДанных**, **ЗначениеПараметраКомпоновкиДанных**, **ГруппировкаКомпоновкиДанных**, **ГруппировкаТаблицыКомпоновкиДанных**, **ГруппировкаДиаграммыКомпоновкиДанных**, **ТаблицаКомпоновкиДанных**, **ДиаграммаКомпоновкиДанных**,

ВложеннаяСхемаКомпоновкиДанных в соответствующих пользовательских элементах заполняется свойство ИдентификаторПользовательскойНастройки и свойства, которые реально редактируются в пользовательских настройках.

Для типов ОтборКомпоновкиДанных, УсловноеОформлениеКомпоновкиДанных, ПорядокКомпоновкиДанных, ВыбранныеПоляКомпоновкиДанных создается объект соответствующего типа. В его коллекцию добавляются элементы, у которых свойство РежимОтображения отлично от Недоступный.

Имеются исключения:

- Не будут добавляться элементы, которые сами отмечены как пользовательские. Например, в пользовательский отбор не будет помещен элемент отбора, который отмечен как пользовательский.
- Не будут добавлены элементы, содержащие пользовательские элементы. Например, не будет добавлена группа условий, если в этой группе присутствуют элементы, отмеченные как пользовательские.
- Для вложенных элементов свойство РежимОтображения не анализируется. Они добавляются или не добавляются вместе с родительскими элементами.

Для типов КоллекцияЭлементовСтруктурыКомпоновкиДанных, КоллекцияЭлементовСтруктурыТаблицыКомпоновкиДанных, КоллекцияЭлементовСтруктурыДиаграммыКомпоновкиДанных создается объект СтруктураНастроекКомпоновкиДанных, в который помещаются группировки, уже присутствующие в структуре. Помещаются только группировки с установленными полями группировок (не помещаются детальные записи). Помещение группировок происходит до того, как будут встречены детальные записи, ветвление, таблица, диаграмма, вложенная схема, неиспользуемая группировка, группировка с пользовательской структурой.

10.3.7.7. Применение пользовательских настроек

Применение пользовательских настроек к основным настройкам выполняется в методе ПолучитьНастройки() компоновщика настроек. При этом выполняются описанные ниже действия:

- Для типов ЭлементОтбораКомпоновкиДанных, ЭлементУсловногоОформленияКомпоновкиДанных, ЗначениеПараметраКомпоновкиДанных содержимое элементов копируется в соответствующие пользовательские элементы настроек.
- Для типов ОтборКомпоновкиДанных, УсловноеОформлениеКомпоновкиДанных, ПорядокКомпоновкиДанных, ВыбранныеПоляКомпоновкиДанных элементы, находящиеся в основных настройках и отмеченные как Недоступный, остаются без изменения. Элементы из пользовательских настроек переносятся в основные. Они добавляются в конец коллекции для Отбора, ВыбранныхПолей и УсловногоОформления и в начало коллекции – для Порядка.
- Для типов ГруппаЭлементовОтбораКомпоновкиДанных, ГруппировкаКомпоновкиДанных, ГруппировкаТаблицыКомпоновкиДанных, ГруппировкаДиаграммыКомпоновкиДанных, ТаблицаКомпоновкиДанных, ДиаграммаКомпоновкиДанных, НастройкиВложенногоОбъектаКомпоновкиДанных устанавливается свойство Использование в соответствующем элементе основных настроек (на основании признака Использование элемента пользовательских настроек).
- Для типа СтруктураНастроекКомпоновкиДанных в элементе структуры основных настроек ищутся соответствующие группировки и располагаются в правильном порядке. Недостающие группировки создаются. Не найденные в пользовательских настройках группировки либо группировки, отключенные пользователем, не удаляются, а помечаются особым образом. Это позволяет сохранить их для возможного использования в будущем. Пользовательские группировки с пустым набором полей (детальные записи) при применении игнорируются.

10.4. Макет компоновки данных

Макет компоновки данных представляется объектом встроенного языка системы «1С:Предприятие» МакетКомпоновкиДанных и состоит из множества других вложенных объектов. Макет компоновки данных является инструкцией по выполнению компоновки данных для системы компоновки данных. Макет компоновки уже содержит в себе описание макетов областей, тексты исполняемых запросов, расположение группировок и т. д.

10.4.1. Составные части макета компоновки данных

Каждый макет компоновки данных содержит множество объектов, описывающих ту или иную часть. Рассмотрим эти составные части.

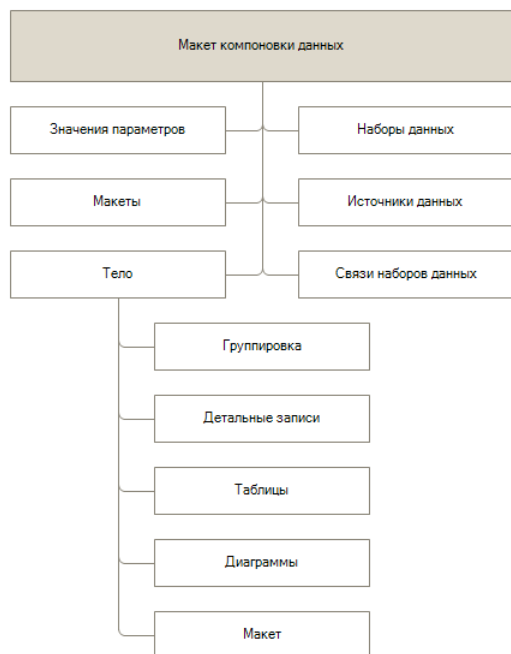


Рис. 280. Составные части макета компоновки данных

Источники данных

Макет компоновки данных может содержать несколько описаний источников данных.

Под источником данных подразумевается источник, из которого будут получаться данные. В качестве источника данных выступает информационная база системы «1С:Предприятие».

Источники данных описываются в свойстве ИсточникиДанных макета, которое содержит коллекцию значений, состоящую из элементов ИсточникДанных МакетаКомпоновкиДанных.

Допускается создание нескольких источников данных, указывающих при помощи строки соединения на одну информационную базу.

Наборы данных

Наборы данных макета компоновки данных содержат описание того, какие данные необходимо получать в компоновке данных.

Они описываются в свойстве `НаборыДанных` макета компоновки данных.

Допускается наличие нескольких наборов данных.

Вложенный набор данных рассматривается как обычный набор данных. Вложенный набор данных всегда связан с его родительским набором данных. Если для вложенного набора данных указано условие фильтра, тогда связь вложенного набора данных с родительским набором данных считается внутренней.

Поле набора данных макета компоновки данных

Набор данных может содержать описания полей, которые будут доступны для этого набора данных. Поля, описания которых в наборе данных отсутствуют, являются недоступными. Если некоторое поле присутствует в запросе набора данных, но отсутствует в описании полей набора данных, то поле не будет доступно для использования.

Поля набора данных описываются в свойстве `Поля` набора данных, которое содержит коллекцию значений, состоящую из элементов `ПолеНабораДанныхМакетаКомпоновкиДанных`.

Значения параметров макета компоновки данных

Макет компоновки данных может содержать параметры в любых выражениях, присутствующих в нем.

Значения параметров описываются в свойстве `ЗначенияПараметров` макета компоновки данных, которое содержит коллекцию значений, состоящую из элементов `ЗначениеПараметраМакетаКомпоновкиДанных`.

Связи наборов данных макета компоновки данных

Наборы данных, присутствующие в макете компоновки данных, могут быть связаны друг с другом. Связи между наборами данных описываются в макете компоновки данных.

Связи наборов данных описываются в свойстве `СвязиНаборовДанных` макета компоновки данных, которое содержит коллекцию значений, состоящую из элементов `СвязьНаборовДанныхМакетаКомпоновкиДанных`.

Макеты областей макета компоновки данных

При исполнении компоновки в результат компоновки будут выводиться макеты областей. Эти макеты областей также находятся в макете компоновки данных.

Описания макетов находятся в свойстве `Макеты` макета компоновки данных, которое содержит коллекцию значений, состоящую из элементов `ОписаниеМакетаОбластиМакетаКомпоновкиДанных`.

Тело макета компоновки данных

Предыдущие составные части макета компоновки данных содержали информацию о том, откуда получать информацию. Само же указание, как следует скомпоновать данные, находится в теле макета компоновки данных. Тело макета компоновки данных состоит из элементов.

Возможно использование следующих типов элементов:

- группировка – описывает выводимую в результат группировку;
- детальные записи – описывает выводимые в результат детальные записи набора данных;
- таблица – описывает таблицу, выводимую в результат;
- диаграмма – описывает диаграмму, выводимую в результат;
- макет – описывает макеты, используемые при выводе.

Группировка таблицы

Описывает группировку таблицы и представляется объектом встроенного языка `ГруппировкаТаблицыМакетаКомпоновкиДанных`.

Группировка таблицы содержит те же свойства, что и обычная группировка, со следующими различиями:

- Тело группировки таблицы и иерархическое тело группировки таблицы могут содержать только те элементы, которые содержатся в теле таблицы, т. е. только группировки таблицы, детальные записи таблицы, макеты группировки таблицы. Иерархическое тело может дополнительно содержать иерархическую группировку таблицы, обозначающую место, в которое будут выводиться иерархические записи группировки.
- В качестве макетов заголовков и подвала используются макеты группировки таблицы.
- Возможно использование свойств:
 - `МакетОбщихИтогов` – указывает макет, используемый при выводе общих итогов по группировке, типа `МакетГруппировкиТаблицыМакетаКомпоновкиДанных`;
 - `РасположениеОбщихИтогов` – содержит расположение общих итогов для группировки, типа `РасположениеИтоговКомпоновкиДанных`.

Детальные записи таблицы

Описывают детальные записи таблицы, представляются объектом встроенного языка `ЗаписиТаблицыМакетаКомпоновкиДанных`.

Макет группировки таблицы

Макет группировки таблицы описывает макеты, используемые при выводе группировки таблицы, и представляется объектом встроенного языка `МакетГруппировкиТаблицыМакетаКомпоновкиДанных`.

Группировка диаграммы

Описывает группировку диаграммы и представляется объектом встроенного языка `ТелоГруппировкиДиаграммыМакетаКомпоновкиДанных`.

Группировка диаграммы содержит те же свойства, что и обычная группировка, со следующими различиями:

- Тело группировки диаграммы и иерархическое тело группировки диаграммы могут содержать только макеты группировки диаграммы и группировки диаграммы. Иерархическое тело может дополнительно содержать иерархическую группировку диаграммы, обозначающую место, в которое будут выводиться иерархические записи группировки.
- Отсутствуют макеты для заголовка и подвала.

Макет группировки диаграммы

Макет группировки диаграммы описывает макеты, используемые при выводе группировки диаграммы, и представляется объектом встроенного языка `МакетГруппировкиДиаграммыМакетаКомпоновкиДанных`.

10.4.2. Макеты областей

Макет области представляет собой декларативное описание расположения выводимых данных, а также их визуальное оформление, необходимое для вывода данных в

Существуют несколько принципиально различных макетов областей:

- собственно макет области,
- макеты областей диаграммы.

Структура макетов показана на [рис. 281](#).

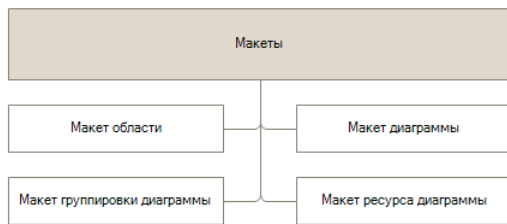


Рис. 281. Структура макета

Во встроенном языке системы «1С:Предприятие» макет области представляется объектом типа `МакетОбластиКомпоновкиДанных`. Данный объект является коллекцией объектов `СтрокаТаблицыОбластиКомпоновкиДанных`.

10.4.2.1. Структура макета области

Как было сказано выше, макет области представляет собой коллекцию объектов типа `СтрокаТаблицыОбластиКомпоновкиДанных`. `Строка` таблицы представляет собой коллекцию ячеек, расположенных горизонтально слева направо. Таким образом, несколько идущих подряд строк таблицы области компоновки данных образуют прямоугольную таблицу.

Коллекция ячеек таблицы

Данный объект представляет собой коллекцию ячеек строки таблицы. Описывается объектом встроенного языка `ЯчейкаТаблицыОбластиКомпоновкиДанных`. Элементами коллекции являются ячейки таблицы – объекты типа `ЯчейкаТаблицыОбластиКомпоновкиДанных`.

Ячейка таблицы представляет собой прямоугольную область, используемую для вывода данных в документы различных форматов. Внутри ячейки могут содержаться выводимые поля, текст и оформления.

Коллекция элементов макета

Коллекция элементов макета (объект `ЭлементыМакетаОбластиКомпоновкиДанных`) представляет собой коллекцию полей (объектов типа `ПолеОбластиКомпоновкиДанных`). Данные объекты могут содержаться в коллекции в произвольном порядке. Данные, содержащиеся в этих объектах, используются при выводе в документы различных форматов.

Поле

Этот элемент представляет собой поле, выводимое в ячейке таблицы или в элементе списка. Внутри поля может содержаться произвольное значение и его оформление. Поле описывается объектом встроенного языка `ПолеОбластиКомпоновкиДанных`.

Оформление ячейки таблицы

Оформление ячейки таблицы представляет собой коллекцию объектов, описывающих оформление ячейки таблицы. Описывается объектом встроенного языка `ОформлениеЯчейкиТаблицыОбластиКомпоновкиДанных`.

Оформление поля

Оформление поля представляет собой коллекцию, содержащую всего один объект – элемент оформления `Формат`. Описывается объектом встроенного языка `ОформлениеПоляОбластиКомпоновкиДанных`.

10.4.2.2. Структура макетов областей диаграммы

Существует три типа макетов областей диаграммы:

- макет диаграммы – объект типа `МакетДиаграммыОбластиКомпоновкиДанных`;
- макет ресурса диаграммы – объект типа `МакетРесурсаДиаграммыОбластиКомпоновкиДанных`;
- макет группировки диаграммы – объект типа `МакетГруппировкиДиаграммыОбластиКомпоновкиДанных`.

При создании макетов диаграммы формируется один макет диаграммы, один макет ресурса диаграммы и несколько макетов группировок диаграммы. Количество макетов группировок соответствует количеству точек и серий в диаграмме.

Принцип работы

Макеты областей используются при выводе отчетов в документы различных форматов. Макет области является составной частью определения макета компоновки данных – объекта типа `ОписаниеМакетаСхемыКомпоновкиДанных`.

Для того чтобы в ячейках или элементах списка макета области выводились значения выводимых полей отчета, необходимо свойству `Значение` поля области компоновки данных (объект типа `ПолеОбластиКомпоновкиДанных`) присвоить значение типа `ПараметрСистемыКомпоновкиДанных`, содержащее имя параметра. Сам параметр необходимо добавить к списку параметров определения макета и в качестве имени присвоить имя параметра, а в качестве выражения – имя выводимого поля или выражение в терминах языка выражений системы компоновки данных.

Макет диаграммы

Макет диаграммы используется для описания типа диаграммы. Описывается объектом встроенного языка `МакетДиаграммыОбластиКомпоновкиДанных`.

Макет ресурса диаграммы

Макет ресурса диаграммы используется для формирования значений диаграммы и описывается объектом встроенного языка `МакетРесурсаДиаграммыОбластиКомпоновкиДанных`.

Макет группировки диаграммы

Макет группировки диаграммы используется для формирования точек и серий диаграммы. Описывается объектом встроенного языка `МакетГруппировкиДиаграммыОбластиКомпоновкиДанных`.

10.4.3. Макеты оформления

Макет оформления представляет собой декларативное описание предопределенных областей отчета. Данные описания используются генератором областей макетов при

формировании макетов областей на основании элементов настройки компоновки данных.

Для интерактивного создания макета оформления необходимо при создании макета с помощью конструктора макетов указать его тип [Макет оформления компоновки данных](#) и нажать кнопку [Готово](#). На экран выводится окно макета оформления.

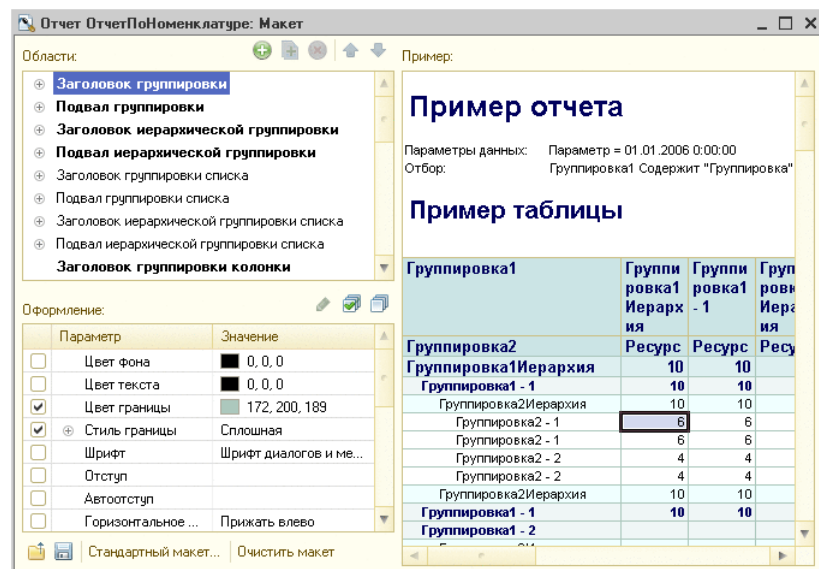


Рис. 282. Окно макета оформления

Окно конструктора состоит из списка областей оформления, таблицы настройки оформления и поля табличного документа [Пример](#) для показа результата выбранных настроек.

Непустые области макета отображаются жирным шрифтом.

С помощью контекстного меню перетаскивания параметры оформления могут быть скопированы параметрами из одной области в другую или заменены.

С помощью кнопки [Стандартный макет](#) можно загрузить уже подготовленный макет из списка стандартных макетов, в который также войдут общие макеты конфигурации.

Кнопка [Очистить макет](#) удаляет оформление в макете.

Последовательность действий для настройки оформления следующая:

- выбирается область оформления;
- в таблице настроек указываются (устанавливаются пометки) параметры, оформление которых требуется изменить, и указываются значения оформления;
- результат внесенных изменений контролируется в поле [Пример](#).

Если предполагается, что в отчет будет выведено несколько уровней группировки, то нужно в списке областей для каждой области, в которой будут уровни, создать подчиненные области по числу уровней группировки. Для создания уровня области нужно указать область и нажать кнопку [Добавить](#) командной панели. В список областей добавляется строка, имя которой [УровеньN](#), где [N](#) – номер уровня группировки. Выбор оформления уровня группировки выполняется как описано выше.

ПРИМЕЧАНИЕ. Если было указано несколько уровней, то при удалении уровня группировки всегда удаляется самый нижний уровень текущей области, независимо от того, какой уровень в списке был выбран.

Структура макета оформления

Каждый макет оформления состоит из некоторого количества областей макета оформления. Область макета оформления имеет имя – строку из перечня областей макетов оформления, указывающую, к какой области применять данный макет, и коллекцию элементов оформления.

Элемент оформления имеет два свойства:

- [Уровень](#) – положительное число, причем если уровень равен 0, то считается, что оформление применяется по умолчанию ко всем макетам области определенного типа;
- [Оформление](#) – коллекция, содержащая имена свойств оформления и их значения.

Во встроенном языке системы «1С:Предприятие» макет оформления представляется объектом типа [МакетОформленияКомпоновкиДанных](#).

Макет оформления компоновки данных

Данный объект представляет собой коллекцию областей макета оформления. Эта коллекция реализована в виде значений параметров. Именем параметра является имя области. Имя области является предопределенной строкой из перечня областей. Значением параметра является объект типа [ОбластьМакетаОформленияКомпоновкиДанных](#).

Область макета оформления компоновки данных

Данный объект представляет собой описание предопределенной области и является коллекцией элементов области макета оформления (объектов типа [ЭлементОбластиМакетаОформленияКомпоновкиДанных](#)).

Элемент области макета оформления компоновки данных

Данный объект представляет собой описание области макета для определенного уровня иерархии. Если в качестве номера уровня указан 0, то считается, что данный макет оформления применяется ко всем областям данного типа.

10.4.4. Генератор областей макетов

Генератор областей макетов позволяет динамически формировать макеты областей компоновки данных, используемых для вывода результата компоновки данных в документы различных форматов. Под макетом области подразумевается декларативное описание расположения выводимых данных и их оформление.

При работе генератора областей макетов можно выделить следующие этапы:

- формирование общих макетов отчета,
- формирование макетов группировок,
- расположение группировок,
- расположение выводимых полей группировок,

- применение макета оформления,
- применение условных оформлений,
- объединение ячеек.

Рассмотрим каждый этап работы генератора областей макетов.

Формирование общих макетов отчета

На данном этапе генератором областей макетов на основании настройки компоновки данных создаются общие макеты. Тип и количество макетов зависят от типа элемента настройки компоновки данных.

Макет группировки

Для группировки формируется специальный макет – шапка группировки. Данный макет содержит названия выводимых полей в левой части макета и названия выводимых ресурсных полей в правой части макета, например:

Контрагент	Контрагент.Код		Количество	Сумма
Номенклатура	Номенклатура.Код	Номенклатура.Наименование		

Макет таблицы

Для таблицы формируется следующая группа макетов:

- **Макет шапки таблицы.** Данный макет содержит названия выводимых полей строк таблицы, например:

Контрагент	Контрагент.Код	
Номенклатура	Номенклатура.Код	Номенклатура.Наименование

- **Макет итогов по строкам.** Данный макет содержит специальное слово **Итого** и названия ресурсных полей, если они выводятся горизонтально, например:

Итого	
Количество	Сумма

- **Макет итогов по колонкам.** Данный макет содержит специальное слово **Итого** и названия ресурсных полей, если они расположены вертикально, например:

Итого	Количество
	Сумма

- **Макет общих итогов.** Данный макет содержит ресурсные поля, выводимые в таблице и необходимые для отображения в таблице общих итогов, например:

Представление(Сумма(Продажи.КоличествоОборот))	Представление(Сумма(Продажи.СуммаОборот))
--	---

Расположение данных макетов внутри таблицы показано ниже.

Шапка таблицы	Область колонок	Макет итогов по строкам
Область строк	Область ресурсов	Область итогов по строкам
Макет итогов по колонкам	Область итогов по колонкам	Макет общих итогов

Расположение макетов итогов по строкам и колонкам управляется свойствами [РасположениеОбщихИтоговПоГоризонтали](#) и [РасположениеОбщихИтоговПоВертикали](#) соответственно. Возможны следующие варианты расположения общих итогов:

- **Нет** – общие итоги не выводятся.

Шапка таблицы	Область колонок
Область строк	Область ресурсов

- **Начало** – общие итоги выводятся в первой колонке или в первой строке таблицы соответственно.

Шапка таблицы	Макет итогов по строкам	Область колонок
Макет итогов по колонкам	Макет общих итогов	Область итогов по колонкам
Область строк	Область итогов по строкам	Область ресурсов

- **Конец** – общие итоги выводятся в последней колонке или в последней строке таблицы соответственно.

Шапка таблицы	Область колонок	Макет итогов по строкам
Область строк	Область ресурсов	Область итогов по строкам
Макет итогов по колонкам	Область итогов по колонкам	Макет общих итогов

- **НачалоИКонец** – общие итоги выводятся в первой колонке/строке, а также в последней колонке/строке таблицы.

Шапка таблицы	Макет итогов по строкам	Область колонок	Макет итогов по строкам
Макет итогов по колонкам	Макет общих итогов	Область итогов по колонкам	Макет общих итогов
Область строк	Область итогов по строкам	Область ресурсов	Область итогов по строкам
Макет итогов по колонкам	Макет общих итогов	Область итогов по колонкам	Макет общих итогов

- **Авто** – итоги по колонкам располагаются в последней строке, а итоги по строкам – в последней колонке.

Макет диаграммы

В макете содержатся оформительские свойства диаграммы.

Формирование макетов группировок

В настройке компоновки данных существует три вида группировок:

- группировка,
- группировка таблицы,
- группировка диаграммы.

Для каждого вида группировок формируется свой набор макетов областей. Однако расположение группировок друг относительно друга, расположение выводимых полей внутри областей группировок и расположение ресурсных полей делаются единообразно. При формировании любого макета области группировки можно выделить следующие этапы.

Определение типа макета группировки. Тип макета группировки получается из параметра **ТипМакетаГруппировкиКомпоновкиДанных** объекта **ЗначенияПараметровВыводаКомпоновкиДанных**. Данное свойство имеет смысл только для простых группировок, т. е. группировок, не входящих в таблицу и диаграмму. Возможны следующие варианты расположения выбранных полей:

- **Авто** – автоматическое определение расположения выбранных полей: если в группировке есть вложенная таблица, диаграмма, вложенный отчет или группировка с типом макета группировки **Вертикальный**, то выбранные поля располагаются вертикально, иначе – горизонтально.
- **Горизонтальный** – расположение выбранных полей горизонтально, друг за другом слева направо.

Контрагент	Контрагент.Код	
Номенклатура	Номенклатура.Код	Номенклатура.Наименование
Алекс-2002	00009	
1С:Бухгалтерия 8	00013	1С:Бухгалтерия 8

- **Вертикальный** – расположение выбранных полей вертикально, друг под другом.

Контрагент	Алекс-2002
Контрагент.Код	00009
Номенклатура	1С:Бухгалтерия 8
Номенклатура.Код	00013
Номенклатура.Наименование	1С:Бухгалтерия 8

Расположение группировок друг относительно друга. Расположение группировок друг относительно друга управляется параметром **РасположениеПолейГруппировки** объекта **ЗначенияПараметровВыводаКомпоновкиДанных**. Возможны следующие варианты расположения:

- **Вместе** – группировки располагаются друг под другом. Например, для группировок по контрагенту и номенклатуре.

Контрагент	Контрагент.Код	
Номенклатура	Номенклатура.Код	Номенклатура.Наименование
Алекс-2002	00009	
1С:Бухгалтерия 8	00013	1С:Бухгалтерия 8

- **Отдельно** – каждая группировка располагается в отдельной области. Группировки располагаются друг за другом слева направо. Выводимые поля группировки выводятся также и во вложенных группировках.

Контрагент	Контрагент.Код	Номенклатура	Номенклатура.Код	Номенклатура.Наименование
Алекс-2002	00009			
Алекс-2002	00009	1С:Аспект 7.7	00015	1С:Аспект 7.7. Компактная торговая система

- **ОтдельноИТолькоВитогах** – каждая группировка располагается в отдельной области. Группировки располагаются друг за другом слева направо. Выводимые поля выводятся только в данной группировке.

Контрагент	Контрагент.Код	Номенклатура	Номенклатура.Код	Номенклатура.Наименование
Алекс-2002	00009			
		1С:Аспект 7.7	00015	1С:Аспект 7.7. Компактная торговая система

- **Расположение выводимых полей.** Существуют поля двух видов: собственно выводимые поля (поля-владельцы и/или их реквизиты) и ресурсные поля. Вывод данных полей имеет существенные различия.

• **Вывод полей.** Как указано выше, существуют поля-владельцы и поля-реквизиты. Поля-владельцы выводятся в макет области в соответствии с их порядком следования в настройке компоновки данных. Расположение полей-реквизитов управляется специальным параметром **РасположениеРеквизитов** объекта **ЗначенияПараметровВыводаКомпоновкиДанных**. Возможны следующие варианты расположения:

- **Вместе** – поля реквизитов располагаются вместе в отдельной колонке и при выводе разделяются запятой.

Контрагент	Контрагент.Код
Номенклатура	Номенклатура.Код, Номенклатура.Наименование

- **Отдельно** – каждое поле реквизита располагается в отдельной колонке.

Контрагент	Контрагент.Код	
Номенклатура	Номенклатура.Код	Номенклатура.Наименование

- **ВместеСВладельцем** – поля реквизитов располагаются в одной колонке с их полем-владельцем и при выводе разделяются запятой.

Контрагент, Контрагент.Код
Номенклатура, Номенклатура.Код, Номенклатура.Наименование

- **ВСпециальнойПозиции** – поля-реквизиты располагаются в специальной колонке, размещенной правее всех остальных колонок группировки.

Контрагент	Контрагент.Код
Номенклатура	Номенклатура.Код, Номенклатура.Наименование

• **Вывод полей, расположенных в папках.** Расположение данных полей управляется свойством **Расположение папки**. Возможны следующие варианты расположения:

- **Авто** – поля выводятся в зависимости от типа группировки. Если группировка простая, то поля выводятся горизонтально; если группировка табличная, то вертикально.

Контрагент	
Номенклатура	Реквизиты номенклатуры
	Номенклатура.Код Номенклатура.Наименование Номенклатура.ОснПост

Контрагент	
Номенклатура	Реквизиты номенклатуры
	Номенклатура.Код
	Номенклатура.Наименование
	Номенклатура.ОснПост

- **Горизонтально** – поля выводятся горизонтально слева направо.

Контрагент	
Номенклатура	Реквизиты номенклатуры
	Номенклатура.Код Номенклатура.Наименование Номенклатура.ОснПост

- **Вертикально** – поля выводятся вертикально друг под другом.

Контрагент	
Номенклатура	Реквизиты номенклатуры
	Номенклатура.Код
	Номенклатура.Наименование
	Номенклатура.ОснПост

- **ВОтдельнойКолонке** – поля выводятся в отдельной колонке, расположенной правее всех остальных колонок.

Контрагент	Контрагент.Код	
Номенклатура	Реквизиты номенклатуры	
	Номенклатура.Код Номенклатура.Наименование Номенклатура.ОснПост	

- **Вместе** – поля выводятся вместе и при выводе разделяются запятой.

Контрагент	
Номенклатура	Реквизиты номенклатуры
	Номенклатура.Код, Номенклатура.Наименование, Номенклатура.ОсновнойПоставщик

• **Вывод ресурсных полей.** Специальное свойство **РасположениеРесурсов** объекта **ПараметрыВывода** управляет расположением ресурсных полей. Возможны следующие варианты расположения:

- **Горизонтально** – поля ресурсов располагаются горизонтально слева направо.
- **Вертикально** – поля ресурсов располагаются вертикально друг под другом.

Если для папки не установлен заголовок, то место для заголовка в макете не выделяется. В этом случае поля выводятся в соответствии со значением свойства **Расположение папки**.

Применение макета оформления

После расположения группировок и полей к сформированной области применяется макет оформления. Смысл применения макета оформления состоит в том, что всем элементам макета области добавляются ранее определенные свойства оформления, например, **ЦветФона** или **ЦветТекста**. Стоит отметить, что существует большое количество макетов оформления областей.

Применение условных оформлений

После применения макета оформления к макету области применяются условные оформления. Применение условного оформления заключается в том, что в макет области добавляются свойства оформления, содержащие логические выражения. В результате вычисления логического выражения принимается решение, какое значение оформительского свойства необходимо применить.

Объединение ячеек

После формирования макета области в ней могут оказаться незаполненные ячейки. Такие ячейки необходимо объединять для более наглядного представления макета области. Возможны следующие варианты расположения ячеек.

- Незаполненные ячейки расположены справа от заполненных ячеек. Ячейки, расположенные справа, объединяются с ячейками слева.

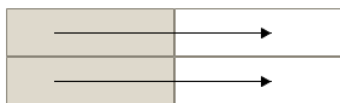


Рис. 283. Незаполненные ячейки справа

- Незаполненные ячейки расположены под заполненными ячейками. Ячейки, расположенные снизу, объединяются с ячейками сверху.



Рис. 284. Незаполненные ячейки снизу

- Незаполненные ячейки расположены и справа, и под заполненными ячейками. Ячейки, расположенные справа, объединяются с ячейками слева. Затем ячейки, расположенные снизу, объединяются с ячейками сверху.

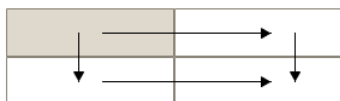


Рис. 285. Незаполненные ячейки снизу и справа

10.4.5. Формирование макета компоновки данных

Формирование макета компоновки данных осуществляется на основании схемы компоновки данных и настроек. Схема компоновки данных создается разработчиком отчета, настройки вводятся пользователем.

Формирование макета компоновки осуществляется при помощи объекта встроенного языка `КомпоновщикМакетаКомпоновкиДанных`.

Данный объект не имеет свойств и имеет один метод `Выполнить()`, в который передается схема компоновки данных, настройки компоновки данных, переменная, в которую помещается объект `ДанныеРасшифровки`, и макет оформления. Данный метод возвращает созданный макет компоновки данных.

В процессе работы компоновщик макета:

- Создает макет компоновки.
- Модифицирует запросы наборов данных для получения нужной пользователю информации и помещает их в макет компоновки.
- Формирует фильтры в тексте запроса или описании набора данных.
- При необходимости создает наборы данных для получения и проверки иерархии.
- В макете компоновки данных создает нужные параметры со значениями, установленными пользователем.
- Заполняет элементы тела макета компоновки данных, в которые помещает группировки, таблицы и т. д., заполняет их параметры.
- Использует генератор областей макета (см. [здесь](#)). Если указан макет оформления (четвертый параметр), данный макет будет использоваться для оформления генерируемого макета.
- Если указан третий параметр, создает объект `ДанныеРасшифровки` и помещает его в переданную переменную. Созданный таким образом объект нужно использовать при работе процессора компоновки и при обработке расшифровки.

Компоновщик макета при генерации макета компоновки данных анализирует значение параметра `Группировка` функции `ВычислитьВыражение()`. Если параметр содержит имя группировки, которая присутствует в настройках, то компоновщик макета оставляет данное имя без изменения. В противном случае компоновщик макета ищет среди родительских группировок текущей группировки группировку, содержащую в полях группировки все поля, указанные в параметре `Группировка` функции `ВычислитьВыражение()`, и проставляет в данный параметр имя найденной группировки.

Например, если в качестве параметра функции `ВычислитьВыражение()` указана строка `Номенклатура, ХарактеристикаНоменклатуры`, то при генерации макета компоновщик макета найдет группировку по полям `Номенклатура` и `ХарактеристикаНоменклатуры` и в сгенерированный макет поместит имя найденной группировки. При этом группировка ищется среди группировок, которые доступны в месте вычисления группировки. Так, если компоновщик макета генерирует выражение для ячейки таблицы, то поиск группировки будет осуществляться в группировках колонок, для которых выводится ячейка, группировках строк, для которых выводится ячейка, и группировках, в которые вложена таблица. Таким образом, ищутся группировки, к которым может быть применена функция `ВычислитьВыражение()`. Если найдено несколько, то выбирается группировка, которая ближе. В таблице сначала ищется группировка в строке, после этого ищется группировка в колонке, затем ищется группировка над таблицей.

Если группировка не найдена, то выражение заменяется на `NULL`.

Фактически это позволяет использовать `ВычислитьВыражение()` в пользовательских полях, указывая группировки не по именам группировок, а по именам полей группировок.

Пример:

```
СхемаКомпоновкиДанных = ПолучитьСхемуКомпоновкиДанных();
ИсполняемыеНастройки = ПолучитьИсполняемыеНастройки();
КомпоновщикМакета = Новый КомпоновщикМакетаКомпоновкиДанных;
МакетКомпоновки = КомпоновщикМакета.Выполнить(СхемаКомпоновкиДанных, ИсполняемыеНастройки);
```

Созданный макет компоновки данных может быть использован для исполнения, т. е. для получения результата.

10.5. Процессор компоновки данных

Исполнение компоновки данных осуществляется при помощи объекта системы «1С:Предприятие» `ПроцессорКомпоновкиДанных`.

На вход процессору компоновки данных передается макет компоновки данных.

Работа с процессором компоновки данных предельно проста: после установки процессору компоновки данных макета компоновки данных у данного объекта можно последовательно получать элементы результата компоновки данных, которые в дальнейшем можно использовать, например, для вывода в табличный документ или сохранить для последующего использования.

Ниже приведен пример работы с процессором компоновки данных.

[Копировать в буфер обмена](#)

```
ЭлементыФормы.ТДРезультатТабличныйДокумент.Очистить();
МакетКД = ПолучитьМакетКомпоновки();
ПроцессорКД = Новый ПроцессорКомпоновкиДанных;
ПроцессорКД.Инициализировать(МакетКД);
ПроцессорВывода =
Новый ПроцессорВыводаРезультатаКомпоновкиДанныхВТабличныйДокумент;
ПроцессорВывода.УстановитьДокумент(ЭлементыФормы.ТДРезультатТабличныйДокумент);
ПроцессорВывода.НачатьВывод();
Пока Истина Цикл
    ЭлементРезультатаКД = ПроцессорКД.Следующий();
    Если ЭлементРезультатаКД = Неопределено Тогда
        Прервать;
    КонецЕсли;
    ПроцессорВывода.ВывестиЭлемент(ЭлементРезультатаКД);
КонецЦикла;
ПроцессорВывода.ЗакончитьВывод();
```

При инициализации процессора компоновки данных можно дополнительно указать:

- объект – внешние наборы данных – структуру, у которой в качестве ключа содержится имя внешнего набора данных, а в качестве значения – набор данных;
- данные расшифровки – объект, в который будет помещаться информация о расшифровке;
- возможность использования внешних функций – признак, можно ли в выражениях использовать функции общих модулей конфигурации.

10.6. Функциональные опции и права на просмотр поля в отчете

При получении настроек отчета по умолчанию система компоновки данных автоматически выполняет следующие действия с настройками:

- Если у пользователя нет права на интерактивный просмотр некоторого поля или данное поле связано с выключенными функциональными опциями, это поле становится недоступным для настройки пользователем. То есть оно не будет отображаться в списке доступных полей.
- Если поле связано с реквизитом, тип которого выключен функциональной опцией, то это поле удаляется из списка доступных полей.
- Таблица запроса и реквизиты этой таблицы становятся недоступными, если функциональная опция отключает объект конфигурации, образующий эту таблицу.
- Если поле, на просмотр которого у пользователя нет права, или поле, которое связано с выключенной функциональной опцией, используется в пользовательском поле, то такое пользовательское поле удаляется и по нему не устанавливается отбор.
- Поле системы компоновки данных считается недоступным, если все поля, используемые в выражении этого поля, связаны с выключенными функциональными опциями или недоступными на просмотр полями. При этом учитываются все поля, входящие в выражение поля в запросе, за исключением полей, участвующих в условиях запросов и полей, участвующих в разделе **КОГДА** операции выбора (**ВЫБОР КОГДА ИНАЧЕ КОНЕЦ**), а также выражения из объединений, если таковые присутствуют.
- Из полей группировки удаляются все поля, на которые у пользователя нет права на просмотр, и поля, которые связаны с выключенными функциональными опциями. Если после удаления поля группировки у группировки не остается ни одного поля группировки с установленным признаком использования, то удаляется вся группировка, а ее содержимое (если, например, группировка включает таблицу) помещается на место группировки.
- Если после удаления из таблицы группировки-строки или группировки-колонки таблица остается без группировок, то таблица удаляется.
- Если после удаления из диаграммы группировки-серии или группировки-точки диаграмма остается без группировок, то диаграмма удаляется.
- Из упорядочивания удаляются все поля, для которых у пользователя нет права на просмотр и которые связаны с выключенными функциональными опциями.
- Из оформляемых полей элемента условного оформления удаляются поля, на которые у пользователя нет права на просмотр и которые связаны с выключенными функциональными опциями. Если после удаления оформляемого поля из элемента условного оформления в элементе не остается ни одного оформляемого поля с включенным признаком использования, то такой элемент условного оформления также удаляется.
- Если в отборе элемента условного оформления использовалось поле, для которого у пользователя нет права на просмотр, или поле, которое связано с выключенными функциональными опциями, то такой элемент условного оформления удаляется.

Если поле ранее было доступно и пользователь каким-либо образом сохранил настройку, а в дальнейшем (когда поле стало для него недоступно) загрузит настройку, поля не будут автоматически удаляться из настройки. Это сделано для того, чтобы пользователь имел возможность заменить недоступные поля другими полями или самостоятельно удалить их использование из настроек.

Если при выполнении метода `Выполнить()` объекта `КомпоновщикМакетаКомпоновки` параметр `ПроверятьДоступностьПолей` установлен в значение `Истина`, то будет осуществляться проверка доступности полей для текущего пользователя и проверка наличия поля во включенной функциональной опции. При использовании в настройках недоступного поля будет выдаваться исключение. Если значение параметра `Ложь`, то проверка осуществляться не будет.

ПРИМЕЧАНИЕ. В случае выполнения отчета из автоматически сгенерированной формы будет осуществляться проверка доступности полей для текущего пользователя и проверка наличия поля во включенной функциональной опции.

10.7. Результат компоновки данных

Результат компоновки данных представляется набором элементов результата компоновки данных. Как объект встроенного языка системы «1С:Предприятие» результат компоновки данных не существует, существует лишь набор элементов результата компоновки данных, которые и образуют результат.

При необходимости элементы результата компоновки данных могут быть помещены в некоторую универсальную коллекцию значений, например `Массив`, для того чтобы манипулировать результатом как единым целым.

Элементы результата могут быть получены при помощи объекта `ПроцессорКомпоновкиДанных`, а также могут быть созданы и заполнены средствами встроенного языка.

Элементы результата компоновки данных можно вывести в табличный документ при помощи процессора вывода.

Рассмотрим пример элементов данных.

Элемент 1

Свойство	Значение
Тип элемента	Начало

Макеты	ЗаголовокТаблицы, ЗаголовокКолонки, ЗаголовокСтроки, Ресурсы
Расположение вложенных элементов	Вертикально

Элемент 2

Свойство	Значение
Тип элемента	Начало
Расположение вложенных элементов	Горизонтально

Элемент 3

Свойство	Значение
Тип элемента	НачалоКонец
Имя макета	ЗаголовокТаблицы

Элемент 4

Свойство	Значение
Тип элемента	НачалоКонец
Имя макета	ЗаголовокКолонки

Элемент 5

Свойство	Значение
Тип элемента	Конец

Элемент 6

Свойство	Значение
Тип элемента	Начало
Расположение вложенных элементов	Горизонтально

Элемент 7

Свойство	Значение
Тип элемента	НачалоКонец
Имя макета	ЗаголовокСтроки

Элемент 8

Свойство	Значение
Тип элемента	НачалоКонец
Имя макета	Ресурсы

Элемент 9

Свойство	Значение
Тип элемента	Конец

Элемент 10

Свойство	Значение
Тип элемента	Конец

Результат вывода таких элементов должен выглядеть следующим образом:

ЗаголовокТаблицы	ЗаголовокКолонки
ЗаголовокСтроки	Ресурсы

Если бы элемент 2 содержал макеты `ЗаголовокТаблицы` и `ЗаголовокСтроки`, то при выводе элемента 3 использовался бы макет из этого элемента, однако при выводе элемента 7 использовался бы макет из элемента 1, т. к. элемент 2 завершается элементом 5.

Элементы результата могут быть сохранены в XML стандартными средствами, например:

```

ЗаписьXML = Новый ЗаписьXML;
ЗаписьXML.УстановитьСтроку();
ЗаписьXML.ЗаписатьНачалоЭлемента("result");
МакетКомпоновкиДанных = ПолучитьМакетКомпоновки();
ПроцессорКомпоновкиДанных = Новый ПроцессорКомпоновкиДанных;
ПроцессорКомпоновкиДанных.Инициализировать(МакетКомпоновкиДанных);
Пока Истина Цикл
    ЭлементРезультатаКомпоновкиДанных = ПроцессорКомпоновкиДанных.Следующий();
    Если ЭлементРезультатаКомпоновкиДанных = Неопределено Тогда
        Прервать;
    КонецЕсли;
    СериализаторXDTO.ЗаписатьXML(ЗаписьXML, ЭлементРезультатаКомпоновкиДанных, "item", "http://v8.1c.ru/8/data-composition-system/result");
КонецЦикла;
ЗаписьXML.ЗаписатьКонецЭлемента();
ЭлементыФормы.РезультатКомпоновкиДанных.УстановитьТекст(ЗаписьXML.Закрыть());

```

10.7.1. Вывод результата компоновки в табличный документ

Вывод отчета в табличный документ осуществляется при помощи объекта `ПроцессорВыводаРезультатаКомпоновкиДанныхВТабличныйДокумент`.

Элементы результата компоновки могут быть получены при помощи процессора компоновки данных либо сформированы любыми другими средствами.

Пример вывода результата компоновки данных в табличный документ:

```

ЭлементыФормы.ТДРезультатТабличныйДокумент.Очистить();
МакетКомпоновкиДанных = ПолучитьМакетКомпоновки();
ПроцессорКомпоновкиДанных = Новый ПроцессорКомпоновкиДанных;
ПроцессорКомпоновкиДанных.Инициализировать(МакетКомпоновкиДанных);
ПроцессорВывода = Новый ПроцессорВыводаРезультатаКомпоновкиДанныхВТабличныйДокумент;
ПроцессорВывода.УстановитьДокумент(ЭлементыФормы.ТДРезультатТабличныйДокумент);
ПроцессорВывода.НачатьВывод();
Пока Истина Цикл
    ЭлементРезультатаКомпоновкиДанных = ПроцессорКомпоновкиДанных.Следующий();
    Если ЭлементРезультатаКомпоновкиДанных = Неопределено Тогда
        Прервать;
    КонецЕсли;
    ПроцессорВывода.ВывестиЭлемент(    ЭлементРезультатаКомпоновкиДанных);
КонецЦикла;
ПроцессорВывода.ЗакончитьВывод();

```

[Копировать в буфер обмена](#)

Также имеется возможность использовать метод `Вывести()` объекта `ПроцессорВывода`. В качестве параметра метода следует указать `ПроцессорКомпоновкиДанных`. В этом случае вывод результата компоновки будет выглядеть так:

```
ПроцессорВывода.Вывести(ПроцессорКомпоновкиДанных);
```

[Копировать в буфер обмена](#)

Процессор компоновки данных выполняет фиксацию заголовка таблицы, если в отчет выводится одна таблица или одна группировка (возможно, с вложенными группировками). Управлять возможностями фиксации (сверху и слева) можно с помощью настроек параметров вывода `Фиксация слева` и `Фиксация сверху`.

10.7.2. Вывод результата компоновки в таблицу и дерево значений

Вывод отчета в таблицу или дерево значений осуществляется при помощи объекта `ПроцессорВыводаРезультатаКомпоновкиДанныхВКоллекциюЗначений`. Метод `УстановитьОбъект()` является аналогом метода `УстановитьДокумент()`. Если метод `УстановитьОбъект()` не был вызван, результат будет выведен в таблицу значений.

Элементы результата компоновки могут быть получены при помощи процессора компоновки данных либо сформированы любыми другими средствами.

Пример вывода результата компоновки данных в дерево значений:

```

КомпоновщикМакета = Новый КомпоновщикМакетаКомпоновкиДанных;
МакетКомпоновкиДанных = КомпоновщикМакета.Выполнить(СхемаКомпоновкиДанных,КомпоновщикНастроек.Настройки, , , Тип("ГенераторМакетаКомпоновкиДанныхДляКоллекц
ПроцессорКомпоновкиДанных = Новый ПроцессорКомпоновкиДанных;
ПроцессорКомпоновкиДанных.Инициализировать(МакетКомпоновкиДанных);
ПроцессорВывода = Новый ПроцессорВыводаРезультатаКомпоновкиДанныхВКоллекциюЗначений;
ПроцессорВывода.УстановитьОбъект(ДеревоРезультата);
ПроцессорВывода.НачатьВывод();
Пока Истина Цикл
    ЭлементРезультатаКомпоновкиДанных = ПроцессорКомпоновкиДанных.Следующий();
    Если ЭлементРезультатаКомпоновкиДанных = Неопределено Тогда
        Прервать;
    КонецЕсли;
    ПроцессорВывода.ВывестиЭлемент(    ЭлементРезультатаКомпоновкиДанных);
КонецЦикла;
ПроцессорВывода.ЗакончитьВывод();

```

[Копировать в буфер обмена](#)

При выводе результата компоновки в таблицу или дерево значений существуют следующие ограничения:

- В настройках должны присутствовать только группировки и детальные записи. Использование таблиц и диаграмм не допускается.
- Все папки, указанные в выбранных полях, игнорируются.
- Не используется условное оформление, а также оформление для поля, указанное в схеме компоновки данных.
- Из параметров вывода используются только следующие:
 - расположение общих итогов по вертикали,
 - тип заголовка полей,
 - количество записей,
 - процент записей.
- Предопределенные макеты не используются.

Для вывода результата в таблицу или дерево значения реализовано два вида макетов: макет для заголовка (`МакетЗаголовкаКоллекцииЗначенийОбластиКомпоновкиДанных`) и макет содержимого (`МакетКоллекцииЗначенийОбластиКомпоновкиДанных`).

10.8. Расчет итогов по полям остатка в системе компоновки данных

Поле остатка с точки зрения макета компоновки данных является то, у которого в роли проставлен признак `Остаток`.

10.8.1. Расчет итогов по полям остатка

Если в макете компоновки данных в некотором наборе данных присутствует поле начального остатка, то в наборе данных также должно присутствовать соответствующее ему поле конечного остатка, и наоборот.

Все поля-периоды, описанные в наборе данных, должны иметь непрерывную нумерацию, начинающуюся с единицы.

Для корректного расчета итогов по полям источника данных данные должны удовлетворять следующему правилу: в данных должна соблюдаться уникальность значений полей-периодов и полей-измерений, т. е. данные не должны содержать строк с одинаковыми значениями полей-периодов и полей-измерений.

При расчете итогов по полям-остаткам используется следующий алгоритм:

- если требуется осуществить расчет итога поля остатка для группировки по полю-периоду:
 - если по всем полям-периодам уже была осуществлена группировка:
 - для каждой комбинации полей измерений, по которым осуществлялась группировка:
 - получается запись, ближайшая к текущему периоду;

- если полученная запись была на текущий период, то из данной записи будут получаться начальные и конечные остатки;
- иначе, если полученная запись имеет предыдущий период, конечный остаток записи будет использован как начальный и конечный остаток;
- иначе начальный остаток полученной записи будет использоваться как начальный и конечный остаток;
- иначе (группировка еще не произведена по всем полям-периодам):
 - для каждой комбинации полей измерений, по которым осуществлялась группировка:
 - получаются первая и последние записи, у которых поля использованных периодов равны текущему периоду;
 - если записи найдены, то первая запись будет использоваться как начальный остаток, последняя – как конечный;
 - если записи не найдены, то получается ближайшая запись и ее остатки используются как начальные остатки и конечные остатки в зависимости от того, предшествует ли найденная запись текущему периоду;
- иначе (не группировка по полю-периоду):
 - первые по хронологии записи для неиспользованных полей-измерений будут использоваться в качестве записей начального остатка, последние – в качестве конечного.

10.8.2. Расчет итогов по полям бухгалтерских остатков

Расчет итогов по полям бухгалтерских остатков выполняется аналогично расчету итогов по обычным полям-остаткам. Кроме того, при расчете итогов по таким полям используется информация о поле-счете. Если итоги рассчитываются для группировки по полю-счету или если до группировки, для которой считаются итоги, была осуществлена группировка по полю-счету, при расчете итога используется вид счета. В противном случае вид счета считается активно-пассивным.

В зависимости от вида счета итоговый остаток будет вычисляться по следующим формулам:

- Если полученный остаток больше 0:

	Дт	Кт
Активный	Остаток	0
Пассивный	0	-Остаток
АП	Остаток	0

- Если полученный остаток меньше 0:

	Дт	Кт
Активный	Остаток	0
Пассивный	0	-Остаток
АП	0	-Остаток

10.8.3. Компоновка макета

Для обеспечения расчета корректных итогов компоновщик макета при генерации макета выполняет дополнительные действия:

- если используется начальный остаток, автоматически добавит в запрос и поле конечного остатка, даже если он не используется, и наоборот;
- если в отчете используется поле-реквизит измерения, автоматически добавит в запрос само поле-измерение, даже если оно не используется.

10.9. Работа с иерархией в системе компоновки данных

В системе компоновки данных имеются следующие аспекты использования иерархии:

- иерархические группировки,
- условие **В ИЕРАРХИИ**.

Рассмотрим перечисленные аспекты подробнее.

10.9.1. Иерархические группировки

При создании группировки пользователь может указать для некоторого поля группировки необходимость выполнения иерархической группировки.

Для того чтобы система выполнила иерархическую группировку, процессору компоновки данных необходимо знать, откуда получать данные для построения иерархии. Реализуется это путем создания набора данных с указанием связи набора данных самого к себе.

Рассмотрим пример. Допустим, необходимо построить иерархию для поля типа `Справочник.Номенклатура`.

Набор данных для построения иерархии будет выглядеть так:

ВЫБРАТЬ [Копировать в буфер обмена](#)
 Номенклатура.Ссылка КАК Ссылка,
 Номенклатура.Родитель КАК Родитель
ИЗ
 Справочник.Номенклатура КАК Номенклатура
ГДЕ
 Номенклатура.Ссылка В (&Ссылки)

Для этого набора данных должна быть определена связь от поля `Родитель` к полю `Номенклатура` с параметром `Ссылка`. Таким образом, набор данных позволит последовательно получить всех родителей элемента.

Набор данных для построения иерархии может быть либо явно описан в схеме компоновки данных, либо автоматически сгенерирован компоновщиком макета компоновки данных.

В случае указания набора данных иерархии в схеме компоновки необходимо добавить, например, запрос следующего вида:

ВЫБРАТЬ [Копировать в буфер обмена](#)
 Номенклатура.Ссылка КАК Ссылка,
 Номенклатура.Родитель КАК Родитель
{ВЫБРАТЬ
 Ссылка.*,
 Родитель}
ИЗ
 Справочник.Номенклатура КАК Номенклатура

ГДЕ [Номенклатура.Ссылка](#) В(&Ссылки)

Этот набор данных необходимо связать сам с собой, как было описано выше. Кроме того, к набору данных нужно создать связь от поля, для которого требуется создать иерархию.

10.9.2. Иерархические детальные записи

Существует возможность выводить в отчет детальные записи с иерархией, используя настройки связей наборов данных.

Например, определим набор данных [Номенклатура](#):

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
Номенклатура.Ссылка,
Номенклатура.Родитель,
Номенклатура.Код,
Номенклатура.Наименование,
Номенклатура.ЭтоГруппа
ИЗ
Справочник.Номенклатура КАК Номенклатура
ГДЕ Номенклатура.Родитель В (&Родители)

```

Определим связь набора данных [Номенклатура](#) с самим собой.

В качестве поля выражения-источника установим поле [Ссылка](#), в качестве выражения-приемника – поле [Родитель](#). Параметр связи – поле [Родители](#) с возможностью получения списка параметров. Таким образом, из каждой строки набора данных система будет выбирать значение поля [Ссылка](#), и для него будут получены записи, в которых это значение содержится в поле [Родитель](#). Для того чтобы дочерние элементы выбирались только в группах, установим выражение условия связи, равное полю [ЭтоГруппа](#). В качестве начального значения иерархической связи будем использовать выражение [Значение\(Справочник.Номенклатура.ПустаяСсылка\)](#). Оно означает, что на первом уровне иерархического набора данных будут получены записи, поле [Родитель](#) в которых является пустой ссылкой на справочник [Номенклатура](#).

Для вывода такого набора данных в отчет в настройках схемы компоновки достаточно вывести детальные записи (добавьте в список выбранных полей поля [Код](#) и [Наименование](#)).

Полученный результат приведен на [рис. 286](#).

0000027	Мышь Ice Mouse MUS-2
0000028	Мышь LOGITECH M-S48 PS/2
0000029	Мышь GENIUS "EASY" (3 кнопки)
0000005	Клавиатуры
0000018	Клавиатура Apple Pro Keyboards
0000020	Клавиатура LK-601 KB-2000 PS/2
0000006	Программное обеспечение
0000045	Windows
0000009	Windows XP Home Edition Russian CD
0000010	Windows XP Home Edition Russian UPG CD
0000011	Windows XP Professional Russian CD
0000012	1С:Бухгалтерия 8. Базовая версия
0000013	1С:Бухгалтерия 8.
0000014	1С:Бухгалтерия бюджетного учреждения 8
0000015	1С:Зарплата и Управление Персоналом 8
0000016	1С:Предприятие 8. Управление торговлей
0000007	Услуги
0000041	Доставка
0000042	Инсталляция ПО
0000043	Консультации по настройке ОС Windows
0000044	Консультации по настройке 1С
0000033	Ноутбуки
0000034	Ноутбук Rover Computers Navigator KT7
0000035	Ноутбук Rover Computers Explorer
0000046	Телефоны
0000047	Телефон Vega 700
0000048	Телефон Vega 300
0000051	Телефон Siemens SL45
0000052	Телефон LG W7200

Рис. 286. Иерархические детальные записи

10.9.3. Вывод одного элемента в нескольких родительских записях

Система компоновки данных позволяет выводить один элемент в нескольких родительских записях при построении иерархии.

Приведем пример.

В конфигурации создадим два справочника: [Дети](#) и [Сотрудники](#), а также регистр сведений [ДетиСотрудников](#) с измерениями [Ребенок](#) и [РодительРебенка](#).

В справочнике [Сотрудники](#) создадим записи: [Иванова Мария](#), [Иванов Иван](#). В справочнике [Дети](#) создадим запись: [Иванов Степан](#).

Допустим, в регистре сведений [ДетиСотрудников](#) есть записи:

Ребенок	Родитель
Иванов Степан	Иванов Иван
Иванов Степан	Иванова Мария

Создадим запрос набора данных [Дети](#), который будет получать список детей:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
Дети.Ссылка
ИЗ
Справочник.Дети КАК Дети

```

Создадим также запрос набора данных [Иерархия](#), который будет получать иерархические записи:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
ДетиСотрудников.Ребенок КАК Ссылка,
ДетиСотрудников.РодительРебенка КАК Родитель
ИЗ
РегистрСведений.ДетиСотрудников КАК ДетиСотрудников
ГДЕ
ДетиСотрудников.Ребенок В(&Ссылка)
ОБЪЕДИНИТЬ ВСЕ
ВЫБРАТЬ
Сотрудники.Ссылка,
NULL
ИЗ
Справочник.Сотрудники КАК Сотрудники

```

ГДЕ
Сотрудники.Ссылка В(&Ссылка)

Первое объединение запроса выбирает родителей детей. Вторая часть запроса выбирает сотрудников, так как иерархический набор должен содержать и сами иерархические записи.

Опишем связи между наборами данных.

В качестве источника связи укажем набор данных *Дети*, приемника – набор данных *Иерархия*. Выражение-источник – поле *Ссылка*, выражение-приемник – поле *Ссылка*. Параметром связи укажем поле *Ссылка*. Укажем, что возможен список параметров.

Опишем иерархическую связь. Набор данных *Иерархия* свяжем сам с собой, поле-источник – *Родитель*, поле-приемник – *Ссылка*. Параметром связи укажем поле *Ссылка* и укажем возможность списка параметров.

ВНИМАНИЕ! Поле иерархического набора данных, с которым осуществляется связь набора данных, должно именоваться так же, как и в исходном наборе данных. В противном случае система не сможет получать реквизиты полей для иерархического набора, в том числе не сможет получать представление для иерархических значений.

В настройках отчета создадим иерархическую группировку по полю *Ссылка*, представление которого изменим на *Родитель*, и выполним отчет:

```

Ребенок
Иванов Иван
  Иванов Степан
Иванова Мария
  Иванов Степан
  
```

Рис. 287. Результирующая таблица

В результате запись *Иванов Степан* выведена в обеих группировках.

10.9.4. Условие «В ИЕРАРХИИ»

Пользователь может указать для поля условие *В ИЕРАРХИИ*. В этом случае пользователю должны выдаться записи, находящиеся в иерархии указанной ссылки.

Если такое условие наложено в глобальном фильтре, оно попадет в текст запроса в виде условия *В ИЕРАРХИИ*. Если условие используется не в глобальном фильтре, то, чтобы обработать условие, процессору компоновки данных необходимо иметь набор данных, который будет содержать ссылки, удовлетворяющие условиям.

Такой набор данных может быть либо явно описан в схеме компоновки данных, либо будет автоматически сгенерирован компоновщиком макета.

10.10.Использование системы компоновки данных при разработке прикладных решений

В процессе разработки прикладных решений система компоновки данных может быть использована средствами встроенного языка в соответствии с описанной выше объектной моделью.

Кроме этого, система компоновки данных может быть задействована при визуальном конструировании отчетов. Например, после создания объекта конфигурации *Отчет* можно создать макет этого отчета, содержащий схему компоновки данных. Для этого следует нажать кнопку открытия у поля ввода *Основная схема компоновки данных*.

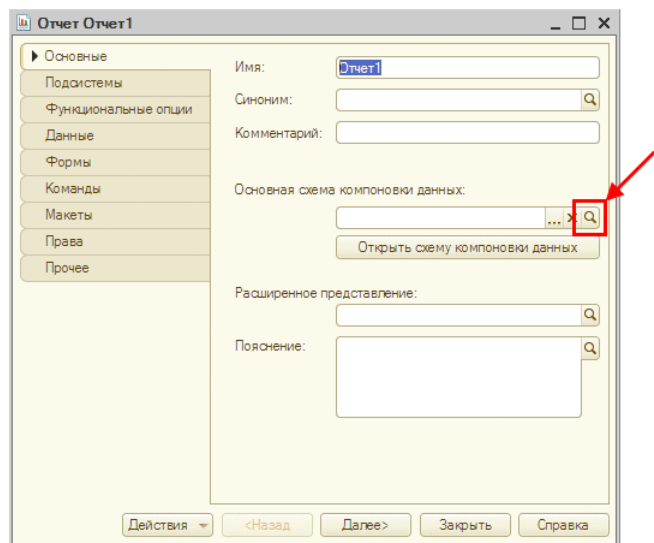


Рис. 288. Создание схемы компоновки данных

Нажатие кнопки *Открыть схему компоновки данных* позволит открыть основную схему компоновки данных. Если схемы нет, то будет создана новая и назначена основной.

В результате этих действий будет открыт конструктор макета, который позволит создать макет, содержащий схему компоновки данных.

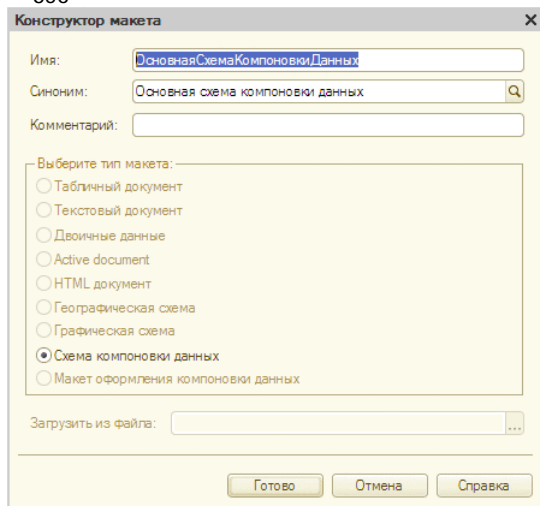


Рис. 289. Конструктор макета

После нажатия кнопки **Готово** будет открыт конструктор схемы компоновки данных, с помощью которого можно создать схему компоновки данных (см. [здесь](#)) или загрузить существующую схему компоновки из документа XML.

После создания схемы компоновки данных отчет готов к работе и может быть запущен в режиме 1С:Предприятие. Система при запуске отчета автоматически сгенерирует форму отчета и форму настроек.

На закладке **Формы** кроме основной формы отчета можно указать:

- форму настроек отчета, которая будет появляться при выполнении команды **Настройка...** в режиме 1С:Предприятие и которая используется для задания параметров пользовательских настроек;
- форму редактирования варианта отчета (которая будет открываться при выборе пункта меню **Еще – Изменить вариант...** (**Все действия – Изменить вариант...**) автоматической формы отчета), с помощью которой продвинутый пользователь сможет редактировать текущий вариант отчета (см. [здесь](#)).

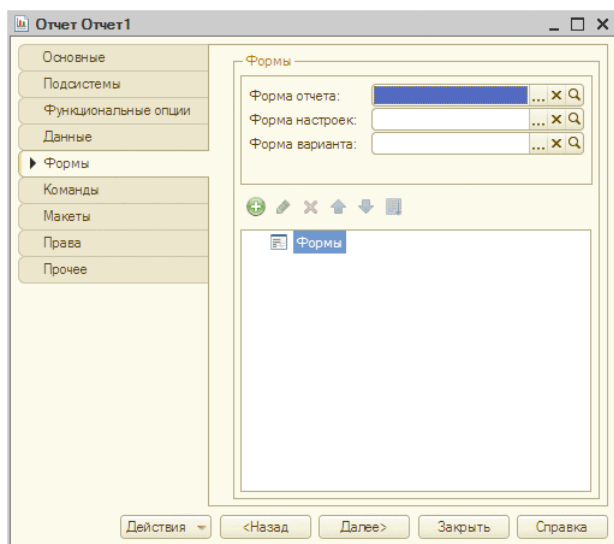


Рис. 290. Настройка форм отчета

Если формы, которые автоматически генерирует система при использовании отчета на базе системы компоновки данных, не устраивают конечных пользователей системы, то прикладной разработчик может создать свои формы. Для этого следует воспользоваться возможностями конструктора форм отчета.

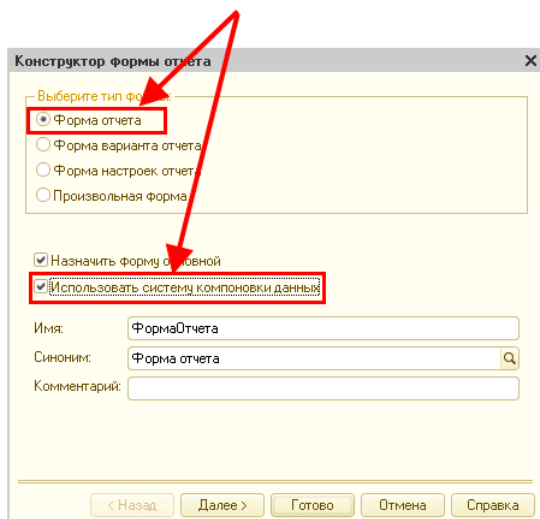


Рис. 291. Конструктор формы отчета

10.10.1. Параметры ввода

В схеме компоновки данных для полей наборов данных, вычисляемых полей и параметров имеется возможность указать параметры ввода. Данные параметры описывают, каким образом следует вводить значения для полей в отборе и параметрах. Например, в отчете [Остатки товаров на складах](#) можно указать, что для поля [Номенклатура](#) будет доступен [Быстрый выбор](#). Тогда, задавая значение отбора по номенклатуре, мы сможем пользоваться этой возможностью. Также можно указать особую форму выбора для поля [Склад](#) того же отчета.

Подробнее о параметрах ввода можно посмотреть во встроенной справке.

10.10.2. Использование в отчетах свойств объектов метаданных

Если в схеме компоновки данных используется набор данных – запрос, то для полей запроса из объектов метаданных получается информация о параметрах ввода и некоторых параметрах оформления. Таким образом, если в схеме компоновки данных не указан некоторый параметр ввода или вывода, то его значение будет автоматически получаться из соответствующего объекта метаданных.

Из объекта метаданных получают следующие параметры ввода:

- маска,
- связи параметров выбора,
- связь по типу,
- элемент связи по типу,
- форма выбора,
- формат редактирования,
- быстрый выбор.

Кроме того, из объекта метаданных получают следующие параметры вывода:

- формат,
- выделять отрицательные,
- многострочный режим.

10.10.3. Фоновое выполнение отчетов

При эксплуатации системы могут возникать ситуации, когда отчеты формируются долго. При этом желательно, чтобы пользователь во время выполнения такого длительного отчета имел возможность выполнять другие операции с системой. Такую возможность предоставляет механизм фонового выполнения отчетов (с использованием механизма фоновых заданий).

Автоматически создаваемая форма отчета использует фоновое выполнение отчета в клиент-серверном варианте работы системы, а в файловом варианте – непосредственное формирование отчета. Если в клиент-серверном варианте работы не требуется фоновое выполнения отчета, то можно использовать метод расширения формы отчета [СкомпоноватьРезультат\(\)](#), указав в качестве параметра признак непосредственного формирования отчета.

Пример:

```
СкомпоноватьРезультат(РежимКомпоновкиРезультата.Непосредственно);
```

[Копировать в буфер обмена](#)

Если во время фонового выполнения отчета будет выполнено интерактивное изменение пользовательских настроек, интерактивное изменение или выбор варианта отчета, то выполнение отчета будет прекращено, а табличный документ будет отображаться более светлыми цветами.

Если во время фонового выполнения будет закрыта форма отчета, то выполнение отчета будет прекращено автоматически, даже если в обработке события формы отчета [ПередЗакрытием](#) параметру [СтандартнаяОбработка](#) будет присвоено значение [Ложь](#). Остановка выполнения отчета (и соответствующего фонового задания) происходит после вызова обработчика события формы отчета [ПриЗакрытии](#).

Фоновое выполнение отчета может сопровождаться отображением окна состояния. Окно состояния будет формироваться, если это разрешено (свойство расширения формы отчета [РежимАвтоОтображенияСостояния](#)) и если отчет выполняется дольше 2 секунд (4 секунды в режиме низкой скорости соединения). Для отображения состояния отчета система компоновки данных использует свойство [ОтображениеСостояния](#) поля табличного документа. Если отчет сформировался за указанное время, то состояние не отображается и сразу отображается результат.

Проверка завершения формирования отчета выполняется клиентским приложением с определенной периодичностью. При этом интервал между каждой следующей проверкой в 1,4 раза длиннее предыдущего (но не длиннее 20 секунд).

Для управления возможностью отображения состояния отчета предназначено свойство расширения формы отчета [РежимАвтоОтображенияСостояния](#). Свойство определяет, каким образом табличный документ, который отображает результат работы отчета, будет отображать состояние отчета. По умолчанию (например, для автоматически генерируемой формы) значение данного свойства равно [Авто](#). Это означает, что окно состояния будет отображать информацию и о неактуальности отчета, и о том, что выполняется формирование отчета.

10.11. Особенности использования системы компоновки данных

- В системе компоновки данных запрещено использовать в качестве пути к данным имена, равные ключевым словам: [ВЫБОР](#), [КОГДА](#), [ТОГДА](#), [ИНАЧЕ](#), [КОНЕЦ](#), [ЕСТЬ](#), [НЕ](#), [ПОДОВНО](#), [СПЕЦИСИМВОЛ](#), [РАЗЛИЧНЫЕ](#).
- Если у параметра – стандартный период дата начала или дата окончания содержит пустую дату, считается, что вложенные параметры [ДатаНачала](#) и [ДатаОкончания](#) не установлены. То есть если у параметра [Период](#) дата начала содержит пустую дату, то параметр [Период.ДатаНачала](#) считается неустановленным. Аналогично для параметра [Период.ДатаОкончания](#) и даты окончания периода. Соответственно, неустановленными считаются параметры, в выражениях которых используются параметры [Период.ДатаНачала](#) и [Период.ДатаОкончания](#).
- Считается некорректным запрос, в котором указано ключевое слово [РАЗЛИЧНЫЕ](#) и в предложении [УПОРЯДОЧИТЬ ПО](#) указано выражение, отсутствующее в списке выборки.
- Запрещена группировка по полям-реквизитам полей-периодов.
- Если у пользователя отсутствует право на интерактивный просмотр объекта метаданных, то система компоновки считает недоступными все поля таблицы объекта.

Допустим, в качестве набора данных используется следующий запрос:

```
ВЫБРАТЬ
    Док.Ссылка.Дата,
    Док.Ссылка.Номер,
    Док.Номенклатура,
    Остатки.Остаток
ИЗ Документ.РасходнаяНакладная.Состав КАК Док
ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.УчетНоменклатуры.Остатки Остатки
ПО Док.Номенклатура = Остатки.Номенклатура
```

[Копировать в буфер обмена](#)

Если у пользователя нет интерактивных прав на таблицу [РасходнаяНакладная](#) или [РасходнаяНакладная.Состав](#), то ему не будут доступны поля [Дата](#), [Номер](#), [Номенклатура](#). Если нет прав на таблицу [РегистрНакопления.УчетНоменклатуры](#), то не будет доступно поле [Остаток](#). Если нет права ни на одну таблицу, то не будет доступно ни одно поле.

- При генерации макета компоновки данных из predetermined макетов удаляются выражения расшифровки, которые содержат поля, недоступные по функциональным опциям.
- Поле иерархического набора данных, с которым осуществляется связь, должно иметь имя, как у поля в наборе данных – источнике. В противном случае из иерархического набора данных не будут получаться поля-реквизиты.
- Если в запросе набора данных в списке выборки получается выражение, которое представляет собой применение функции `ПРЕДСТАВЛЕНИЕ()` к выражению, получаемому в другом поле, то при автоматическом заполнении доступных полей поле, в котором получается представление, не будет доступным для использования в настройках.
- При выполнении компоновки данных будет выдана ошибка, если выполняется группировка по вычисляемому полю, и в группировке указана необходимость дополнения в периоде.
- В макетах группировки системы компоновки данных автоотступ выполняется только для верхней левой ячейки макета.
- При использовании в системе компоновки данных пакетного запроса, поля, которые не используются в последующих запросах пакета, удаляются из списка выборки и полей группировки. Если запрос, формирующий временную таблицу, не содержит в себе ни одного поля – такой запрос удаляется из пакетного запроса. Также из результирующего макета компоновки данных исключаются наборы данных, из которых не используется ни одно поле. Чтобы исключить подобное удаление полей, следует либо использовать это поле или установить флажок **Обязательное** в роли поля схемы компоновки данных.
- Если поля временной таблицы, создаваемой в запросе пакетного запроса системы компоновки данных, не используются последующими запросами пакетного запроса, а также если создаваемая временная таблица не находится в обязательных соединениях, то запрос, создающий временную таблицу, удаляется из пакетного запроса макета компоновки данных.

Допустим, в качестве набора данных используется следующий запрос:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    Товар
ПОМЕСТИТЬ
    ВременнаяТаблица
ИЗ РегистрНакопления.Продажи
СГРУППИРОВАТЬ ПО Товар
;
ВЫБРАТЬ Склад, Товар
ИЗ РегистрНакопления.Остатки ЛЕВОЕ СОЕДИНЕНИЕ ВременнаяТаблица ПО ИСТИНА

```

Если в настройках компоновки данных не будет задействовано поле `Товар`, то первый запроса пакета удаляться не будет, т. к. таблица `ВременнаяТаблица` находится в обязательном соединении.

- При выводе результатов компоновки данных в коллекцию значений, поля различных группировок, ссылающихся на одно поле компоновки данных, попадают в одну колонку коллекции.
- При использовании в запросе набора данных системы компоновки данных с автоматическим заполнением доступных настроек конструкции `ОБЪЕДИНИТЬ` или `ОБЪЕДИНИТЬ ВСЕ`, и накладывании отбора на реквизит поля, отбор применяется только в тех частях объединения, в которых данный реквизит существует. При применении отбора к полю, отбор применяется только в тех частях, где выражение поля не `NULL`.
- При формировании списка доступных полей учитывается привилегированный режим работы в случае использования из встроенного языка. Если требуется, чтобы состав доступных полей учитывал права доступа пользователя, но при этом в результат компоновки попадали все данные, то рекомендуется выполнять компоновку макета компоновки данных в обычном (не привилегированном режиме), а саму компоновку данных выполнять в привилегированном режиме.
- Во время вычисления выражений системы компоновки данных происходит переход в безопасный режим (см. [здесь](#)).

Глава 11. Бухгалтерский учет

Ведение бухгалтерского учета в системе «1С:Предприятие» обеспечивают объекты конфигурации [Планы счетов](#) и [Регистры бухгалтерии](#). Так как эти объекты тесно связаны между собой, в данной главе дается общая характеристика возможностей, предоставляемых этими объектами.

В последующих разделах будут рассмотрены вопросы конфигурирования перечисленных объектов.

11.1. Обзор

Средства конфигуратора позволяют организовать учет по нескольким планам счетов, при этом для каждого плана счетов может строиться произвольная иерархия субсчетов большой вложенности. Бухгалтерские итоги хранятся системой «1С:Предприятие» отдельно для каждого плана счетов.

В планах счетов может применяться гибкая нумерация счетов с использованием цифр и букв.

Аналитический учет можно вести для любого счета или субсчета и включать различные разрезы аналитики. Объекты аналитического учета могут представлять собой элементы справочников системы «1С:Предприятие», документы или произвольные числа, даты и строки.

Средства конфигурирования позволяют вести неограниченное число видов учета для каждого плана счетов. Типичным примером видов учета служит количественный и валютный учет.

Включение вида учета выполняется при редактировании планов счетов. Для каждого счета можно указать, будет ли на этом счете вестись тот или иной вид учета.

Конфигуратор предоставляет средства для настройки ведения бухгалтерского учета для нескольких предприятий в одной информационной базе. Итоговую информацию можно получать как по отдельному предприятию, так и в целом, по всем предприятиям. Такая возможность будет полезна для предприятий с консолидированным балансом.

11.2. Планы счетов

План счетов является одним из основных понятий бухгалтерского учета. Планом счетов называется совокупность синтетических счетов, предназначенных для группировки информации о хозяйственной деятельности предприятия. Информация, накапливаемая на таких синтетических счетах, позволяет получить полную картину состояния средств предприятия в денежном выражении.

Система «1С:Предприятие» предоставляет гибкие возможности по ведению планов счетов. Собственно путем настройки плана счетов и организуется требуемая система учета.

Несколько планов счетов. В системе «1С:Предприятие» может быть несколько планов счетов и учет по всем планам счетов можно вести одновременно. Общее число планов счетов, которое может быть организовано в системе, с технической точки зрения неограниченно и определяется исключительно реальными потребностями учета.

Например, такой «многоплановый» учет, очевидно, понадобится для совместных предприятий, которым требуется вести учет одновременно по двум или более стандартам бухгалтерского учета.

Субсчета. Планы счетов в системе «1С:Предприятие» поддерживают многоуровневую иерархию «счет – субсчета». Каждый план счетов может включать неограниченное число счетов первого уровня. К каждому счету может быть открыто также неограниченное количество субсчетов. В свою очередь, каждый субсчет может иметь свои субсчета и так далее. Количество уровней субсчетов в системе «1С:Предприятие» неограниченно.

Структура кода счета может быть задана при создании плана счетов в виде шаблона, состоящего из произвольной последовательности символов. Технически структура кода счета не влияет на иерархию счетов, однако при создании структуры счетов рекомендуется придерживаться структуры кодов.

Разделенное редактирование счетов. Известно, что зачастую ошибки в бухгалтерском учете возникают из-за неправильного применения утвержденного плана счетов. Наиболее распространенная ситуация – отсутствие аналитического учета на тех счетах, по которым необходимо получать

развернутое сальдо. Тогда бухгалтерский учет, являясь верным по форме (все «сходится»), по сути дает неправильные результаты. Поэтому зачастую бывает полезно ограничить творчество конечных пользователей в части настройки планов счетов.

С этой целью система «1С:Предприятие» позволяет разделять процесс редактирования плана счетов.

Во-первых, редактирование планов счетов может выполняться в конфигураторе специалистом, выполняющим конфигурирование системы «1С:Предприятие».

В конфигураторе могут создаваться новые планы счетов. При конфигурировании планов счетов задаются их основные характеристики: длина кода счета и наименования счета; максимальное количество субконто, которое может быть у одного счета (субсчета), и другие. Здесь же в планы счетов могут быть введены требуемые счета и субсчета (предопределенные счета), а также выполнена настройка видов учета на счетах.

Во-вторых, при работе с системой «1С:Предприятие» конечный пользователь может добавлять собственные счета и субсчета в планы счетов, однако он не сможет удалять предопределенные счета и субсчета, созданные в конфигураторе.

Хранение бухгалтерских итогов. Хранение бухгалтерских итогов осуществляется в регистрах бухгалтерии в соответствии со структурой плана счетов. Итоговая информация может быть просмотрена в формах регистра, а также извлекается средствами встроенного языка системы «1С:Предприятие». Для этого в языке существуют методы получения остатков и оборотов по счетам по любому виду учета, как в целом по счету или субсчету, так и с разбивкой по объектам аналитического учета.

11.3. Аналитический учет

Информация о средствах предприятия, которая накапливается на счетах бухгалтерского учета, зачастую имеет обобщающий характер. Например, в «стандартном» плане счетов существует счет **10 «Материалы»**, который предназначен «...для обобщения информации о наличии и движении принадлежащих предприятию сырья, материалов, топлива, запасных частей, тары и т. п. ценностей». К счету 10 могут быть открыты субсчета для учета различных видов материалов. Но при такой организации учета на субсчетах будет накапливаться информация об общей стоимости материалов одного вида, а в целом на счете 10 – стоимость всех материалов.

Для получения детальной информации о наличии конкретных материалов необходима организация аналитического учета по материалам. В этом случае общие суммы на субсчетах разбиваются на более мелкие – стоимости конкретных материалов.

В системе «1С:Предприятие» аналитический учет организуется при помощи специального механизма субконто.

Понятие «субконто». Субконто в системе «1С:Предприятие» называется объект аналитического учета. Термином «субконто» могут быть обозначены любые объекты аналитического учета: основные средства, нематериальные активы, материалы, организации, подотчетные лица, договоры, бюджеты. Видом субконто, в свою очередь, называется множество однотипных объектов аналитического учета.

Например, учет задолженности предприятия перед покупателями и заказчиками обязательно ведется согласно нормативным документам, отдельно по каждому покупателю и заказчику. В системе «1С:Предприятие» такой список покупателей и заказчиков (предположим, что это только организации) будет называться «видом субконто «Организации»», а любая организация из этого списка будет именоваться «субконто».

Конфигуратор системы «1С:Предприятие» позволяет создать любое количество видов субконто в соответствии с требованиями полноты аналитического учета на предприятии.

В качестве универсальных средств описаний свойств объектов аналитического учета используются объекты типа [План видов характеристик](#).

ВНИМАНИЕ! Для плана видов характеристик, используемого в качестве видов субконто плана счетов, **не рекомендуется** использовать примитивные типы. Это может существенно сказаться на производительности при записи движений регистра бухгалтерии. Рекомендуется использовать только ссылочные типы.

Организация аналитического учета. В системе «1С:Предприятие» аналитический учет можно вести по любому счету или субсчету. Для этого при редактировании планов счетов к требуемому счету или субсчету «прикрепляется» нужный вид субконто. К счету или субсчету можно прикрепить несколько видов разных субконто (их максимальное число определяется в свойстве [Макс. количество субконто](#)) и таким образом организовать ведение аналитического учета в любых необходимых разрезах.

Ввод информации о хозяйственных операциях в систему «1С:Предприятие» выполняется в виде записей регистров бухгалтерии. Для каждого из корреспондирующих счетов записи (если для этих счетов указано ведение аналитического учета) необходимо указать объекты аналитического учета.

Например, на производственных предприятиях к счету учета затрат на производство можно прикрепить виды субконто: [Виды затрат](#) для учета по видам затрат, [Продукция](#) для учета по видам выпускаемой продукции (работ, услуг) и [Подразделения](#) – для учета по подразделениям предприятия. Аналитическую информацию по затратам можно получать по любому из этих видов субконто.

Кроме этого, возможности ведения аналитического учета в системе «1С:Предприятие» позволяют вести учет одного и того же объекта аналитики в разных разрезах.

Так, один и тот же справочник может быть назначен разными видами субконто. Например, организация закупает комплектующие у поставщиков, а продает готовую продукцию покупателям. В этом случае в системе «1С:Предприятие» справочник [Организации](#) можно назначить видом субконто [Поставщики](#) и видом субконто [Покупатели](#).

Субконто и субсчета. Свойства планов счетов в системе «1С:Предприятие» (в частности, большое число уровней вложенности субсчетов или большая длина кода субсчета), тем не менее, позволяют организовать ведение аналитического учета с использованием субсчетов, а не субконто. Однако возможности, которые предоставляет ведение аналитического учета на субсчетах и с использованием субконто, принципиально различаются.

Субсчета представляют собой иерархическую структуру, подчиненную конкретному счету синтетического учета. Например, для учета задолженности предприятию со стороны покупателей на синтетическом счете учета расчетов с покупателями (в стандартном плане счетов это счет 62) для каждого покупателя можно открыть отдельный субсчет и вести учет задолженности на этом субсчете. Такая схема позволит получить информацию как о сумме задолженности предприятия перед конкретным покупателем, так и общую сумму задолженности перед всеми покупателями – она получится путем суммирования информации о суммах задолженности со всех субсчетов.

Однако если какая-либо из организаций-покупателей становится также и поставщиком, то потребуются также организовать учет расчетов с этой организацией как с поставщиком. При ведении аналитического учета на субсчетах для нового поставщика, очевидно, потребуются открыть новый субсчет уже на счете учета расчетов с поставщиками (счет 60). Теперь, если вести учет по этому субсчету, можно получить информацию о взаиморасчетах с организацией как с поставщиком.

Но чтобы получить данные об общем состоянии взаиморасчетов с организацией, необходимо объединить информацию о расчетах с этой организацией с двух счетов. Для этого нужно помнить, какой субсчет на счете учета расчетов с покупателями и какой субсчет на счете учета расчетов с поставщиками соответствуют конкретной организации, получить информацию о состоянии расчетов именно с этих субсчетов и каким-то образом ее обработать.

При использовании субконто один и тот же список субконто, например [Организации](#), прикрепляется ко всем счетам (субсчетам), на которых предполагается вести аналитический учет по организациям. Учет с конкретной организацией как с поставщиком ведется на соответствующем синтетическом счете. Если организация становится еще и покупателем, нет необходимости создавать новую позицию в списке организаций – она уже присутствует в этом списке как поставщик. Объединение информации для одной организации с двух счетов синтетических счетов в значительной мере выполняется системой «1С:Предприятие» автоматически. Аналогичным образом можно объединять данные с любых счетов, на которых велся учет операций с участием конкретной организации.

Многоуровневая аналитика. Многоуровневая аналитика позволяет получать бухгалтерские итоги с разной степенью детализации. Если для ведения аналитического учета используются субконто, то для реализации многоуровневой аналитики необходимо назначать видами субконто план видов характеристик (см. [здесь](#)). Использование объектов [Планы видов характеристик](#) позволяет вести учет с необходимой степенью детализации. Ограничение по числу уровней вложенности накладывает использование типа значения характеристики (например, в качестве типа значения может быть выбран

любой иерархический справочник, а также допускается использование составного типа).

При ведении аналитического учета с использованием субсчетов многоуровневый аналитический учет реализуется за счет использования субсчетов разных уровней.

11.4. Виды учета

Как уже говорилось выше, система «1С:Предприятие» позволяет вести неограниченное число видов учета. Примерами видов учета может служить количественный и валютный учет.

Конфигуратор предоставляет возможность настроить неограниченное количество разнообразных учетов. Виды учета определяются подчиненными объектами типа [Признак учета](#) объекта [План счетов](#).

11.5. Записи и наборы записей регистров бухгалтерии

Основным понятием бухгалтерского учета является понятие **хозяйственная операция**. Операцией считается любое хозяйственное действие, способное вызывать изменение состояния средств предприятия.

Для ввода информации о хозяйственных операциях в системе «1С:Предприятие» используются документы. Документ позволяет занести в систему информацию о хозяйственной операции, зафиксировать дату и время совершения операций, сумму и содержание операции.

Учетные данные по операции автоматически формируются на основе документа (объекта данных типа [Документ](#)) и фиксируются в регистрах бухгалтерии. Порядок такой генерации определяется в конфигураторе средствами встроенного языка. При записи данных по операции можно описать заполнение реквизитов операции различной информацией из документа, породившего эту операцию.

В системе «1С:Предприятие» учет хозяйственной операции всегда связывается с породившим ее документом: если документ необходимо отредактировать, то при его проведении данные по операции будут сформированы заново; при удалении документа будут удалены и данные хозяйственной операции.

Каждая хозяйственная операция состоит в общем случае из произвольного числа проводок. В системе «1С:Предприятие» каждой бухгалтерской проводке соответствует запись регистра бухгалтерии, а набор записей регистра бухгалтерии является аналогом хозяйственной операции. Описание процедуры работы с проводками см. [здесь](#).

11.6. Ведение консолидированного учета

В соответствии с действующим законодательством организации, имеющие структурные подразделения, в том числе и выделенные на отдельный баланс (филиалы, представительства), составляют отчетность с учетом деятельности всех структурных подразделений. Для таких организаций важным моментом будет являться возможность вести в одной информационной базе бухгалтерский учет всех структурных подразделений и получать консолидированную отчетность. При составлении бухгалтерской отчетности, кроме того, целесообразно отдельно представлять основные показатели деятельности структурных подразделений во избежание лишних вопросов со стороны налоговых органов.

Инструментальные средства для ведения бухгалтерского учета системы «1С:Предприятие» позволяют вести в одной информационной базе бухгалтерию нескольких предприятий (или структурных подразделений одного предприятия) одновременно. Решение подобных задач осуществляется за счет формирования измерений регистра бухгалтерии, в котором будет указываться ссылка на структурное подразделение.

Информация, которая вводится в это измерение при работе с системой «1С:Предприятие», используется в качестве признака для раздельного хранения бухгалтерских итогов. Программа «1С:Предприятие» автоматически организует систему хранения бухгалтерских итогов, разделяя их по значению указанного измерения. При помощи встроенного языка информацию из итогов можно извлекать и по конкретному предприятию, и в целом по всем предприятиям.

11.7. Создание плана счетов

Для ведения планов счетов в системе «1С:Предприятие» используются объекты конфигурации [План](#)

счетов. Объектами данных этого типа являются бухгалтерские счета – учетные регистры, по которым будет выполняться группировка средств при работе с системой «1С:Предприятие». Конфигуратор позволяет создавать практически неограниченное количество планов счетов. Все созданные в конфигураторе планы счетов можно использовать одновременно.

Как описывалось выше, планы счетов в системе «1С:Предприятие» поддерживают многоуровневую иерархию «**счет – субсчета**».

11.7.1. Свойства планов счетов

Редактирование свойств объектов типа **План счетов** и создание подчиненных объектов выполняются в окне редактирования (см. [здесь](#)), а также в палитре свойств. В этом разделе будут описаны специфические свойства объекта конфигурации типа **План счетов**.

Маска кода – используется для описания структуры кода счетов и субсчетов. В строке маски допустимо использование следующих специальных символов:

- «!» – любой введенный символ преобразуется в верхний регистр;
- «9» – допустимо ввести произвольный символ цифры;
- «#» – допустимо ввести произвольный символ цифры, или «-» (знак минус), или «+» (знак плюс), или пробел;
- «N» – допустимо ввести любые алфавитно-цифровые символы (буквы или цифры);
- «U» – допустимо ввести любые алфавитно-цифровые символы (буквы или цифры), и любой введенный символ преобразуется в верхний регистр;
- «X» (латинского алфавита) – допустимо ввести произвольный символ;
- «@» – допустимо ввести любые алфавитно-цифровые символы (буквы или цифры) в верхнем регистре или пробел;
- «» – недопустимо вводить этот символ интерактивно пользователем, он может устанавливаться только из языка;
- «h» – допустим ввод символов обозначения шестнадцатеричных цифр.

Для того чтобы использовать в маске один из специальных символов, нужно поставить перед ним символ «\».

Если маска кода содержит точки или не содержит запятых, то при вводе кода счета все точки будут автоматически заменяться запятыми.

Автопорядок по коду – если свойство установлено, то вместо упорядочивания по коду счета будет использоваться упорядочивание по полю **Порядок**.

Длина порядка – если длина больше нуля, то поле **Порядок** используется по умолчанию для упорядочивания плана счетов.

ПРИМЕЧАНИЕ. Максимальная длина реквизитов **Длина кода**, **Длина наименования** и **Длина порядка** равна 628.

Реквизиты – задаются различные описания счета или субсчета, например, наименование счета 10 «Материалы».

Признаки учета – задается список видов учета. Типовым примером служит валютный и количественный учет. Разработчики конфигурации могут создавать столько видов учета, сколько им требуется для реализации конкретной задачи.

Табличные части – для «стандартных» планов счетов не используются табличные части. Примером

использования подчиненных объектов типа [Табличная часть](#) может служить организация одновременного учета данных в разных планах счетов для обеспечения развернутой корреспонденции счетов различных планов счетов.

[Максимальное количество субконто](#) – определяет максимальное количество субконто (до 50), используемых для организации аналитического учета.

[Виды субконто](#) – указывается объект типа [План видов характеристик](#) (см. [здесь](#)).

[Признаки учета субконто](#) – создается список признаков учета субконто. При настройке аналитического учета по счетам данные признаки позволяют установить признак использования того или иного вида субконто при формировании predeterminedенных счетов.

11.7.2. Организация различных видов учета

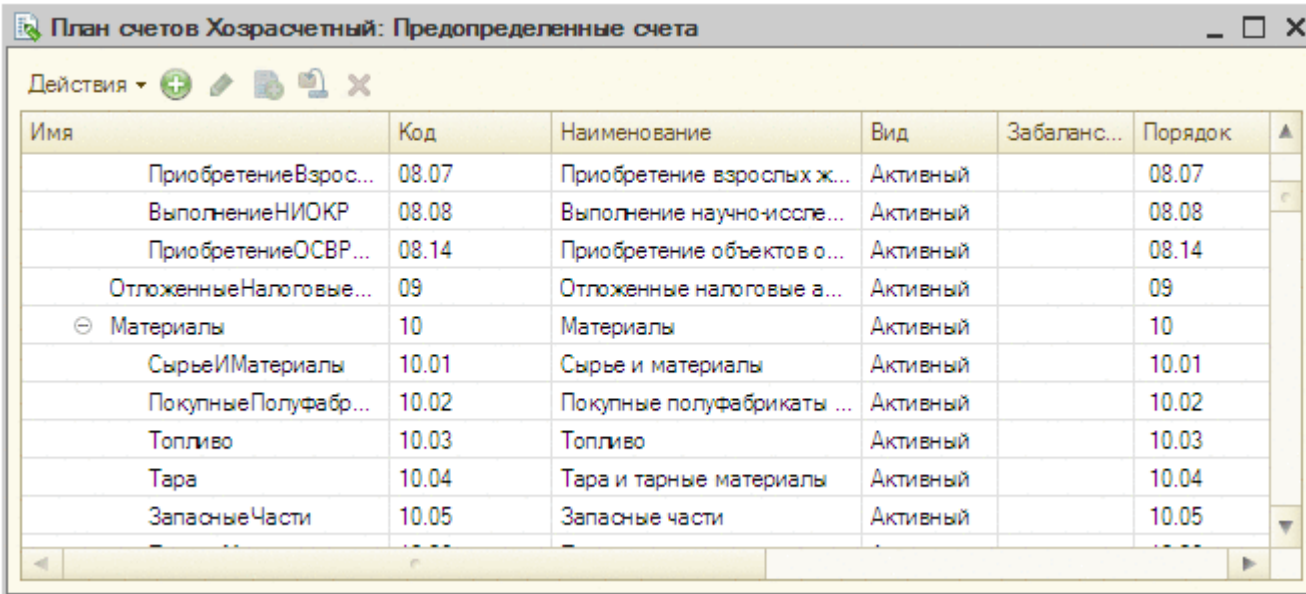
Для каждого плана счетов можно установить произвольное число видов учета.

Для каждого счета или субсчета при их создании необходимо указать признак использования каждого определенного для данного плана счетов вида учета.

11.7.3. Формирование списка счетов (предeterminedенные счета)

Бухгалтерские счета как объекты данных создаются в конфигураторе.

Для создания списка predeterminedенных счетов в палитре свойств объекта типа [План счетов](#) в свойстве [Предeterminedенные](#) нужно щелкнуть ссылку [Открыть](#). На экран выводится окно списка счетов.



Имя	Код	Наименование	Вид	Забаланс...	Порядок
ПриобретениеВзрос...	08.07	Приобретение взрослых ж...	Активный		08.07
ВыполнениеНИОКР	08.08	Выполнение научно-иссле...	Активный		08.08
ПриобретениеОСВР...	08.14	Приобретение объектов о...	Активный		08.14
ОтложенныеНалоговые...	09	Отложенные налоговые а...	Активный		09
⊖ Материалы	10	Материалы	Активный		10
СырьеИМатериалы	10.01	Сырье и материалы	Активный		10.01
ПокупныеПолуфабр...	10.02	Покупные полуфабрикаты ...	Активный		10.02
Топливо	10.03	Топливо	Активный		10.03
Тара	10.04	Тара и тарные материалы	Активный		10.04
ЗапасныеЧасти	10.05	Запасные части	Активный		10.05

Рис. 292. План счетов

Ведение списка счетов осуществляется командами меню [Действия](#).

Данные по счету формируются в окне редактирования счета.

Предопределенный счет

Родитель:

Имя:

Код:

Наименование:

Вид:

Забалансовый

Порядок:

Признак учета	Учитывать
Валютный	<input type="checkbox"/>
Количественный	<input type="checkbox"/>
НалоговыйУчет	<input checked="" type="checkbox"/>

Вид субконто	Только обороты	Суммовой	Валютный	Количественный
ОбъектыСтроит...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
СтатьиЗатрат	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
СпособыСтроит...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рис. 293. Редактирование предопределенного счета

В реквизите **Вид** указывается вид счета: **Активный**, **Пассивный** или **Активный/Пассивный**.

Для забалансовых счетов нужно установить флажок **Забалансовый**. Итоговые данные забалансовых счетов не учитываются при подведении баланса.

Поле **Порядок** предназначено для произвольного упорядочивания счетов. Длина поля задается в свойстве **Длина** порядка. Если длина поля больше нуля, то данное поле используется по умолчанию для упорядочивания плана счетов. Если у плана счетов установлено свойство **Автопорядок по коду**, то вместо упорядочивания по коду счета или наименованию будет использоваться упорядочивание по полю **Порядок**. Для заполнения поля **Порядок** по коду счета используется метод `ПолучитьПорядокКода()`, формирующий строку порядка счета с учетом маски кода счета.

В первом списке перечислены признаки учета, определенные для объекта **План счетов**. Для указания того, что редактируемый счет принадлежит определенному виду учета, следует установить флажок в колонке **Учитывать**.

По каждой записи нижнего списка указываются следующие параметры. В колонке **Вид субконто** выбирается один из предопределенных видов характеристик. Если для данного плана счетов в ветви **Признаки учета субконто** определены подчиненные объекты, то в список добавляются колонки по числу этих объектов. Для каждого вида субконто следует установить флажок, если по данному виду субконто требуется вести аналитический учет. При этом следует учитывать следующую особенность: если по счету в целом учет по какому-либо признаку учета не ведется, то соответствующий признак учета по субконто должен быть всегда установлен, в противном случае признак учета по субконто следует выключать в соответствии с потребностями учета.

В режиме 1С:Предприятие конечный пользователь может добавлять собственные счета и субсчета в планы счетов и редактировать свойства счетов, однако он не сможет удалить счета и субсчета, созданные в конфигураторе. Для счетов, созданных в конфигураторе, в режиме 1С:Предприятие можно изменять только следующие реквизиты:

- родитель счета,
- код счета,

- наименование счета,
- порядок.

11.8. Регистры бухгалтерии

Для отражения в бухгалтерском учете информации о хозяйственных операциях в системе «1С:Предприятие» используются регистры бухгалтерии, описываемые в ветви дерева конфигурации [Регистры бухгалтерии](#).

11.8.1. Свойства регистра бухгалтерии

По своему виду регистр бухгалтерии напоминает регистр накопления. При его редактировании разрабатывается структура регистра: создаются наборы измерений, ресурсов и реквизитов регистра; если необходимо, создаются экранные и печатные формы просмотра движений регистра (см. [здесь](#)).

Отличительной особенностью регистра бухгалтерии является его связь с планом счетов и поддержка механизма двойной записи. Каждая запись регистра содержит обязательные реквизиты [Счет Дт](#) (счет дебета) и [Счет Кт](#) (счет кредита) для регистров, поддерживающих корреспонденцию, и реквизит [Счет](#) для не поддерживающих.

Свойство [Разрешить разделение итогов](#) позволяет задействовать механизм разделителя итогов, который обеспечивает более высокую параллельность работы при записи в регистр (описание аналогичного свойства для регистров накопления см. [здесь](#)).

ПРИМЕЧАНИЕ. В качестве типа измерения регистра бухгалтерии не могут выступать значения типа [УникальныйИдентификатор](#), [ДвоичныеДанные](#), строка неограниченной длины.

11.8.2. Записи регистра бухгалтерии

Чтобы введенная документом хозяйственная операция вызвала изменение в бухгалтерских итогах, такой документ должен формировать записи регистра бухгалтерии.

Записи конкретного документа составляют группу (по документу-регистратору) и при любых изменениях в реквизитах документа или в самих записях всегда «держатся» вместе. Порядок генерации записей также определяется средствами встроеного языка.

Структура записи строится системой «1С:Предприятие» динамически в зависимости от настройки различных элементов учета, выполненной при редактировании планов счетов. Так, например, запись может содержать реквизиты для ввода корреспондирующих счетов, сумм, объектов аналитического учета, количества, вида валюты и суммы в валюте.

Помимо этих реквизитов в конфигураторе можно создать для записи необходимое число дополнительных реквизитов для отражения любой другой необходимой информации. Например, это может быть реквизит для хранения комментария записи.

При работе с системой «1С:Предприятие» пользователь может «отключать» записи и затем вновь «включать» их. «Отключенные» записи не влияют на бухгалтерские итоги – остатки и обороты по счетам. Отключение и включение записей производятся с помощью свойства [Активность](#).

Такая возможность удобна для учета плановых хозяйственных операций – тех, что должны произойти в будущем. Отключив записи такой операции, можно не изменять реальные бухгалтерские итоги. В момент действительного совершения хозяйственного действия достаточно установить для этих записей активность.

При вводе записей система «1С:Предприятие» выполняет различные предопределенные действия, облегчающие и ускоряющие работу пользователя с системой. Например, если по какому-либо корреспондирующему счету указано ведение аналитического учета, то система автоматически откроет требуемый список объектов аналитического учета (субконто) для того, чтобы пользователь выбрал нужный объект. Многие аспекты поведения системы «1С:Предприятие» могут быть настроены путем редактирования свойств записей регистра бухгалтерии.

Глава 12. Периодические расчеты

Периодические расчеты – это вычисления, осуществляемые с определенной периодичностью, тесно связанные друг с другом по некоторым правилам и взаимно влияющие друг на друга в пределах некоторого периода.

Механизмы периодических расчетов позволяют настроить порядок и взаимосвязь вычислений и организовать учет их результатов.

Наиболее типичным примером использования периодических расчетов является расчет заработной платы, при котором производятся расчеты начислений и удержаний. Расчеты обычно выполняются с месячной периодичностью, а результаты расчетов одного вида могут зависеть от наличия и результатов расчетов другого вида.

Ведение периодических расчетов в системе «1С:Предприятие» обеспечивают объекты конфигурации **Планы видов расчета** и **Регистры расчета**. Так как эти объекты тесно связаны между собой, в данной главе дается общая характеристика возможностей, предоставляемых этими объектами.

В последующих разделах будут рассмотрены вопросы конфигурирования перечисленных объектов.

12.1. Основные понятия

Период. Для расчетов важным является понятие периода. Обычно период описывается датой начала и датой окончания. Если для расчета определена периодичность (см. описание ниже), то для описания периода (действия, регистрации) данного расчета достаточно указать любую дату. По этой дате вычисляется дата начала периода, и именно эта дата будет описывать период. Такой порядок определения периода позволяет оптимизировать выполнение запросов, в которых требуется выбрать записи, относящиеся к указанному периоду.

Периодичность расчетов. Определяет, с каким периодом будут (могут) выполняться расчеты, учитываемые данным регистром. Задается в свойстве **Периодичность** регистра расчетов. По значению этого свойства (если регистр периодический) определяется период действия записи регистра расчета. Например, регистр имеет периодичность **Месяц**, тогда при формировании записи регистра в качестве периода действия выбирается дата документа (например, некий расчет за ноябрь 2008 г.), и по ней система определяет период действия на начало 01.11.2008.

Период регистрации – дата начала периода, указанного при регистрации расчета (вычисляется по дате документа-регистратора). Например, в июне 2008 г. производится начисление оклада за май 2008 г. (расчеты с периодичностью **Месяц**). Май 2008 г. – период действия (в базе данных записывается дата 01.05.2008), а июнь 2008 г. – период регистрации (в базе данных записывается дата 01.06.2008).

Период действия – начальная дата периода, определяемая в соответствии со значением свойства **Периодичность**. Например, в документе указывается, что расчет производится за май 2008 г. Для значения свойства **Периодичность** регистра **Месяц** период действия определяется датой 01.05.2008; для регистра **Квартал** – 01.04.2008.

Период действия расчета – указывает период, за который производится расчет. Период определяется датой начала и датой окончания периода. Например, запись о больничном листе за май 2008 г. имеет период действия 01.05.2008, а период действия расчета определяется датой начала (например, 06.05.2008) и датой окончания (например, 15.05.2008).

Механизм вытеснения или конкуренция за период действия расчета – проявление связи видов расчетов по периоду действия расчетов. Конкуренция возникает вследствие невозможности выполнения нескольких видов расчетов одновременно. Необходимо выбрать тот расчет, который в данном периоде будет выполнен. Настройка механизма вытеснения задается в описании конкретного вида расчета. Такая настройка выполняется в разделе **Вытесняющие** (виды расчетов). Например, расчет **Оплата по окладу** не может применяться одновременно с расчетом **Оплата по больничному листу**. При этом говорят, что расчет больничного вытесняет расчет оклада, т. е. за период, в котором «действует» больничный, оклад не начисляется. Описание механизма вытеснения см. [здесь](#).

Фактический период действия – если расчет не вытесняется другими расчетами, то фактический период совпадает с периодом действия. Если есть вытесняющие виды расчетов, то фактический период

определяется как совокупность непересекающихся периодов, в которых данный расчет не вытеснялся. На [рис. 294](#) приводится графическое представление фактического периода действия в случае вытеснения расчетом [Больничный](#) расчета [Оклад](#).

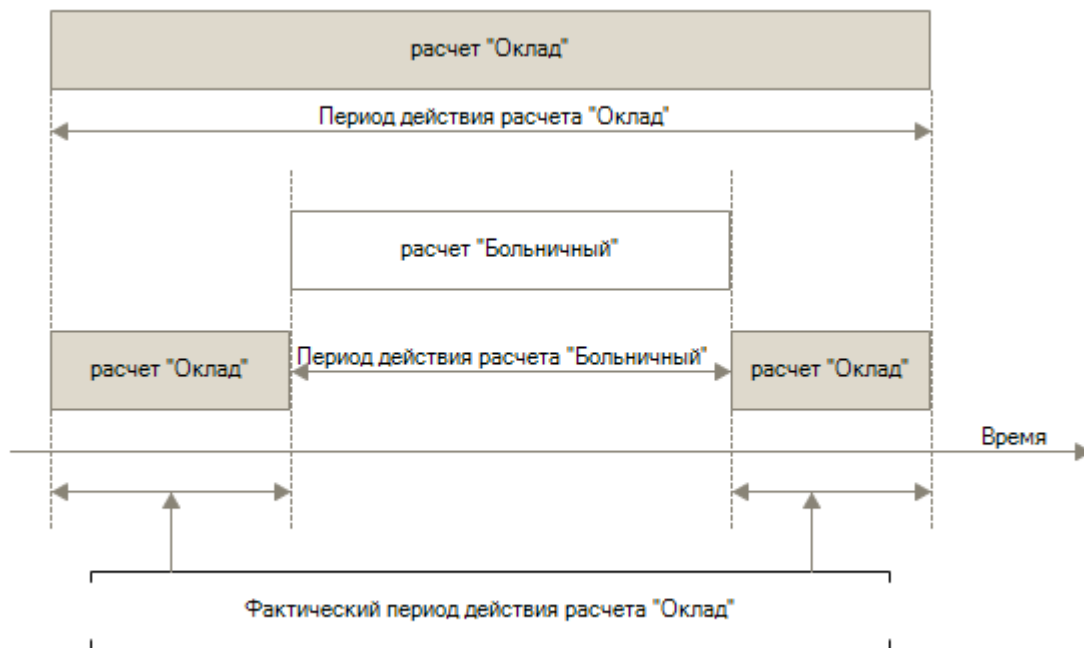


Рис. 294. Механизм вытеснения

Фактический период действия представляет собой вторичные данные, т. е. результат вычислений, выполняемых системой. Пересчет фактического периода действия выполняется при любой записи набора, в том числе пустого (отмена проведения).

ВНИМАНИЕ! Результат расчетов не зависит от последовательности ввода (регистрации) данных.

Базовый период – определяет период, за который будут выбираться результаты расчетов, используемых (являющихся базовыми) для данного расчета. Например, при начислении премии за май 2008 г. учитываются результаты начислений, выполненных определенными видами расчетов за некоторый период (это могут быть [Оклад](#), [Доплата](#), [Отпуск](#)). Этот период и будет являться базовым для расчета премии. Настройка связи видов расчетов по базовому периоду производится в описании конкретного вида расчета в разделе [Базовые](#) (виды расчетов).

12.2. Планы видов расчета

Объекты конфигурации данного вида предназначены для создания видов расчетов, используемых в регистрах расчета.

Ссылка на вид расчета – одно из основных свойств записей регистра расчета, которое позволяет придать качественные различия разным учетным записям регистра расчета.

Редактирование свойств объектов типа [План видов расчета](#) и создание подчиненных объектов выполняются в окне редактирования (см. [здесь](#)), а также в палитре свойств. Далее будут описаны уникальные свойства плана видов расчета в дополнение к общим свойствам.

12.2.1. Категория свойств «Расчет»

Зависимость от базы – если свойство [Зависимость от базы](#) отличается от значения [Не зависит](#), то видам расчета такого плана видов расчета можно задать список базовых видов расчета. Состав базовых видов расчета определяется свойством [Базовые планы видов расчета](#).

Базовые планы видов расчета – задается список планов видов расчета, виды расчета которых могут включаться в список базовых видов расчета. Этот список базовых видов расчета используется регистром

расчета при работе механизма получения базы.

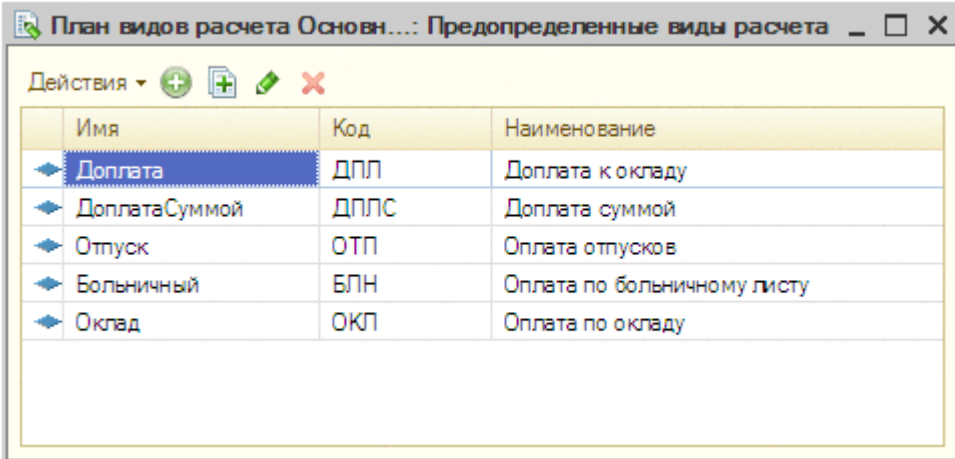
Использует период действия – если свойство установлено, то данный план видов расчета может быть назначен регистру расчета с периодом действия. При этом для каждого вида расчета может быть задан список вытесняющих видов расчета. Список вытесняющих видов расчета определяет работу механизма вытеснения соответствующего регистра расчета (см. [здесь](#)).

Указанные в данной категории свойства определяют состав реквизитов, включаемых в список при создании форм вида расчета. Если выбрано значение свойства **Зависимость от базы – Зависит от периода действия (регистрации)**, то в форму можно поместить табличное поле для выбора тех видов расчетов, которые являются базовыми для редактируемого расчета. Если свойство **Использует период действия** установлено, то в форму можно поместить табличное поле для выбора видов расчетов, которые являются вытесняющими для редактируемого вида расчета.

12.2.2. Предопределенные виды расчета

Особенностью плана видов расчета является наличие предопределенных видов расчета, сформированных на этапе конфигурирования. В режиме 1С:Предприятие пользователь не может их удалить.

Для редактирования списка предопределенных видов расчета нужно перейти в палитре свойств на закладку **Прочее** и щелкнуть по кнопке **Предопределенные**. На экран выводится список предопределенных видов расчета данного плана видов расчета (пример см. на [рис. 295](#)).



План видов расчета Основн...: Предопределенные виды расчета

Действия + + -

Имя	Код	Наименование
Доплата	ДПЛ	Доплата к окладу
ДоплатаСуммой	ДПДС	Доплата суммой
Отпуск	ОТП	Оплата отпусков
Больничный	БЛН	Оплата по больничному листу
Оклад	ОКЛ	Оплата по окладу

Рис. 295. Предопределенные планы видов расчета

Редактирование списка предопределенных видов расчета производится с помощью пунктов меню **Действия**.

Состав реквизитов (базовые и ведущие) предопределенных видов расчета зависит от установленных свойств данного объекта плана видов расчета, размещенных в категории **Расчет**.

Каждый предопределенный вид расчета редактируется в модальном окне. На закладке **Основное** вводятся основные свойства вида расчета.

Вид расчета

Основное Базовые Вытесняющие Ведущие

Имя:

Код:

Наименование:

Период действия является базовым периодом

OK Отмена Справка

Рис. 296. Редактирование плана видов расчета

Если выбрано значение свойства **Зависимость от базы** – **Зависит по периоду действия (регистрации)**, то форма имеет закладку **Базовые**. На ней указываются те виды расчета, которые являются базовыми для данного вида расчета и определяют работу механизма получения базы соответствующего регистра расчета.

Вид расчета

Основное Базовые Ведущие

Основные Начисления

- Доплата
- ДоплатаСуммой
- Отпуск
- Больничный
- Оклад

OK Отмена Справка

Рис. 297. Базовые начисления вида расчета

Так, в приведенном примере (см. [рис. 297](#)) для вида расчета **Больничный** базовыми видами расчета являются **Оклад** и **Доплата**.

Если свойство **Использует период действия** установлено, то форма имеет закладку **Вытесняющие**. На ней указываются те виды расчета, которые вытесняют данный вид расчета по периоду действия и таким образом настраивают работу механизма вытеснения соответствующего регистра расчета.

Вид расчета

Основное Базовые Вытесняющие Ведущие

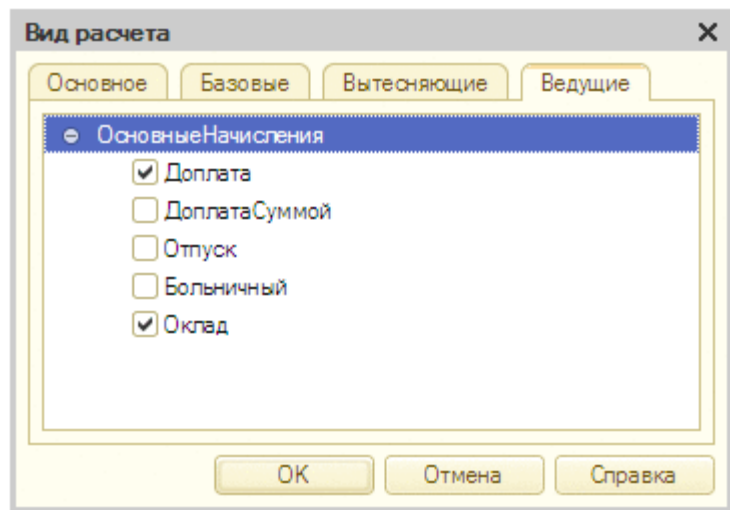
- Доплата
- ДоплатаСуммой
- Отпуск
- Оклад

OK Отмена Справка

Рис. 298. Вытесняющие виды расчетов

Из перечисленных в списке расчетов (см. [рис. 298](#)) следует указать вытесняющий вид расчета. В приведенном примере для вида расчета **Больничный** таких нет. Но при описании расчета **Оклад** следует отметить **Больничный**.

На закладке **Ведущие** указываются те виды расчета, которые определяют работу механизма перерасчетов соответствующего регистра расчета (т. е. того, которому назначен данный план видов расчета).

**Рис. 299. Ведущие виды расчета**

Помимо predetermined видов расчета пользователь в режиме 1С:Предприятие может создавать собственные виды расчета и устанавливать им нужные свойства. Для этих целей в конфигураторе необходимо создать форму элемента, в которой следует разместить табличные поля для указания базовых, ведущих и вытесняемых видов расчета в зависимости от установленных свойств данного плана видов расчета.

12.3. Регистры расчета

12.3.1. Общая информация о регистрах расчета

Регистр расчета – это объект конфигурации, который позволяет организовать учет результатов вычислений, выполненных расчетами, указанными в плане видов расчетов. Регистры расчета позволяют, например, реализовать регистрацию начислений в пользу физических лиц (оплата труда, оплата больничных листов, оплата дней отпуска и т. д.).

Учетные записи регистра расчета могут влиять на состояние других записей регистра. Различают два характера взаимного влияния учетных записей регистра расчета: конкуренция за период действия и зависимость по базовому периоду. Поясним эти способы взаимного влияния учетных записей на примерах.

12.3.1.1. Механизм вытеснения

Конкуренцию за период действия можно пояснить на примере начислений в пользу физического лица оплаты по основной форме оплаты труда (по окладу) и оплаты дней отпуска. Учетная запись, соответствующая оплате по окладу, и учетная запись, соответствующая оплате отпуска, не могут соответствовать одному и тому же промежутку времени, т. е. промежутки времени, за которые произведены такие начисления, не могут пересекаться. В регистре расчета такое поведение учетных записей (движений регистра) реализовано за счет использования понятия период действия записи регистра. Каждая запись регистра расчета имеет фактический период действия, задаваемый в общем случае набором из нескольких промежутков времени.

Так, если учетная запись, соответствующая отпуску, имеет период действия с 03.03.2008 по 13.03.2008, то учетная запись, соответствующая начислению по окладу, может, например, иметь фактический период действия, состоящий из двух промежутков: с 01.03.2008 по 02.03.2008 и с 14.03.2008 по 20.03.2008.

Конкуренция за период действия поддерживается механизмом вытеснения регистра расчета, работа которого определяется составом вытесняющих видов расчета, датами периода действия и т. д.

12.3.1.2. Механизм перерасчетов

Зависимость учетных записей по базовому периоду действия можно рассмотреть на примере начислений в пользу физического лица оплаты по окладу и зависящего от него начисления по среднему заработку. Если в регистре происходит изменение состояния учетных записей оплаты по окладу в некотором периоде (удаление, изменение или ввод новых учетных записей), то это приводит к необходимости пересмотра результатов учетных записей оплаты по среднему заработку, если такие записи имеются в соответствующем периоде. Для реализации такого взаимного влияния учетных записей регистра расчета введено понятие базовый период записи регистра. Так, если учетная запись, соответствующая оплате по среднему заработку, имеет базовый период с 01.01.2008 по 31.03.2008 (т. е. использует средний заработок за три месяца), то любые изменения в учетных записях оплаты по окладу за этот период приведут к необходимости пересмотра учетной записи оплаты по среднему заработку.

12.3.2. Редактирование регистра расчета

Для работы с регистрами расчета предназначена ветвь [Регистры расчета](#) дерева конфигурации.

При редактировании регистра расчета определяется план видов расчета, поддержка периода действия и базового периода, периодичность, разрабатывается структура регистра: создаются наборы измерений, ресурсов и реквизитов регистра; если необходимо, создаются экранные и печатные формы просмотра движений регистра.

ПРИМЕЧАНИЕ. В качестве типа измерения регистра расчета не могут выступать значения типа [УникальныйИдентификатор](#), [ДвоичныеДанные](#), строка неограниченной длины.

В этом разделе будут описаны уникальные свойства регистра расчета в дополнение к общим свойствам объектов.

Редактирование свойств объектов типа [Регистр расчета](#) и создание подчиненных объектов выполняются в окне редактирования (см. [здесь](#)).

[План видов расчета](#) – основная характеристика регистра. Подробнее см. [здесь](#). Для регистра расчета можно указать только один план видов расчета.

[Период действия](#) – если свойство установлено, то взаимное влияние движений данного регистра принимает конкурирующий характер. Примером конкурирующих движений может служить начисление заработной платы и оплата больничного листа – нельзя одновременно и болеть, и работать, т. е. получать и оклад, и деньги по больничному листу. Такие расчеты взаимно исключают друг друга во времени, и система должна гарантировать, что ввод одного из них приведет к исключению другого.

Регистр расчета с поддержкой периода действия имеет ограничение на количество записей с одинаковыми значениями измерений, периода регистрации и вида расчета.

При расчете фактического периода действия, который выполняется при каждой записи набора записей регистра, может возникнуть ошибка нехватки памяти в том случае, если в расчете фактического периода действия участвует большое число одинаковых записей. Фактически, это значит, что есть ограничение на число записей, действующих в одном и том же отрезке времени. Например, для одного и того же сотрудника многократно введено начисление оклада за один и тот же период.

Значение ограничения зависит от различных условий (используемой СУБД, технических данных компьютера и т. д.), но, в любом случае, это ограничение не проявляется в условиях реальной практики.

[График](#) – свойство доступно, если установлено свойство [Период действия](#). Свойство представляет ссылку на регистр сведений, в котором описывается временная схема исходных данных, участвующих в расчете. График следует указывать для тех расчетов, которые зависят от исходных данных, распределенных в пределах периода действия по определенному правилу. Например, это может быть график учета рабочего времени организации с разбивкой по дням, учет лекционных часов с разбивкой по часам и т. д.

[Значение графика](#) – свойство доступно, если установлено свойство [Период действия](#). В свойстве

выбирается ресурс регистра сведений, определенного в свойстве **График**.

Например, в качестве графика указан регистр сведений **ГрафикРаботыОрганизации**. Этот регистр имеет реквизиты **РабочийДень** (тип **Булево**) и **РабочиеЧасы** (тип **Число**). В первом указывается признак, является ли данная дата рабочим днем, а во втором – число рабочих часов в рабочем дне. Выбор реквизита **РабочийДень** означает, что при расчете будет анализироваться, является ли конкретный день периода расчета рабочим (т. е. следует ли производить расчет заработной платы за указанный день). Выбор реквизита **РабочиеЧасы** означает, что при расчете будет выбираться число рабочих часов и на основе этого значения будет произведен указанный расчет. Собственно распределение задается в режиме 1С:Предприятие (вручную или средствами встроенного языка).

Дата графика – свойство доступно, если установлено свойство **Период действия**. В свойстве выбирается измерение регистра сведений, определенного в свойстве **График** и имеющее тип **Дата**. По значению данного свойства осуществляется привязка к значениям ресурса регистра сведений, указанного в свойстве **Значение графика**.

Базовый период – если свойство установлено, то устанавливается связанный характер взаимного влияния движений данного регистра. Примером связанных движений может служить связь расчета сумм начисления выплат по средней от сумм начисления в базовом периоде.

Периодичность – определяет период, с которым регистрируются движения и в пределах которого движения могут влиять друг на друга (для регистров, поддерживающих период действия).

Перерасчеты – подчиненные объекты регистра, которые позволяют задать правила взаимного влияния движений регистров. В группе **Связь** палитры свойств объекта в свойстве **Измерение** регистра указывается основное измерение текущего регистра, которое следует пересчитать при изменении данных ведущих регистров, указанных в свойстве **Данные** ведущих регистров. Например, перерасчет суммы удержания по физическому лицу будет формироваться при изменениях начислений (оплата труда, премии).

Если установлено свойство **Базовый период**, то формирование данных перерасчетов будет выполнено автоматически. Если свойство не установлено, то формирование данных перерасчетов должно производиться пользователем вручную (при проектировании следует разработать специальную форму ввода перерасчетов и механизм их выполнения).

Глава 13. Бизнес-процессы и задачи

13.1. Основные понятия

Бизнес-процессы в системе «1С:Предприятие» предназначены для объединения отдельных операций в цепочки взаимосвязанных действий, приводящих к достижению конкретной цели. Например, цепочку по выписке счета, приему наличной оплаты и отпуску товара со склада можно представить как бизнес-процесс [Продажа товара за наличный расчет](#).

Бизнес-процессы в системе «1С:Предприятие» позволяют формализовать процедуры обработки тех или иных событий, возникающих в деятельности организации, и обеспечить участие в них исполнителей.

Применение механизмов бизнес-процессов в прикладных решениях позволяет повысить их эффективность, улучшить конечный результат и получить новые возможности.

Бизнес-процессы дают возможность перейти к процессному управлению и качественно улучшить деятельность предприятия за счет реинжиниринга и автоматизации бизнес-процессов.

Наибольший эффект дает автоматизация ключевых бизнес-процессов, которые начинаются и заканчиваются во внешней по отношению к организации среде.

Цепочки взаимосвязанных действий бизнес-процесса представляются с помощью карты маршрута бизнес-процесса. Карта маршрута описывает логику бизнес-процесса и весь его жизненный цикл от точки старта до точки завершения в виде схематического изображения последовательности прохождения взаимосвязанных точек маршрута.

Последовательное выполнение цепочки взаимосвязанных действий будем называть движением бизнес-процесса.

Точка маршрута – отражает этап жизненного цикла бизнес-процесса, связанный с выполнением, как правило, одной автоматической или ручной операции.

Задачи в системе «1С:Предприятие» позволяют вести учет заданий по исполнителям и служат отражением продвижения бизнес-процессов по точкам маршрута. При этом задачи могут создаваться не только бизнес-процессами, но и другими объектами информационной базы, и непосредственно пользователями.

13.2. Общая часть

Механизм бизнес-процессов в системе «1С:Предприятие» обеспечивается сразу несколькими объектами конфигурирования:

- Бизнес-процессы,
- Задачи,
- Регистр сведений,
- Параметр сеанса.

Как правило, типы реквизитов адресации задачи и измерений регистра сведений имеют ссылочный тип (например, [СправочникСсылка](#), поэтому к четырем вышеперечисленным видам добавляются еще справочники).

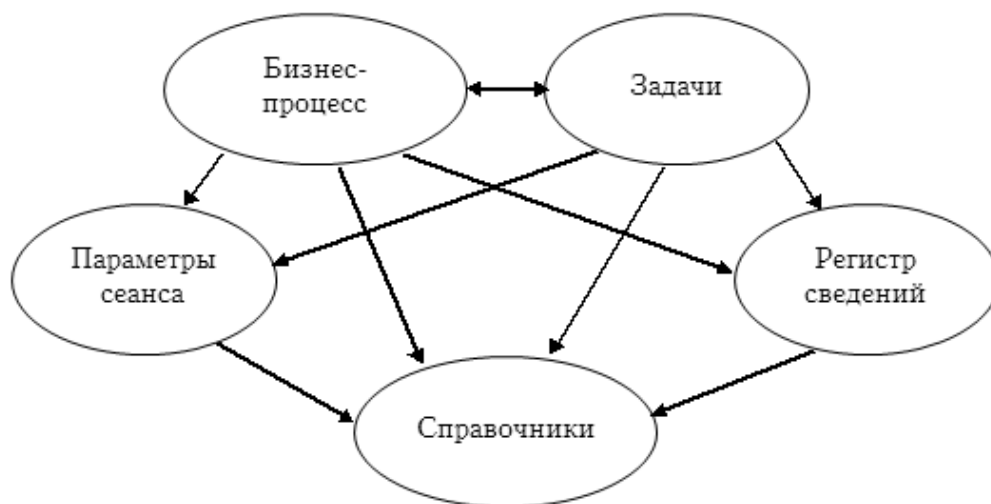


Рис. 300. Схема бизнес-процессов

Основные объекты – бизнес-процессы и задачи – взаимодействуют друг с другом (например, бизнес-процесс создает задачи, а задача в процессе выполнения приводит к продвижению его по маршруту).

Вспомогательные объекты – параметры сеанса, регистр сведений и справочники – не используют друг друга и основные объекты.

При создании карты маршрута бизнес-процесса используются справочники с predetermined данными (ролями, подразделениями и пр.) для установки их значений в свойства адресации точек маршрута. Бизнес-процессы создают задачи при переходе на точки маршрута и используют [Адресацию](#) (регистр сведения, см. ниже) для обработки групповых точек.

Задачи сообщают бизнес-процессам о своем выполнении, чем вызывают их движение дальше по маршруту. Регистр сведения используется ими для отбора задач для текущего исполнителя в соответствии с установленным параметром сеанса.

13.3. Маршрутизация

Бизнес-процессы в системе «1С:Предприятие» допускают следующие виды маршрутизации:

- **Жесткая.** Бизнес-процесс имеет строгую карту маршрута, не включающую в себя условных и параллельных переходов, с жестко определенными адресатами для каждой точки маршрута. Данный вид не допускает свободной и условной маршрутизации.
- **Свободная.** Адресаты точки карты маршрута бизнес-процесса не установлены и определяются программно или интерактивно в течение жизненного цикла бизнес-процесса.
- **Условная.** Карта маршрута предусматривает проверку условий и переход по соответствующим ветвям. Переходы могут быть как бинарными (условие), так и множественными (выбор варианта).
- **Параллельная.** Карта маршрута предусматривает разделение бизнес-процесса на параллельные ветви с возможностью последующего слияния (ожидания). Продвижение бизнес-процесса по каждой из параллельных ветвей происходит независимо по мере выполнения соответствующих задач.

Как правило, в реальных картах бизнес-процессов встречаются все эти типы маршрутизации.

13.4. Система адресации

Ключевым понятием в механизме бизнес-процессов и задач в «1С:Предприятии» является система адресации. Основное назначение системы адресации – обеспечить возможность не только персональной, но и ролевой адресации задач участникам бизнес-процессов.

Ролевая адресация (ролевая маршрутизация) – набор правил и соглашений, зафиксированных в настройках объектов метаданных, который позволяет определять конечного адресата (исполнителя), исходя из назначенных ему ролей, принадлежности к подразделению, а также других реквизитов адресации.

Реквизиты адресации задачи задают размерность адресного пространства в контексте автоматизируемой предметной области и используются для определения принадлежности задач конкретным исполнителям.

Определение конкретного исполнителя осуществляется с помощью свойств задачи – [Адресация](#), [Основной реквизит адресации](#) и [Текущий исполнитель](#).

Процесс определения основного реквизита адресации из остальных реквизитов адресации называется **разыменованием**.

[Адресация](#) – регистр сведений, который хранит актуальную на текущий момент информацию о соответствии исполнителей (основной реквизит адресации) структурным подразделениям, рабочим группам, выполняемым функциям и т. д., то есть всем остальным реквизитам адресации задач.

Один из реквизитов адресации задачи является основным и означает конкретного сотрудника – исполнителя заданий.



Рис. 301. Схема адресации

Поясним на примере работу системы адресации.

Допустим, что есть регистр сведений, состоящий из двух измерений – роль и сотрудник, в который внесены следующие записи.

Роль	Сотрудник
Кассир	Иванов
Менеджер	Петров

Допустим, что есть бизнес-процесс (например, [Принять наличную оплату](#)), в одной из точек которого в свойствах адресации установлена только роль [Кассир](#). При переходе бизнес-процесса на эту точку будет сформирована одна задача.

Свойство задачи	Значение
Наименование	Принять наличную оплату
Роль	Кассир
Сотрудник	-

При просмотре сотрудником [Ивановым](#) списка задач для себя система адресации покажет ему эту задачу, т. к. в регистре сведений есть запись о том, что для [Иванова](#) установлена роль [Кассир](#). Сотрудник [Петров](#) эту задачу не увидит.

Приведем примерную последовательность действий для создания двух различных бизнес-процессов:

1. Будем исходить из того, что выбрана 3-мерная система адресации – [сотрудник](#), [роль](#), [подразделение](#).
2. Создадим справочники для каждого из планируемых измерений адресации ([Сотрудники](#), [Роли](#),

Подразделения) и заполним их предопределенными значениями:

Сотрудники	Роли	Подразделения
Иванов	Кассир	Бухгалтерия
Петров	Менеджер	Отдел продаж
Сидоров	Руководитель отдела	Склад
	Кладовщик	

3. Создадим регистр сведений и добавим к нему измерения, по одному для каждого из ранее созданных справочников. Тип измерений следует установить как ссылку на соответствующий справочник:

Измерение	Тип
Сотрудник	СправочникСсылка.Сотрудники
Роль	СправочникСсылка.Роль
Подразделение	СправочникСсылка.Подразделения

4. Создадим параметр сеанса `ТекущийИсполнитель` и установим ему тип `СправочникСсылка.Сотрудники`.

5. Проинициализируем параметр сеанса при запуске системы:

[Копировать в буфер обмена](#)

Процедура `УстановкаПараметровСеанса(ТребуемыеПараметры)`

`Пользователь = Справочники.Сотрудники.`

`НайтиПоНаименованию(ИмяПользователя());`

`ПараметрыСеанса.ТекущийИсполнитель = Пользователь;`

КонецПроцедуры

6. Создадим задачу.

7. Установим созданный ранее регистр сведений в свойство задачи `Адресация`.

8. Добавим к задаче реквизиты адресации аналогично измерениям регистра сведений:

- `Сотрудник`,
- `Роль`,
- `Подразделение`.

9. Установим для созданных реквизитов адресации задачи тип в виде ссылки на соответствующий справочник и в свойстве `Реквизиты адресации` установим ссылку на измерение регистра сведений.

Реквизит адресации	Тип	Измерение адресации
<code>Сотрудник</code>	СправочникСсылка. Сотрудники	<code>Сотрудник</code>
<code>Роль</code>	СправочникСсылка. Роль	<code>Роль</code>
<code>Подразделение</code>	СправочникСсылка. Подразделения	<code>Подразделение</code>

10. Выберем реквизит `Сотрудник` в качестве основного реквизита адресации, установив его в соответствующем свойстве задачи.

11. Создадим первый бизнес-процесс и установим у него ссылку на созданную ранее задачу (свойство **Задача**).

Спроектируем маршрутную карту бизнес-процесса, устанавливая нужные реквизиты адресации для точек маршрута, выбирая их из predetermined данных соответствующих справочников.

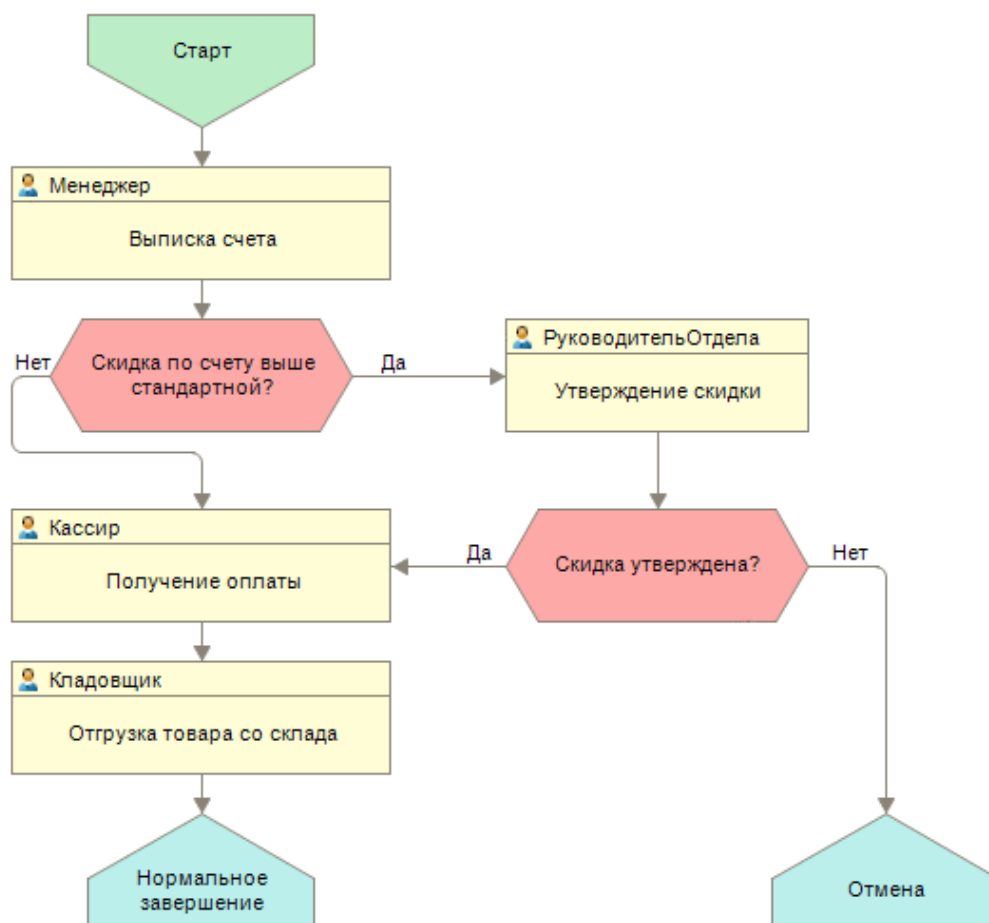


Рис. 302. Карта бизнес-процесса

12. Создадим следующий бизнес-процесс и установим у него ссылку на эту же задачу.

13. Спроектируем маршрутную карту созданного бизнес-процесса. И так далее.

В дальнейшем будем использовать этот пример для комментирования ключевых особенностей бизнес-процессов в системе «1С:Предприятие».

Рассмотрим подробнее несколько ключевых особенностей механизма бизнес-процессов.

13.5. Старт бизнес-процесса

Бизнес-процесс стартует при вызове метода `Старт()` или нажатии кнопки **Стартовать и закрыть** в форме объекта.

Нижеследующий фрагмент кода показывает программное создание, запись и старт бизнес-процесса.

[Копировать в буфер обмена](#)

```

БП = БизнесПроцессы.Согласование.СоздатьБизнесПроцесс();
БП.Дата = ТекущаяДата();
БП.Записать();
БП.Старт();
  
```

При старте выполняется следующая последовательность действий.

№	Внутренний механизм	Обработчики на встроенном языке

1		Вызов обработчика ПередСтартом у точки старта
2	Продвижение по карте маршрута до точки действия	
3	Формирование задач (см. здесь)	

Бизнес-процесс может быть записан, но не стартован. Это может оказаться полезным, если создание бизнес-процесса и его старт разделены во времени. Например, когда бизнес-процесс стартует при наступлении некоторого события.

13.6. Выполнение задач

При вызове метода **ВыполнитьЗадачу()** осуществляется проверка выполнения, после которой задача помечается как выполненная и об этом оповещается бизнес-процесс. Если все необходимые условия выполнены, то бизнес-процесс осуществляет переход на следующую точку маршрута.

№	Внутренний механизм	Обработчики на встроенном языке
1	Начало транзакции	
2		Вызов обработчика ПередВыполнением у задачи
3		Вызов обработчика ПередВыполнением у соответствующей точки маршрута
4	Установка свойства Выполнена у задачи равным Истина	
5		Вызов обработчика ПриВыполнении у задачи
6	Запись объекта задачи	
7		Вызов обработчика ПриВыполнении у соответствующей точки маршрута
8	Переход бизнес-процесса на следующую точку маршрута	
9	Формирование задач по новой точке маршрута (см. здесь)	
10	Завершение транзакции	

13.7. Разделение и слияние

Для разделения бизнес-процесса на несколько параллельно (одновременно и независимо) исполняемых ветвей используется точка разделения. Точка разделения имеет один вход и неограниченное количество выходов.

Для синхронизации разделенных ранее ветвей используется точка слияния. Бизнес-процесс не будет выполняться дальше точки слияния, пока все входящие в нее ветви не будут пройдены. Таким образом, точка слияния является этапом бизнес-процесса, на котором должны быть завершены все задачи по разделенным ранее веткам.

Допускается вложенное разделение и слияние. При этом каждая точка слияния будет синхронизировать только ветки «своей» точки разделения.

ВНИМАНИЕ! Разделение может быть и без слияния. В этом случае бизнес-процесс будет иметь несколько параллельных ветвей до своего завершения.

Слияние без разделения не допускается, о чем выдается соответствующее сообщение при проверке карты маршрута: **Не все линии, вошедшие в точку слияния, вышли из точки разделения.**

13.8. Ручное управление

Признаки завершения бизнес-процесса и выполнения у задачи можно устанавливать вручную, в обход

механизма бизнес-процессов, однако делать это нужно с полным пониманием всех возможных последствий.

13.8.1. Признак завершенности бизнес-процесса

Если установить признак завершенности, то бизнес-процесс будет считаться завершенным, даже несмотря на то, что связанные с ним задачи еще не выполнены. И при выполнении этих задач завершенный бизнес-процесс уже не будет двигаться дальше по маршруту.

Признак завершения можно установить у нестартованного бизнес-процесса. В этом случае его старт в дальнейшем будет невозможен.

Если вручную снять признак завершения с завершенного бизнес-процесса, то связанные с ним задачи все равно останутся выполненными. Таким образом, бизнес-процесс не будет завершен, но по нему не будет ни одной невыполненной задачи. Повторный старт такого бизнес-процесса невозможен, т. к. система будет считать его стартованным (по нему есть одна или более задач). Поэтому при ручном снятии признака завершения следует снять признаки выполнения у нужных задач таким образом, чтобы вернуть бизнес-процесс на нужную точку маршрута.

13.8.2. Признак выполнения задачи

Если установить признак выполнения задачи вручную, то это не приведет к продвижению бизнес-процесса дальше по маршруту. При этом также не будут вызваны обработчики событий `ПередВыполнением()` и `ПриВыполнении()` у задачи и соответствующей ей точке маршрута.

Ручная установка признака выполнения может привести к остановке бизнес-процесса – он не будет завершен, но по нему не будет ни одной невыполненной задачи.

Снятие признака выполнения у задачи может привести к появлению параллельного потока в незавершенном бизнес-процессе. Допустим, бизнес-процесс еще не завершен и по нему есть одна выполненная и одна невыполненная задача. Если снять признак выполнения с выполненной задачи, то у данного бизнес-процесса появятся две невыполненные задачи. При выполнении каждой из них бизнес-процесс будет двигаться дальше по карте маршрута от точки, соответствующей выполненной задаче. При этом бизнес-процесс будет считаться завершенным, когда все задачи в обеих параллельных ветвях будут выполнены.

Снятие признака выполнения у задачи, бизнес-процесс которой уже завершен, приведет к тому, что задача будет видна в списках как невыполненная, но ее выполнение не будет продвигать бизнес-процесс дальше по маршруту.

13.8.3. Удаление задач

Если удалить невыполненные задачи незавершенного бизнес-процесса, то он может остановиться. Такой бизнес-процесс будет незавершенным, но по нему не будет ни одной активной (невыполненной) задачи.

Задачи используются для отображения реальной карты маршрута бизнес-процесса, чтобы показать уже пройденные точки маршрута и активные (невыполненные). Поэтому удаление задач может привести к некорректному и противоречивому отображению пройденных и активных точек маршрута.

Удаление всех задач для незавершенного бизнес-процесса переводит его в статус нестартованного.

13.8.4. Добавление задач

Если вручную создать новую задачу по завершенному бизнес-процессу, то бизнес-процесс все равно будет считаться завершенным и выполнение этой задачи не приведет к его продвижению по карте маршрута.

Если вручную создать новую задачу для еще не стартовавшего бизнес-процесса, то он получает статус стартованного и выполнение этой задачи приведет к его продвижению дальше по карте маршрута.

Создание новой задачи для уже стартовавшего и незавершенного бизнес-процесса приводит к его распараллеливанию.

13.9. Условный переход

Для условного ветвления бизнес-процесса используется точка условного перехода. Важной особенностью этой точки является обработчик проверки условия, наличие которого обязательно и контролируется при проверке карты маршрута перед сохранением бизнес-процесса. Если обработчик отсутствует, то будет выдано

предупреждение: **Точка условия не имеет обработчика события "Проверка условия"**.

Этот обработчик должен вернуть результат проверки условия, от которого будет зависеть выбор следующей точки маршрута. Если результат **Истина**, то бизнес-процесс пойдет по ветке **Да**, в противном случае – по ветке **Нет**. По умолчанию результат устанавливается равным значению **Ложь**.

Обработчик проверки условия может, например, иметь такой вид:

[Копировать в буфер обмена](#)

```
Процедура ОграничениеСкидкиПроверкаУсловия(ТочкаМаршрутаБП, Результат)
    Если ПолучитьСкидкуПоСчету() > ПолучитьОбычнуюСкидку() Тогда
        Результат = Истина;
    Иначе
        Результат = Ложь;
    КонецЕсли;
КонецПроцедуры
```

Для реализации многовариантного выбора можно использовать несколько последовательно соединенных точек условного перехода, однако удобнее для этого применять точку выбора варианта.

13.10. Выбор варианта

Для выбора одного из нескольких возможных путей используется точка выбора варианта. Важной особенностью этой точки является обработчик выбора варианта, наличие которого обязательно и контролируется при проверке карты маршрута перед сохранением бизнес-процесса. Если обработчик отсутствует, то будет выдано предупреждение: **Точка выбора варианта не имеет обработчика события Выбор варианта**.

Этот обработчик должен установить параметр **Результат** равным одному из предусмотренных вариантов. Процедура-обработчик может иметь примерно такой вид:

[Копировать в буфер обмена](#)

```
Процедура ВыборВарианта(ТочкаВыбораВарианта, Результат)
    Если ВидОплаты = Перечисления.ВидОплаты.Наличная Тогда
        Результат = ТочкаВыбораВарианта.Варианты.Наличная;
    ИначеЕсли ВидОплаты = Перечисления.ВидОплаты.Безналичная Тогда
        Результат = ТочкаВыбораВарианта.Варианты.Безналичная;
    ИначеЕсли ВидОплаты = Перечисления.ВидОплаты.Взаимозачет Тогда
        Результат = ТочкаВыбораВарианта.Варианты.Взаимозачет;
    ИначеЕсли ВидОплаты = Перечисления.ВидОплаты.Кредит Тогда
        Результат = ТочкаВыбораВарианта.Варианты.Кредит;
    КонецЕсли;
КонецПроцедуры
```

В этом обработчике **ВидОплаты** – реквизит бизнес-процесса.

Если в процедуре-обработчике выбора варианта не установить какое-либо значение параметра **Результат**, то это приведет к ошибке и отмене транзакции, в рамках которой выполнялся выбор варианта.

13.11. Формирование задач

Задачи формируются только при поступлении бизнес-процесса в точки действия или точки вложенных бизнес-процессов. При прохождении других точек (условный переход, разделение, слияние, обработка и пр.) бизнес-процесс автоматически выполняет предусмотренные действия и переходит к следующей точке в соответствии с картой маршрута.

Рассмотрим, например, процесс перехода бизнес-процесса на первую точку действия в результате вызова у него метода **Старт()**.

При прохождении маршрута бизнес-процесс в точках действия или точках вложенных бизнес-процессов может создавать одну или несколько задач. Несколько задач будут сформированы в том случае, если у точки маршрута установлен признак «групповая». В этом случае бизнес-процесс отбирает в регистре сведений (**Адресация**) все записи, соответствующие установленным в данной точке реквизитам адресации, и для каждой из них формирует свою задачу.

Например, если в точке маршрута установлена адресация только по роли **Кассир**, а в регистре сведений имеются две записи вида, то будут сформированы две задачи, у которых будут установлены оба реквизита

адресации – и роль, и конечный исполнитель.

Сотрудник	Роль	Подразделение
Иванов	Кассир	
Петров	Кассир	

Таким образом, для групповых точек маршрута ролевая маршрутизация применяется только один раз – в момент формирования списка задач.

Рассмотрим последовательность вызова обработчиков событий на встроенном языке в момент перехода на первую точку маршрута [Выписка счета](#).

№	Внутренний механизм	Обработчики на встроенном языке
1	Начало транзакции	
2		Вызов обработчика <code>ПередСозданиемЗадач()</code>
3	Формирование списка задач	
4		Вызов обработчика <code>ПриСозданииЗадач()</code>
5	Запись всех сформированных задач	
6		Вызов обработчика <code>ПередЗаписью()</code> у задачи
7	Запись задачи	
8		Вызов обработчика <code>ПриЗаписи()</code> у задачи
9	Завершение транзакции	

На втором шаге происходит вызов обработчика `ПередСозданиемЗадач()`. Этот обработчик вызывается до формирования списка задач самим бизнес-процессом, поэтому ему передается пустой массив формируемых задач с тем, чтобы его можно было сформировать самостоятельно и отказаться от стандартной обработки.

На третьем шаге бизнес-процесс проверяет, вернул ли предыдущий обработчик `СтандартнаяОбработка = Истина`. Если да, то производится разыменование установленных в точке маршрута реквизитов адресации и формируется одна задача или список задач (для групповой точки) по количеству результатов разыменования (например, количество сотрудников отдела). При этом каждой сформированной задаче устанавливается наименование, ссылка на бизнес-процесс и точку маршрута и соответствующие реквизиты адресации.

На четвертом шаге осуществляется вызов обработчика `ПриСозданииЗадач()`. В этот обработчик передается список задач, сформированный ранее в обработчике `ПередСозданиемЗадач()` или самим бизнес-процессом. Задачи еще не записаны. В этом обработчике можно предусмотреть тонкую настройку сформированных задач – установку контрольного срока, приоритета и других дополнительных реквизитов. Также в этом обработчике можно добавить к массиву сформированных задач новые задачи.

На пятом шаге проверяется нормальное завершение обработчика `ПриСозданииЗадач()`. Если обработчик в параметре `Отказ` вернул значение `Истина`, то процесс создания задач прерывается и вызывается исключение. В нашем случае это приведет к отмене выполнения метода `Старт()`. Если же `Отказ = Ложь`, то производится запись всех задач из сформированного массива с вызовом обработчиков `ПередЗаписью()` и `ПриЗаписи()` у каждой отдельной задачи (шаги 6 и 8 соответственно).

При формировании бизнес-процессом массива задач у них автоматически заполняются следующие реквизиты:

- Наименование устанавливается равным наименованию соответствующей точки маршрута, например [Выписка счета](#).
- Ссылка на экземпляр бизнес-процесса, породившего эту задачу.
- Ссылка на точку маршрута бизнес-процесса.
- Реквизиты адресации задачи устанавливаются равными реквизитам адресации соответствующей точки

маршрута. Например, если точка маршрута адресована роли **Кассир**, то в реквизит адресации задачи **Роль** будет установлено **Кассир**.

13.12. Проверка выполнения

Выполнение задач может осуществляться не только пользователями, но и автоматизированными процедурами. Например, если задача предусматривает проведение документа, то автоматическая процедура слежения за такими задачами может определять, что нужный документ уже проведен, и пометить задачу как выполненную путем вызова у нее метода **Выполнить ()**.

Для организации такого рода автоматизированных процедур предназначен метод **ПроверкаВыполнения ()** у задачи и соответствующие ему обработчики у точек маршрута.

Рассмотрим последовательность действий, которая произойдет в результате работы следующего кода на встроенном языке.

[Копировать в буфер обмена](#)

```
Если Задача.ПроверитьВыполнение ( ) Тогда
    Задача.ВыполнитьЗадачу ( ) ;
КонецЕсли
```

№	Внутренний механизм	Обработчики на встроенном языке
1	Обработка вызова метода ПроверитьВыполнение ()	
2		Вызов обработчика ОбработкаПроверкиВыполнения () у задачи. Если Результат равен Ложь , то метод ПроверитьВыполнение () сразу возвращает Ложь
3		Вызов обработки ОбработкаПроверкиВыполнения () у соответствующей точки маршрута
4	Возврат результата вызова обработчика из предыдущего пункта и, если он равен Истина , вызов метода ВыполнитьЗадачу ()	

Описание одного из способов использования автоматизированного выполнения задач см. [здесь](#).

13.13. Выполнение вложенных процессов

При проектировании маршрутной карты можно предусматривать старт вложенных бизнес-процессов. В этом случае основной бизнес-процесс ждет завершения вложенного бизнес-процесса и только после этого переходит к следующей точке маршрута.

При переходе на точку маршрута вида **Вложенный бизнес-процесс** выполняется следующая последовательность действий.

№	Внутренний механизм	Обработчики на встроенном языке
1	Начало транзакции	
2		ПередСозданиемВложенныхБизнесПроцессов ()
3		Вызов обработчика ПередСозданиемЗадач () для точки маршрута
4	Если СтандартнаяОбработка , то формируется массив задач	
5		ПриСозданииЗадач ()
6	Запись массива сформированных задач и создание массива вложенных бизнес-процессов	
7		ПриСозданииВложенныхБизнесПроцессов ()

8	Запись и старт сформированных бизнес-процессов
9	Завершение транзакции

Рассмотрим подробнее.

На втором шаге происходит вызов обработчика `ПередСозданиемВложенныхБизнесПроцессов()`, в котором можно добавить свои бизнес-процессы в массив формируемых бизнес-процессов (по умолчанию массив приходит пустым). Если были добавлены бизнес-процессы в массив, то стандартная механика генерации бизнес-процессов будет отменена.

На третьем шаге происходит вызов обработчика `ПередСозданиемЗадач()`. В него передается пустой, еще не сформированный массив задач. Если этот обработчик не изменит стандартную обработку, то сформированный им массив задач будет очищен на третьем шаге и заполнен задачами (по одной задаче на каждый стартующий вложенный бизнес-процесс) с установленным наименованием и ссылками на бизнес-процесс и точку маршрута.

На пятом шаге можно «донастроить» сформированные задачи и добавить к ним новые в случае необходимости.

На шестом шаге происходит запись сформированных задач, после чего по каждой из них создается вложенный бизнес-процесс установленного в точке маршрута типа. У созданных бизнес-процессов устанавливается дата и ссылка на ведущую задачу.

На седьмом шаге происходит вызов обработчика `ПриСозданииВложенныхБизнесПроцессов()`. Обработчик этого события может «донастроить» автоматически сформированные бизнес-процессы (их количество равно количеству задач после обработчика `ПриСозданииЗадач()`) или удалить некоторые из них, а также добавить к ним новые бизнес-процессы. Запись списка бизнес-процессов будет осуществлена после завершения обработчика.

13.14. Завершение бизнес-процесса

Завершение является последним этапом в жизненном цикле бизнес-процесса. Бизнес-процесс автоматически становится завершенным (свойству `Завершен` устанавливается значение `Истина`) при достижении точки завершения и при условии отсутствия невыполненных задач по этому бизнес-процессу.

Если у бизнес-процесса установлено свойство `Ведущая задача`, т. е. он является вложенным, то при своем завершении он помечает эту задачу как выполненную. Это, в свою очередь, приводит к продвижению основного бизнес-процесса дальше по маршруту.

При переходе на точку завершения вызывается обработчик `ПриЗавершении()`. Если он установит `Отказ` равным `Истина`, например, если не выполнены все необходимые условия завершения бизнес-процесса, то обработка прерывается. Задача по точке маршрута, выполнение которой привело к переходу на точку завершения, при этом остается невыполненной.

Установка свойству `Завершен` значения `Истина` (средствами встроенного языка или интерактивно) может использоваться для прерывания хода бизнес-процесса или для исключения его из списка активных (незавершенных) бизнес-процессов. При этом никакие обработчики, кроме `ПередЗаписью` и `ПриЗаписи`, не вызываются. Выполнение ведущей задачи при этом не производится.

13.15. Стандартные реквизиты бизнес-процессов и задач

В таблицах перечислены предопределенные поля бизнес-процессов и задач.

Стандартные реквизиты бизнес-процессов:

Реквизит	Тип
ПометкаУдаления	Булево
Номер	Строка или Число
Дата	Дата
Завершен	Булево

ВедущаяЗадача	ЗадачаСсылка.<Имя задачи>
Ссылка	БизнесПроцессСсылка . <Имя бизнес-процесса>

Стандартные реквизиты задач:

Реквизит	Тип
ПометкаУдаления	Булево
Номер	Строка или Число
Дата	Дата
Наименование	Строка
Выполнена	Булево
БизнесПроцесс	БизнесПроцессСсылка . <Имя бизнес-процесса>
ТочкаМаршрута	БизнесПроцессСсылка . <Имя бизнес-процесса>
Ссылка	ЗадачаСсылка.<Имя задачи>

13.16. Бизнес-процессы с несколькими точками старта

Наличие нескольких точек старта предполагает, что выбор конкретной точки для старта определяется внешними по отношению к бизнес-процессу условиями.

Если же бизнес-процесс обладает всей необходимой информацией, чтобы при старте самостоятельно принять решение о выборе того или иного маршрута, то достаточно одной точки старта, следом за которой будет идти точка проверки условия или точка выбора варианта.

Если бизнес-процесс имеет несколько точек старта, то при вызове метода `Старт()` необходимо указать конкретную точку. В противном случае будет выдано сообщение об ошибке. Поэтому при создании бизнес-процесса с несколькими точками старта необходимо сделать следующее:

- определить интерактивные команды старта для указания корректной точки старта в каждом случае старта бизнес-процесса.
- если данный бизнес-процесс является вложенным для других бизнес-процессов, то в соответствующих точках маршрута нужно прописать обработчик `ПриСозданииВложенныхБизнесПроцессов()` так, чтобы записывать и стартовать с нужной точки все бизнес-процессы из массива сформированных.

Пример:

[Копировать в буфер обмена](#)

```
Процедура ВложенноеСогласованиеПриСозданииВложенныхБП(ТочкаМаршрутаВП, ФормируемыеПроцессы, Отказ)
    Для каждого БизнесПроцесс из ФормируемыеПроцессы Цикл
        БизнесПроцесс.Записать();
        Точки = БизнесПроцессы.СогласованиеДокумента.ТочкиМаршрута;
        БизнесПроцесс.Старт(Точки.УпрощенноеСогласование);
    КонецЦикла
КонецПроцедуры
```

В остальном использование бизнес-процессов с несколькими точками старта ничем не отличается от обычных бизнес-процессов.

13.17. Обратная связь

Другие объекты информационной базы (документы, элементы справочников) могут быть вовлечены в бизнес-процессы и могут влиять на них.

Для эффективного использования механизма бизнес-процессов возникает необходимость автоматически выполнять соответствующие задачи при выполнении требуемых операций с другими объектами информационной базы (например, при проведении документа, при установке скидки по выписанному счету, при резервировании товара на складе и т. д.).

Важной особенностью механизма бизнес-процессов в системе «1С:Предприятие» является то, что он не требует существенного изменения используемых прикладных решений. Поэтому реакция бизнес-процессов и задач на изменение других объектов информационной базы может настраиваться без существенного изменения этих объектов.

Рассмотрим сказанное на примере. Допустим, что задача требует проведения документа и нужно, чтобы при проведении документа она выполнялась автоматически и пользователю не требовалось открывать список задач, находить в нем нужную задачу и выполнять ее.

Для этого последовательно выполним следующие действия:

- Добавим оповещение о проведении в форму документа:

[Копировать в буфер обмена](#)

```
Процедура ПослеЗаписи(Отказ)
Если БылоПроведение Тогда
    Оповестить ("ПроведениеДокумента", , ЭтотОбъект.Ссылка);
КонецЕсли;
КонецПроцедуры
```

- Зарегистрируем обработчик оповещения. Это можно сделать в обработчике ПриНачалеРаботыСистемы() модуля управляемого приложения:

[Копировать в буфер обмена](#)

```
ПодключитьОбработчикОповещения ("ОбработчикОповещения");
```

- Из обработчика оповещения вызовем метод серверного общего модуля (например, РаботаСБизнесПроцессами), который выполнит необходимые проверки и задачу:

[Копировать в буфер обмена](#)

```
// Обработчик оповещения в модуле управляемого приложения
Процедура ОбработчикОповещения(ИмяСобытия, Параметр, Источник) Экспорт
Если ИмяСобытия = "ПроведениеДокумента" Тогда
    РаботаСБизнесПроцессами.ВыполнитьЗадачуПоДокументу(Источник);
КонецЕсли;
КонецПроцедуры
...
&НаСервере
// Метод серверного общего модуля РаботаСБизнесПроцессами
Процедура ВыполнитьЗадачуПоДокументу(ДокументСсылка) Экспорт
Запрос = Новый Запрос;
Запрос.УстановитьПараметр("Парам", ДокументСсылка);
Запрос.Текст =
    "ВЫБРАТЬ
    | Ссылка
    | ИЗ
    | Задача.Задача.ЗадачиПоИсполнителю
    |
    | ГДЕ
    | Документ = &Парам";
Выборка = Запрос.Выполнить().Выбрать();
Пока Выборка.Следующий() Цикл
    ТекущаяЗадача = Выборка.Ссылка.ПолучитьОбъект();
    Если ТекущаяЗадача.ПроверитьВыполнение() Тогда
        ТекущаяЗадача.ВыполнитьЗадачу();
    КонецЕсли;
КонецЦикла;
КонецПроцедуры
```

Другим способом автоматизированного выполнения задач является создание регламентного задания (с необходимым расписанием), которое будет отбирать все задачи по нужному исполнителю, проверять их выполнение и, в случае успешной проверки, автоматически выполнять их.

Рассмотрим специфические особенности конфигурирования объектов бизнес-процессов и задач.

13.18. Карта маршрута

Карта маршрута описывает логику бизнес-процесса и весь его жизненный цикл от точки старта до точки завершения в виде схематического изображения последовательности прохождения взаимосвязанных точек маршрута.

Для редактирования карты маршрута на закладке [Прочее](#) окна редактирования бизнес-процесса нужно нажать кнопку [Карта маршрута](#) (подробнее см. [здесь](#)).

13.19. Редактирование бизнес-процесса

В процессе конфигурирования может быть создано произвольное количество видов бизнес-процессов. Назначение каждого бизнес-процесса определяет его структуру и свойства, которые описываются в конфигурации.

Конфигуратор позволяет описать структуру бизнес-процесса, создать формы и карту маршрута бизнес-процесса.

Свойства бизнес-процессов редактируются в палитре свойств или окне редактирования объекта (см. [здесь](#)).

Наряду с общими свойствами, присущими всем объектам метаданных, бизнес-процессы обладают рядом специфических свойств.

Свойство [Задачи](#) определяет ссылку на сконфигурированный ранее объект задачи. Бизнес-процессу обязательно должна быть назначена одна задача из числа уже существующих в конфигурации. Задачи используются бизнес-процессом для формирования заданий по исполнителям или для запуска вложенных бизнес-процессов. Если задача не назначена, то при сохранении конфигурации базы данных будет выдана ошибка: [Не выбрана задача бизнес-процесса](#).

[Автонумерация](#). Установка свойства приводит к тому, что вновь введенному бизнес-процессу номер будет присваиваться автоматически. Автоматически присвоенный номер можно исправить.

[Длина номера](#). Устанавливает максимальную длину номера бизнес-процесса. Конфигуратор позволяет установить длину номера равной 0, если в бизнес-процессе данного вида номер не используется.

[Тип номера](#). Свойство позволяет выбрать тип значения для номера бизнес-процесса – [Число](#) или [Строка](#). Это свойство аналогично свойству [Тип номера документа](#).

Выбор строкового типа кода бывает полезен, когда используется сложная система нумерации, и номер может включать помимо цифр также буквы и символы-разделители.

[Контроль уникальности](#). Если это свойство установлено, то при вводе нового бизнес-процесса его номер проверяется на уникальность в пределах, установленных в свойстве [Периодичность](#).

[Периодичность](#). Свойство устанавливает пределы контроля уникальности номеров бизнес-процессов и период повторяемости номеров. Если установлено свойство [Контроль уникальности](#), то в свойстве [Периодичность](#) указывается, в каких пределах будет осуществляться этот контроль.

При установленном свойстве [Автонумерация](#) система «1С:Предприятие» будет присваивать очередной порядковый номер каждому новому бизнес-процессу. После завершения периода, установленного в свойстве [Периодичность](#), нумерация бизнес-процессов начнется с 1.

На закладке [Права](#) имеется возможность установки привилегированного режима при создании задач (свойство [Привилегированный режим при создании задач](#)):

- Если свойство установлено, то все действия по формированию задач система будет выполнять в привилегированном режиме (при исполнении на стороне сервере и в файловом варианте). Однако привилегированный режим не будет установлен, если формирование задач выполняется в клиент-серверном варианте на стороне толстого клиента.
- При создании новых бизнес-процессов свойство установлено в значение [Истина](#), если в свойствах конфигурации указан основной режим запуска – управляемое приложение, и в значение [Ложь](#), если основным режимом запуска указан обычный.

Помимо основных реквизитов можно создать набор реквизитов, позволяющих хранить дополнительную

информацию.

Если объект предметной области, которой соответствует бизнес-процесс, имеет не только такие «простые» свойства, как, например, дату, номер, важность или контрольный срок, но и составные (списочные) свойства, как, например, список документов на согласование, список резолюций, список участников бизнес-процесса, для бизнес-процесса может быть создан набор табличных частей.

13.20. Редактирование задачи

Объекты типа **Задачи** предназначены для выдачи и исполнения заданий пользователями системы. Задания могут формироваться как самими пользователями, так и конкретными бизнес-процессами.

Задачи могут применяться самостоятельно или использоваться для обеспечения функционирования бизнес-процессов разного вида.

В процессе конфигурирования может быть создано произвольное количество видов задач, однако, как правило, задача создается одна для всех видов бизнес-процессов.

Описываемые в конфигурации структура и свойства задачи определяются особенностями автоматизируемой предметной области.

Для каждой задачи может быть создано несколько форм списка, выбора, просмотра и редактирования.

Все задачи характеризуются номером, датой, временем и наименованием. При формировании задач бизнес-процессами наименование устанавливается аналогичным наименованию соответствующей точки маршрута бизнес-процесса.

Свойства задачи редактируются в палитре свойств или окне редактирования объекта (см. [здесь](#)).

Наряду с общими свойствами, присущими всем объектам метаданных, задачи обладают рядом специфических свойств.

Адресация. Задаче может быть назначен неперiodический регистр сведений, с измерениями которого можно связать реквизиты адресации задачи. Это связывание позволяет определять значение основного реквизита адресации задачи на основании данных, содержащихся в соответствующем регистре сведений, что делает возможной не только прямую адресацию задач конкретным исполнителям, но и ролевую.

Основной реквизит адресации. Один из реквизитов адресации задачи может быть назначен основным. В этом случае именно в этом реквизите адресации необходимо будет указывать конкретного исполнителя задания. Если исполнитель не будет указан, то значение этого реквизита адресации будет определяться из связанного с задачей регистра сведений (см. свойство **Адресация**).

Текущий исполнитель. Это свойство устанавливает ссылку на параметр сеанса, в котором будет храниться текущий исполнитель. Свойство используется, например, как значение по умолчанию для свойства **Исполнитель** табличного поля списка задач.

Автопрефикс номера задачи. Может принимать значения **НеИспользовать** и **НомерБизнесПроцесса**. Если это свойство установлено в значение **НомерБизнесПроцесса**, то при создании новой задачи ее номер автоматически дополняется номером соответствующего ей бизнес-процесса.

Группа подчиненных объектов **Реквизиты адресации** устанавливает набор реквизитов, которые определяют тип и размерность системы адресации задач этого вида в контексте автоматизируемой предметной области. Один из этих реквизитов может быть установлен основным (см. свойство **Основной реквизит адресации**). Реквизиты адресации можно связать с измерениями регистра сведений. Это связывание используется системой для определения значения основного реквизита адресации, если оно не указано, и делает возможным не только прямую, но и ролевую адресацию.

Длина номера. Устанавливает максимальную длину номера задачи.

Тип номера. Свойство позволяет выбрать тип значения для номера задачи – **Число** или **Строка**. Выбор строкового типа кода бывает полезен, когда используется сложная система нумерации и номер может включать помимо цифр также буквы и символы-разделители.

Контроль уникальности. Если это свойство установлено, то при вводе новой задачи ее номер проверяется на уникальность.

Автонумерация. Установка свойства приводит к тому, что вновь введенной задаче номер будет присваиваться автоматически. Автоматически присвоенный номер можно исправить.

Если объект предметной области, которой соответствует задача, имеет не только такие «простые» свойства, как, например, дату, номер, важность или контрольный срок исполнения, но и составные (списочные) свойства, как, например, список документов для согласования, то может быть создан набор табличных частей.

Глава 14. Анализ данных и прогнозирование

Механизм анализа данных и прогнозирования позволяет реализовывать в прикладных решениях различные средства для выявления закономерностей, которые обычно скрываются за большими объемами информации.

Например, проанализировав данные о продажах товаров, можно выявить группы товаров, которые обычно покупаются вместе. В дальнейшем (один из многочисленных вариантов) эта информация может использоваться при раскладке товара в розничном магазине. Товары могут располагаться рядом (пришел покупатель за мангалом, увидел рядом жидкость для растопки, угли, мясо, удочки, резиновую лодку... в итоге купил все), могут располагаться в разных углах торгового зала (пришел покупатель за молоком, пока дойдет до хлеба, пройдет через весь магазин и еще чего-нибудь купит).

Другим примером использования механизма анализа данных является прогнозирование поведения контрагента, исходя из имеющихся данных о нем. Проведя такой анализ, можно выяснить, как зависит объем их закупок от территориального расположения, размера компании, времени сотрудничества и прочих показателей. На основании этих зависимостей можно спрогнозировать поведение нового контрагента и выбрать соответствующую стратегию для работы с ним.

Используя возможность построения прогнозов, можно планировать закупочную кампанию. Рассмотрим пример: в прошлом месяце зоомагазин продал 100 морских свинок. Нужно спланировать закупку товара на следующий месяц. Одним из вариантов (очень часто используемых) является ввод поправочного коэффициента на продажи прошлых периодов. Считаем, что поправочный коэффициент (коэффициент повышения спроса) равен 1,5. В итоге при планировании закупок описанным методом будем к следующему месяцу планировать закупку 150 морских свинок. Но если проанализировать, что обычно покупают клиенты после покупки таких питомцев, то можно прийти к совершенно другому выводу. Воспользовавшись возможностью проведения анализа данных, построения прогноза по этому анализу, можно прийти к выводу, что закупать на следующий месяц нужно корм, различные наполнители, сено и другие «аксессуары».

Следует отметить, что материал данной главы в первую очередь будет посвящен обзору механизмов «1С:Предприятия» и только косвенно будет касаться (на простых примерах) способов использования полученной информации.

14.1. Общие положения

Общую схему работы механизма анализа и прогнозирования данных можно представить следующим образом:



Рис. 303. Схема взаимодействия элементов механизма анализа данных

Механизм позволяет работать как с данными, полученными из информационной базы, так и с данными, полученными из другого источника, предварительно загруженными в таблицу значений или табличный документ.

Применяя к исходным данным один из видов анализа, можно получить результат анализа. Результат анализа представляет собой некую модель поведения данных; может быть отображен в итоговом документе или сохранен для дальнейшего использования.

Дальнейшее использование результата анализа заключается в том, что на его основе может быть создана модель прогноза, позволяющая прогнозировать поведение новых данных в соответствии с имеющейся моделью.

Например, можно проанализировать, какие товары приобретаются вместе (в одной накладной), и сохранить созданную на

основе данного анализа модель прогноза в базе данных. В дальнейшем при создании очередной накладной ранее сохраненную модель прогноза можно извлечь из информационной базы, подать ей на вход новые данные, содержащиеся в этой накладной, и на выходе получить прогноз – список товаров, которые очередной клиент тоже приобретет (с определенной долей вероятности), если их ему предложить.

14.1.1. Основные объекты механизма

Схематично взаимосвязь основных объектов механизма анализа данных и

прогнозирования можно показать следующим образом:

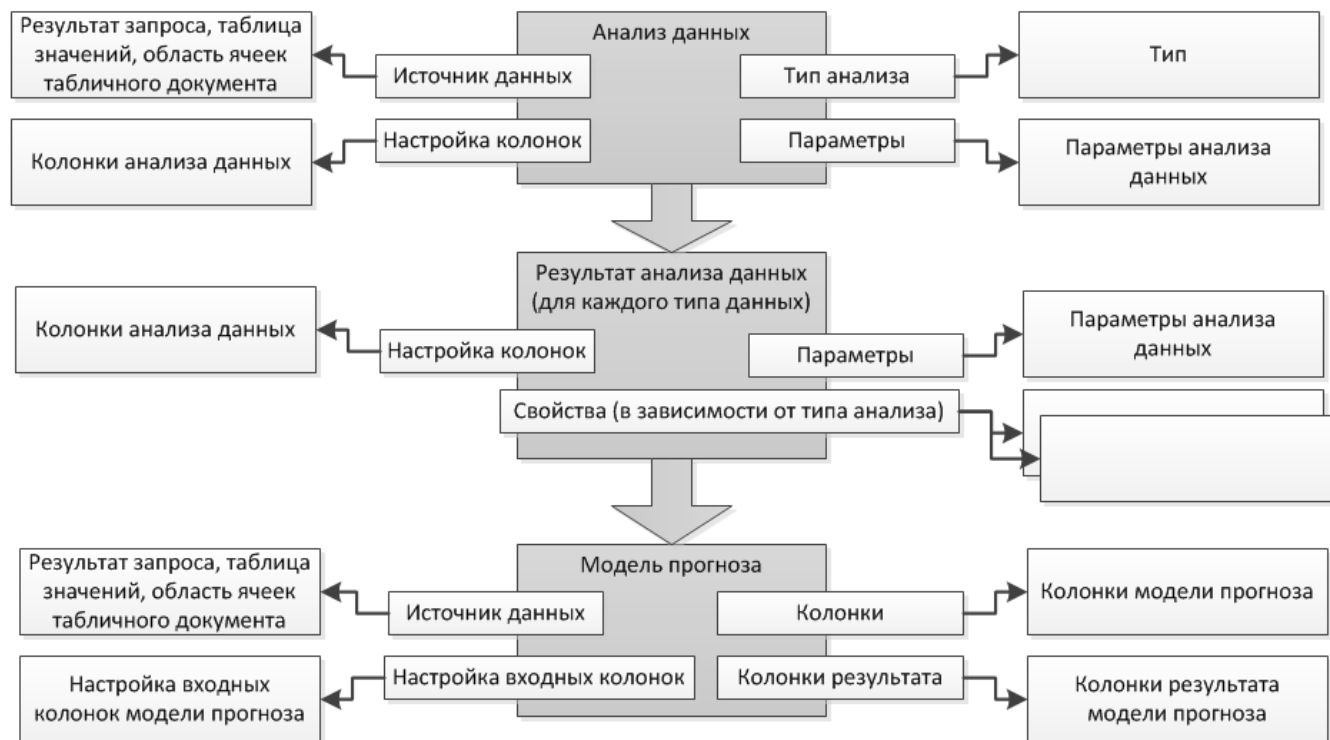


Рис. 304. Взаимосвязь основных объектов

Анализ данных – объект, непосредственно выполняющий анализ данных. Ему устанавливается источник данных, задаются параметры и исходные данные. Результатом работы данного объекта является **результат анализа данных**, причем для каждого типа анализа существует свой объект для работы с результатом анализа:

- **РезультатАнализаДанныхОбщаяСтатистика**,
- **РезультатАнализаДанныхПоискАссоциаций**,
- **РезультатАнализаДанныхПоискПоследовательностей**,
- **РезультатАнализаДанныхДеревоРешений**,
- **РезультатАнализаДанныхКластеризация**.

Настройка колонок анализа данных – коллекция входных колонок анализа данных. Для каждой колонки указывается тип данных, содержащихся в ней, роль, выполняемая колонкой, дополнительные настройки, зависящие от типа производимого анализа.

Параметры анализа данных – набор параметров производимого анализа данных. Состав параметров зависит от типа анализа. Например, для кластерного анализа указывается количество кластеров, на которые необходимо разбить исходные объекты, тип измерения расстояния между объектами и т. п.

Источник данных – исходные данные для анализа. В качестве источника данных может выступать результат запроса, область ячеек табличного документа, таблица значений.

Результат анализа данных – специальный объект, содержащий информацию о результате анализа. Для каждого вида анализа предусмотрен свой результат. Например, результатом анализа данных **Дерево решений** будет объект типа **РезультатАнализаДанныхДеревоРешений**. В дальнейшем результат может быть выведен в табличный документ при помощи **построителя отчета анализа данных**, может быть выведен посредством программного доступа к его содержимому, может быть использован для создания **модели прогноза**. Любой результат анализа данных может быть сохранен для последующего использования.

Модель прогноза – специальный объект, позволяющий выполнять прогноз на основании входных данных (выборка для прогноза, настройки колонок выборки и результата, результат анализа). Тип модели прогноза зависит от типа результата

анализа данных. Например, модель, созданная для типа анализа **Поиск ассоциаций**, будет иметь тип **Модель Прогноза Поиск Ассоциаций**. Такая модель сможет выдавать прогнозы типа: т. к. данный покупатель купил заданный набор товаров, то с определенной вероятностью он должен купить и другой набор товаров. На вход модели прогноза передается **источник данных для прогноза**. Результатом является таблица значений, содержащая прогнозируемые значения.

Настройка входных колонок – набор специальных объектов, показывающих соответствие между колонками модели прогноза и колонками выборки прогноза. Например, колонке модели прогноза с именем **Товар** может соответствовать колонка выборки **Номенклатура**.

Настройка колонок результата – позволяет управлять тем, какие колонки будут помещены в результирующую таблицу модели прогноза. Например, для поиска ассоциаций можно вывести в результат номенклатуру, которую, скорее всего, приобретет клиент, и вероятность подобной покупки.

Колонки результата – таблица значений, состоящая из колонок, указанных в настройках результирующих колонок, и содержащая прогнозируемые данные. Конкретное содержимое определяется типом анализа.

14.1.2. Типы анализа данных

В механизме анализа данных и прогнозирования реализовано несколько типов анализа данных:

- общая статистика,
- поиск ассоциаций,
- поиск последовательностей,
- кластерный анализ,
- дерево решений.

14.1.2.1. Общая статистика

Тип анализа **Общая статистика** представляет собой механизм для сбора общей информации о данных, находящихся в полученном источнике данных. Этот тип анализа предназначен для предварительного исследования анализируемой информации.

Анализ показывает ряд характеристик дискретных и непрерывных полей. При выводе отчета в табличный документ заполняются круговые диаграммы для отображения состава полей.

14.1.2.2. Поиск ассоциаций

Данный тип анализа осуществляет поиск часто встречаемых вместе групп объектов или значений характеристик, а также производит поиск правил ассоциаций. Поиск ассоциаций может использоваться, например, для определения часто приобретаемых вместе товаров или услуг.

Этот тип анализа может работать с иерархическими данными, что позволяет, например, находить правила не только для конкретных товаров, но и для их групп. Важной особенностью этого типа анализа является возможность работать как с объектным источником данных, в котором каждая колонка содержит некоторую характеристику объекта, так и с событийным источником, где характеристики объекта располагаются в одной колонке.

14.1.2.3. Поиск последовательностей

Тип анализа **Поиск последовательностей** позволяет выявлять в источнике данных последовательные цепочки событий. Например, это может быть цепочка товаров или услуг, которые часто последовательно приобретают клиенты.

Этот тип анализа позволяет осуществлять поиск по иерархии, что дает возможность отслеживать не только последовательности конкретных событий, но и последовательности родительских групп.

14.1.2.4. Кластерный анализ

Кластерный анализ позволяет разделить исходный набор исследуемых объектов на группы объектов таким образом, чтобы каждый объект был более схож с объектами из своей группы, чем с объектами других групп. Анализируя в дальнейшем полученные группы, называемые кластерами, можно определить, чем характеризуется та или иная группа, принять решение о методах работы с объектами различных групп. Например, при помощи кластерного анализа можно разделить клиентов, с которыми работает компания, на группы, для того, чтобы применять различные стратегии при работе с ними.

При помощи параметров кластерного анализа пользователь может настроить алгоритм, по которому будет производиться разбиение, а также может динамически изменять состав характеристик, учитываемых при анализе, настраивать для них весовые коэффициенты.

Результат кластеризации может быть выведен в дендрограмму – специальный вид диаграммы, предназначенный для графического отображения результатов кластерного анализа.

14.1.2.5. Дерево решений

Тип анализа **Дерево решений** позволяет построить иерархическую структуру классифицирующих правил, представленную в виде дерева.

Для построения дерева решений необходимо выбрать целевой атрибут, по которому будет строиться классификатор, и ряд входных атрибутов, которые будут использоваться для создания правил. Целевой атрибут может содержать, например, информацию о том, перешел ли клиент к другому поставщику услуг, удачна ли была сделка, качественно ли была выполнена работа и т. д. Входными атрибутами, например, могут выступать возраст сотрудника, стаж его работы, материальное состояние клиента, количество сотрудников в компании и т. п.

Результат работы анализа представляется в виде дерева, каждый узел которого содержит некоторое условие. Для принятия решения, к какому классу следует отнести некий новый объект, необходимо, отвечая на вопросы в узлах, пройти цепочку от корня до листа дерева, переходя к дочерним узлам в случае утвердительного ответа и к соседнему узлу – в случае отрицательного.

Набор параметров анализа позволяет регулировать точность полученного дерева.

14.1.3. Модели прогноза

Модели прогноза, создаваемые механизмом, представляют собой специальные объекты, которые создаются из результата анализа данных и позволяют в дальнейшем автоматически выполнять прогноз для новых данных.

Например, модель прогноза поиска ассоциаций, построенная при анализе покупок клиентов, может быть использована при работе с осуществляющим покупку клиентом, для того чтобы предложить ему товары, которые он с определенной степенью вероятности приобретет вместе с выбранными им товарами.

14.2. Тип анализа «Общая статистика»

Тип анализа **Общая статистика** может использоваться для предварительного анализа данных (перед выполнением другого вида анализа) и т. п.

В качестве источника данных, которые будут подвергаться анализу, может использоваться результат запроса, таблица значений или область ячеек табличного документа.

Данные в источнике (с точки зрения проводимого анализа) могут иметь непрерывный или дискретный вид. К непрерывным относятся такие типы, как **Число**, **Дата**. Остальные типы относятся к дискретным.

Для колонок разных видов предусмотрено получение различной информации.

Дискретные данные:

- Количество значений – количество значений, встреченных в колонке источника данных (**NULL** значением не считается);
- Количество уникальных значений (с исключением повторяющихся значений);
- Мода – значение, которое в источнике данных встречается наиболее часто. Если в данных несколько значений, встречаемых с одинаковой частотой, в качестве моды берется первое найденное;
- Частота – количество вхождений значения в выборку данных;
- Относительная частота – определяется как отношение количества вхождения значения к общему количеству значений;
- Накопленная частота – считается как сумма частоты значения и сумма частот предыдущих значений выборки данных;
- Накопленная относительная частота – считается как сумма накопленной частоты значения и сумма относительных частот предыдущих значений.

Непрерывные данные:

- Количество значений;
- Минимум значения;
- Максимум значения;
- Среднее;
- Размах – разность между максимальным и минимальным значениями;
- Стандартное отклонение (среднеквадратичное отклонение);
- Медиана – значение, лежащее в середине выборки.

Следует отметить, что если анализируется одновременно несколько полей различных видов, их анализ проводится вне зависимости друг от друга (исключается взаимная корреляция).

Рассмотрим указанные характеристики на примере.

Выборка данных (источник анализа) имеет следующее наполнение:

Номенклатура	Количество
Стол кухонный раскладной	1
Табурет круглый	2
Диван «УЮТ»	1
Диван «Джинс»	1
Кресло «Джинс»	2
Стол «Kitchen» 0.9x1.7	1
Диван «Комфорт»	1
Стол «Kitchen» 0.9x1.7	1
Стул «Summer»	4
Диван «УЮТ»	1
Стол кухонный раскладной	1
Табурет прямоугольный	3
Кресло «УЮТ»	2
Кресло «УЮТ»	2
Шкаф «Wardrobe»	1
Стол кухонный раскладной	1
Табурет прямоугольный	2
Стол обеденный	1
Стул «Summer»	2
Табурет круглый	2

В результате анализа данных для поля **Количество** (вид данных анализа **Непрерывные**) будут рассчитаны следующие характеристики:

Характеристика	Значение
Значений	20
Минимум	1
Максимум	4
Среднее	1,6
Размах	3
Стандартное отклонение	0,8208
Медиана	1

Для поля **Номенклатура** будут получены следующие характеристики:

Характеристика	Значение
Количество значений	20
Количество уникальных значений	12
Мода	Стол кухонный раскладной

Таблица частот для значений номенклатуры будет иметь следующий вид:

Таблица частот

Значение	Частота	Относительная частота	Накопленная частота	Накопленная относительная частота
Стол кухонный раскладной	3	15,00	3	15,00
Табурет круглый	2	10,00	5	25,00
Табурет прямоугольный	2	10,00	7	35,00
Кресло "УЮТ"	2	10,00	9	45,00
Диван "УЮТ"	2	10,00	11	55,00
Стол "Kitchen" 0.9x1.7	2	10,00	13	65,00
Стул "Summer"	2	10,00	15	75,00
Стол обеденный	1	5,00	16	80,00
Шкаф "Wardrobe"	1	5,00	17	85,00
Диван "Комфорт"	1	5,00	18	90,00
Кресло "Джинс"	1	5,00	19	95,00
Диван "Джинс"	1	5,00	20	100,00

Рис. 305. Таблица частот

Относительная частота в виде диаграммы представлена далее:

Диаграмма частот



Рис. 306. Диаграмма частот

Для выполнения данного анализа может использоваться фрагмент кода, аналогичный приведенному:

[Копировать в буфер обмена](#)

```

&НаКлиенте
Процедура ОбщаяСтатистика(Команда)
    Результат = АнализОбщаяСтатистика();
КонецПроцедуры
&НаСервереБезКонтекста
Функция АнализОбщаяСтатистика()
    Анализ = Новый АнализДанных;
    Анализ.ТипАнализа = Тип("АнализДанныхОбщаяСтатистика");
    Запрос = Новый Запрос;
    Запрос.Текст = "
        |ВЫБРАТЬ
        |Продажи.Номенклатура,
        |Продажи.Количество
        |ИЗ
        |РегистрНакопления.Продажи КАК Продажи";
    Анализ.ИсточникДанных = Запрос.Выполнить();
    РезультатАнализа = Анализ.Выполнить();
    Построитель = Новый ПостроительОтчетаАнализаДанных();
    Построитель.Макет = Неопределено;
    Построитель.ТипАнализа = Тип("АнализДанныхОбщаяСтатистика");
    ТабДок = Новый ТабличныйДокумент;
    Построитель.Вывести(РезультатАнализа, ТабДок);
    Возврат ТабДок;
КонецФункции

```

Работа по проведению анализа данных выполняется в серверной внеконтекстной функции, которая возвращает на клиента табличный документ с результатами анализа. Сначала создается сам объект `АнализДанных`. После этого производится выбор типа проводимого анализа.

Далее по тексту определяется запрос. Результат запроса устанавливается как источник данных анализа. Сам анализ

выполняется в процессе работы метода `Выполнить()` объекта `АнализДанных`.

Сам анализ не имеет средств по визуализации результата полученного анализа. Для этой цели используется объект `ПостроительОтчетаАнализаДанных`. При создании данного объекта повторно указывается тип проводимого анализа. Далее в качестве первого параметра метода `Вывести()` передается результат полученного анализа, вторым параметром передается ранее созданный объект `ТабличныйДокумент`.

В конце алгоритма табличный документ с результатом анализа возвращается на клиента в реквизит обработки `Результат`, имеющий тип `ТабличныйДокумент`.

В результате будут получены данные, аналогичные рассмотренным выше.

14.3. Тип анализа «Поиск ассоциаций»

Как уже было сказано ранее, данный тип анализа осуществляет поиск часто встречаемых вместе комбинаций объектов или значений характеристик. С его помощью можно определять группы одновременно закупаемых товаров, выявлять наиболее привлекательные источники информации (в процессе «оптимизации» затрат на них) и т. п.

Схематично процесс проведения анализа `Поиск ассоциаций` можно представить следующим образом:



Рис. 307. Схема выполнения анализа «Поиск ассоциаций»

В качестве источника данных может использоваться результат запроса, таблица значений или область ячеек табличного документа. С точки зрения данного типа анализа колонки источника можно разделить на следующие:

- **НеИспользуемая** – игнорируются анализом;
- **Объект** – данные из этой колонки используются как объекты (или события) проводимого анализа. Исходя из значения данной колонки, значения другой колонки (**Элемент**) относятся к одной ассоциируемой группе;
- **Элемент** – данные из этой колонки используются для получения устойчивых групп значений, построения ассоциативных правил.

Кроме настройки типов колонок, на результат проводимого анализа влияют следующие параметры анализа:

- **МинимальныйПроцентСлучаев** – определяется минимальный процент случаев, в которых должна встречаться комбинация

элементов. Группы, у которых данное значение меньше указанного, не попадают в результат анализа;

- **МинимальнаяДостоверность** – показывает минимальное значение процента случаев, когда правило соблюдается;
- **МинимальнаяЗначимость** – группы, у которых данное значение меньше указанного, не попадают в результат анализа;
- **ТипОтсеченияПравил** – один из вариантов системного перечисления **ТипОтсеченияПравилАссоциации**:
 - **Избыточные** – отсекают избыточные правила,
 - **Покрытые** – отсекают правила, покрытые другими правилами.

В результате выполнения анализа получаем:

- информацию о данных (количество объектов, количество элементов, среднее количество элементов в объекте, количество найденных групп, количество найденных правил ассоциаций);
- найденные группы элементов – указывается состав группы, количество случаев, процент случаев, в которых эта группа встречается;
- найденные ассоциативные правила – указывается исходный состав элементов, следствие (состав элементов), процент случаев, достоверность, значимость правила.

Рассмотрим особенности проведения данного типа анализа на следующей выборке данных (постараемся определить состав одновременно закупаемых товаров):

Регистратор	Номенклатура
Расходная накладная № 000000001	Стол кухонный раскладной
	Табурет круглый
Расходная накладная № 000000002	Диван «УЮТ»
Расходная накладная № 000000003	Диван «Джинс»
	Кресло «Джинс»
Расходная накладная № 000000005	Стол «Kitchen» 0.9x1.7
	Диван «Комфорт»
Расходная накладная № 000000004	Стол «Kitchen» 0.9x1.7
	Стул «Summer»
	Диван «УЮТ»
Расходная накладная № 000000006	Стол кухонный раскладной
	Табурет прямоугольный
Расходная накладная № 000000007	Кресло «УЮТ»
Расходная накладная № 000000008	Кресло «УЮТ»
Расходная накладная № 000000009	Шкаф «Wardrobe»
Расходная накладная № 000000010	Стол кухонный раскладной
	Табурет прямоугольный
	Стол обеденный
Расходная накладная № 000000011	Стул «Summer»
	Табурет круглый

Признаком, по которому данные относятся к одной группе, будем считать значение регистратора (номенклатура, указанная в одном документе, считается закупленной одновременно). То есть **Регистратор** будет объектом анализа, **Номенклатура** – элементом.

Для проведения анализа будем использовать следующий фрагмент кода:

[Копировать в буфер обмена](#)

```
&НаКлиенте
Процедура ПоискАссоциаций(Команда)
    Результат = АнализПоискАссоциаций();
КонецПроцедуры
&НаСервереБезКонтекста
Функция АнализПоискАссоциаций()
    Анализ = Новый АнализДанных;
```

```

Анализ.ТипАнализа = Тип("АнализДанныхПоискАссоциаций");
Запрос = Новый Запрос;
Запрос.Текст = "
|ВЫБРАТЬ
|Продажи.Регистратор,
|Продажи.Номенклатура
|ИЗ
|РегистрНакопления.Продажи КАК Продажи";
Анализ.ИсточникДанных = Запрос.Выполнить();
// Строка приводится в качестве примера,
// такое значение типа колонки по умолчанию.
Анализ.НастройкаКолонок.Номенклатура.ТипКолонки = ТипКолонкиАнализаДанныхПоискАссоциаций.Элемент;
// Строка приводится в качестве примера,
// такое значение типа отсеечения по умолчанию.
Анализ.Параметры.ТипОтсеченияПравил.Значение = ТипОтсеченияПравилАссоциации.Избыточные;
РезультатАнализа = Анализ.Выполнить();
Построитель = Новый ПостроительОтчетаАнализаДанных();
Построитель.Макет = Неопределено;
Построитель.ТипАнализа = Тип("АнализДанныхПоискАссоциаций");
ТабДок = Новый ТабличныйДокумент;
Построитель.Вывести(РезультатАнализа, ТабДок);
Возврат ТабДок;
КонецФункции

```

Результат анализа будет выглядеть следующим образом:

Информация о данных

Количество элементов:	12
Количество объектов:	11
Среднее количество элементов в объекте:	1,82

Результат анализа

Найдено часто встречаемых групп:	1
Найдено ассоциативных правил:	2

Рис. 308. Результат анализа «Поиск ассоциаций»

В выборе используются данные из одиннадцати документов (ссылка содержится в поле [Регистратор](#)), количество различных номенклатурных позиций равно двенадцати:

Номенклатура
Стол кухонный раскладной
Табурет круглый
Диван «УЮТ»
Диван «Джинс»
Кресло «Джинс»
Стол «Kitchen» 0.9x1.7
Диван «Комфорт»
Стул «Summer»
Табурет прямоугольный
Кресло «УЮТ»
Шкаф «Wardrobe»
Стол обеденный

Найдена следующая группа товаров:

Часто встречаемые группы

№	Количество случаев	Процент случаев
Состав		
1	2	18,18
Номенклатура = Табурет прямоугольный		
Номенклатура = Стол кухонный раскладной		

Рис. 309. Найденная группа товаров

Вся группа встречается в документе только в двух случаях из одиннадцати (это и показано в колонках [Количество случаев](#) и [Процент случаев](#)).

Получены следующие ассоциативные правила:

Ассоциативные правила

Исходный набор	Следствие	Процент случаев	Достоверность	Значимость
1		18,18	100,00	3,67
Номенклатура = Табурет прямоугольный	Номенклатура = Стол кухонный раскладной			
2		18,18	66,67	3,67
Номенклатура = Стол кухонный раскладной	Номенклатура = Табурет прямоугольный			

Рис. 310. Ассоциативные правила

Разберем второе из них. В двух случаях из одиннадцати в документе вместе с позицией **Стол кухонный раскладной** встречалась позиция **Табурет прямоугольный**. Исходя из этого, был рассчитан процент случаев: $(2 / 11 * 100 = 18,18 \%)$.

Достоверность была рассчитана следующим образом: обе номенклатурные позиции закупались в двух случаях, товарная позиция **Стол кухонный раскладной** встречалась в покупках 3 раза. Исходя из этого, достоверность равна: $2 / 3 * 100 = 66,67 \%$.

Значимость определяется как отношение достоверности правила к проценту нахождения **Табурет прямоугольный** в закупаемых товарах. Эта позиция встречается в двух документах из одиннадцати (18,18 %). Значимость равна: $66,67 \% / 18,18 \% = 3,67$.

14.3.1. Типы отсеечения правил

Рассмотрим такой важный параметр данного типа анализа, как **ТипОтсееченияПравил**. Возможные значения отсеечения содержатся в системном перечислении **ТипОтсееченияПравилАссоциации**, их состав:

- **Покрытые**,
- **Избыточные**.

Перед тем как перейти к рассмотрению особенностей вариантов отсеечения, рассмотрим несколько общих моментов, применимых к правилам ассоциации.

Любое правило состоит из **предпосылки** и **следствия**. Например:

- Предпосылка: **Если** купили Товар № 1.
- Следствие: **То** купят Товар № 2.

При этом не нужно забывать, что следствие наступает с определенной достоверностью. При отсечении правил вероятностные характеристики могут рассматриваться, а могут и игнорироваться (важно только содержание правила).

14.3.1.1. Отсечение покрытых правил

Рассмотрим вариант отсеечения **Покрытые**.

Правило может быть покрыто как по предпосылке, так и по следствию. Например:

- Правило № 1: **Если** купили товар № 1 и № 3, **То** купят товар № 2.
- Правило № 2: **Если** купили товар № 1, **То** купят товар № 2.

В этом случае правило № 1 будет считаться покрытым, т. к. предпосылка первого правила получается «избыточной» по отношению к предпосылке второго правила.

Пример покрытия по следствию:

- Правило № 1: **Если** купили товар № 1, **То** купят товар № 2, № 3.
- Правило № 2: **Если** купили товар № 1, **То** купят товар № 3.

Правило № 2 будет покрыто по следствию, так как следствие правила № 1 более полное.

14.3.1.2. Отсечение избыточных правил

Покрытие не учитывает вероятностных характеристик правил, они учитываются в случае, если используется вариант отсеечения **Избыточные**.

Правило будет считаться избыточным по предпосылке, если оно покрыто по предпосылке и его достоверность равна достоверности покрывающего правила. Например:

- Правило № 1: **Если** купили товар № 1 и № 3, **То** купят товар № 2 с достоверностью 75 %.

- Правило № 2 **Если** купили товар № 1, **То** купят товар № 2 с достоверностью 75 %.

Правило № 1 избыточно по отношению к правилу № 2 (оно содержит дополнительное условие, не вносящее «возмущения» в вероятностные характеристики правила).

Правило № 1 будет считаться избыточным по следствию, если количество случаев данного правила равно количеству случаев покрывающего правила.

- Правило № 1: **Если** купили товар № 1, **То** купят товар № 2, № 3 в трех случаях.

- Правило № 2: **Если** купили товар № 1, **То** купят товар № 3 в трех случаях.

Правило № 2 будет считаться избыточным по отношению к правилу № 1, так как оно содержит более простое следствие с теми же вероятностными характеристиками.

14.4. Тип анализа «Поиск последовательностей»

Данный тип анализа позволяет выявить цепочки возникающих событий (шаблоны последовательностей). Он может использоваться тогда, когда одним из важных анализируемых показателей является последовательность наступления событий во времени. Например, можно выявить последовательность товаров, которые покупаются друг за другом в течение какого-либо определенного промежутка времени и т. п.

Схематично процесс проведения анализа [Поиск последовательностей](#) можно представить следующим образом:



Рис. 311. Схема выполнения анализа «Поиск последовательностей»

В качестве источника данных может использоваться результат запроса, таблица значений, область ячеек табличного

документа. С точки зрения данного типа анализа колонки источника можно разделить на следующие:

- **НеИспользуемая** – игнорируются анализом;
- **Последовательность** – данные из этой колонки используются для анализа как объект события последовательности. По значению данной колонки анализ и ассоциирует данные с одной цепочкой событий;
- **Элемент** – данные из этой колонки используются как элементы последовательности;
- **Время** – именно по данной колонке определяется время наступления события. Наличие данной колонки обязательно при проведении данного типа анализа.

Кроме настройки типов колонок, на результат проводимого анализа влияют следующие параметры анализа:

- **МинимальныйПроцентСлучаев** – минимальный процент последовательностей, в которых наблюдается найденный шаблон последовательности;
- **МинимальныйИнтервал** – признак установки минимального интервала последовательности (должна быть установлена единица измерения интервала, кратность);
- **МаксимальныйИнтервал** – признак установки максимального интервала последовательности (должна быть установлена единица измерения интервала, кратность);
- **ИнтервалЭквивалентностиВремени** – признак установки интервала эквивалентности времени (должна быть установлена единица интервала эквивалентности времени, ее кратность);
- **МинимальнаяДлина** – минимальная длина искомым последовательностей;
- **ПоискПоИерархии** – признак осуществления поиска по иерархии (распространяется на колонки с типом **Элемент**).

Ряд свойств оперирует таким понятием, как **ТипЕдиницыИнтервалаВремениАнализаДанных**. Данное системное перечисление содержит следующие значения:

Секунда	
Минута	ТекущаяМинута
Час	ТекущийЧас
День	ТекущийДень
Неделя	ТекущаяНеделя
Декада	ТекущаяДекада
Месяц	ТекущийМесяц
Квартал	ТекущийКвартал
Полугодие	ТекущееПолугодие
Год	ТекущийГод

Основным результатом анализа являются найденные шаблоны последовательностей. Эти шаблоны содержат следующую информацию:

- состав шаблона последовательности;
- количество случаев, в которых наблюдалась данная последовательность;
- максимальные интервалы между событиями (в случае, если событий 2, то интервал один);
- минимальные интервалы между событиями (в случае, если событий 2, то интервал один);
- процент случаев, когда данная последовательность выполнялась;
- средние интервалы между событиями (в случае, если событий 2, то интервал один).

Особенности выполнения данного типа анализа рассмотрим на примере следующей выборки данных:

Контрагент	Первая покупка	Вторая покупка	Третья покупка	Интервал
Бондарев В.И.	Стол кухонный раскладной	Диван «УЮТ»	Кресло «УЮТ»	25 дней, 31 день
	Табурет круглый			
Иванов И.П.	Диван «Джинс»			

	Кресло «Джинс»			
Петров Б.С.	Стол «Kitchen» 0.9x1.7	Кресло «УЮТ»		43 дня
	Стул «Summer»			
	Диван «УЮТ»			
Сидоров Г.О.	Стол кухонный раскладной			
	Табурет прямоугольный			
Степанов В.К.	Стол кухонный раскладной			
	Табурет прямоугольный			
	Стол обеденный			
Федоров Д.Е.	Стол «Kitchen» 0.9x1.7	Шкаф «Wardrobe»	Стул «Summer»	58 дней, 29 дней
	Диван «Комфорт»		Табурет круглый	

Данные из колонки **Контрагент** будут определять принадлежность к конкретной цепочке событий, т. е. они определяют последовательность анализа. **Номенклатура** будет являться элементом получаемой последовательности.

Для проведения анализа может использоваться фрагмент кода, аналогичный приведенному ниже:

[Копировать в буфер обмена](#)

```
&НаКлиенте
Процедура ПоискПоследовательностей(Команда)
    Результат = АнализПоискПоследовательностей();
КонецПроцедуры
&НаСервереБезКонтекста
Функция АнализПоискПоследовательностей()
    Анализ = Новый АнализДанных;
    Анализ.ТипАнализа = Тип("АнализДанныхПоискПоследовательностей");
    Запрос = Новый Запрос;
    Запрос.Текст = "
        |ВЫБРАТЬ
        |Продажи.Контрагент,
        |Продажи.Номенклатура,
        |Продажи.Период
        |ИЗ
        |РегистрНакопления.Продажи КАК Продажи";
    Анализ.ИсточникДанных = Запрос.Выполнить();
    Анализ.НастройкаКолонки.Период.ТипКолонки = ТипКолонкиАнализаДанныхПоискПоследовательностей.Время;
    РезультатАнализа = Анализ.Выполнить();
    Построитель = Новый ПостроительОтчетаАнализаДанных();
    Построитель.Макет = Неопределено;
    Построитель.ТипАнализа = Тип("АнализДанныхПоискПоследовательностей");
    ТабДок = Новый ТабличныйДокумент;
    Построитель.Вывести(РезультатАнализа, ТабДок);
    Возврат ТабДок;
КонецФункции
```

Поле **Период** определяется как **Время** непосредственно из кода (самостоятельно анализом не определяется).

Параметры анализа, установленные по умолчанию:

Параметры анализа

Минимальный процент случаев:	10
Минимальный интервал:	---
Максимальный интервал:	---
Интервал эквивалентности времени:	---
Минимальная длина последовательности:	2

Рис. 312. Параметры анализа

После проведения анализа получены следующие общие данные:

Информация о данных

Количество элементов:	12
Количество последовательностей:	6

Результат анализа

Найдено последовательностей:	2
------------------------------	---

Рис. 313. Общие данные анализа

Количество элементов равно двенадцати. Ровно столько номенклатурных позиций встречается в приведенной выборке данных.

Найдены две последовательности:

Последовательности

№	Количество случаев	Процент случаев	Средний интервал	Минимальный интервал	Максимальный интервал
Состав					
1	2	33,33			
Номенклатура = Диван "УЮТ"					
Номенклатура = Кресло "УЮТ"					
2	1	16,67	1 мес. 7 д.	1 мес. 1 д.	1 мес. 13 д.
Номенклатура = Стол "Kitchen" 0.9x1.7					
Номенклатура = Стул "Summer"					
			2 мес. 28 д.	2 мес. 28 д.	2 мес. 28 д.

Рис. 314. Найденные последовательности

Первая последовательность встречается в двух случаях из пяти. Исходя из этого, процент случаев равен 40 %. Так как глубина последовательности равна 2, существует по одному значению каждого из приводимых интервалов.

14.5. Тип анализа «Дерево решений»

С помощью данного типа анализа можно получить причинно-следственную иерархию условий, приводящую к определенному решению. Например, получить дерево условий, по которому (с определенной долей вероятности) можно понять причину расторжения договоров с клиентами компании, определения условий, влияющих на вариант заключаемого договора. Можно проводить «профилирование» менеджеров компании по различным видам ее клиентов и т. п.

Схематично процесс проведения анализа [Дерево решений](#) можно представить следующим образом:

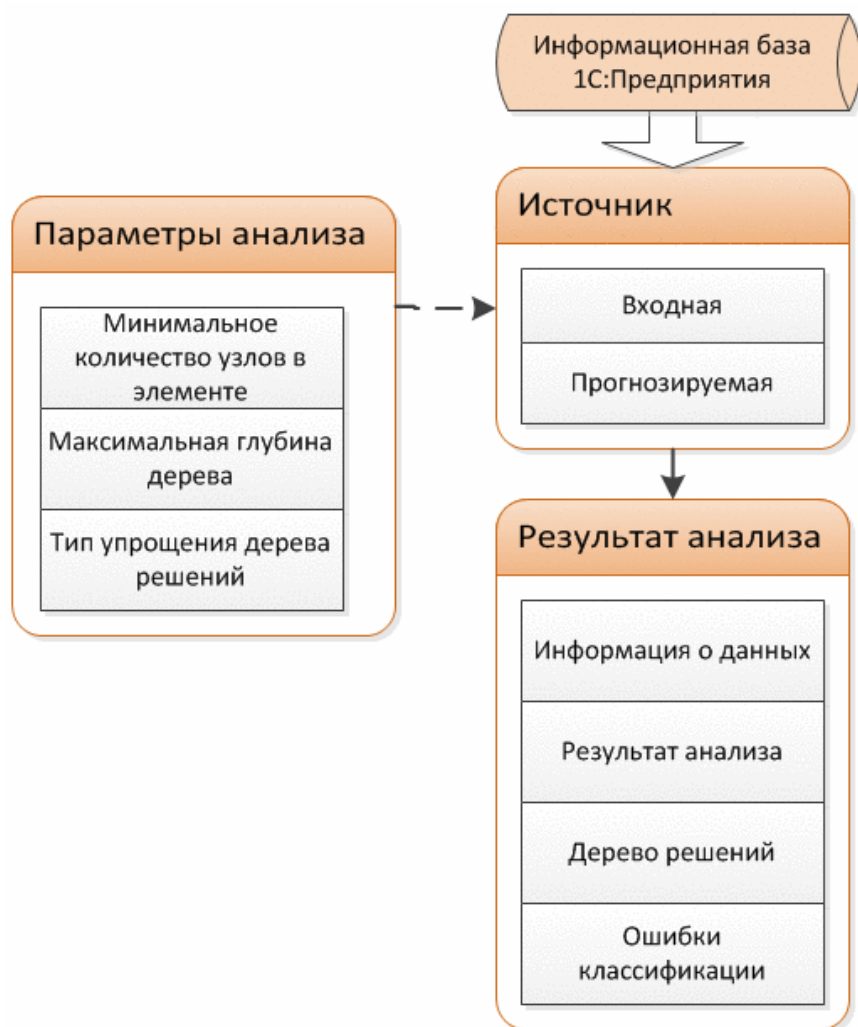


Рис. 315. Схема выполнения анализа «Дерево решений»

С точки зрения данного типа анализа колонки источника можно разделить на следующие:

- НеИспользуемая,
- Входная,

- Прогнозируемая.

Используемые параметры анализа:

- МинимальноеКоличествоСлучаев – минимальное количество элементов в узле;
- МаксимальнаяГлубина – максимальная глубина дерева;
- ТипУпрощения – тип упрощения дерева решений.

В результате проведенного анализа можно получить:

- дерево решений,
- ошибки классификации.

Разберемся с особенностью данного типа анализа на примере следующей выборки данных:

Контрагент	Количество розничных точек	Количество автомобилей	Время работы организации	Время заключения договора	Вид договора	Состояние взаимоотношений
ЗАО Игорь	1	0	Меньше года	Меньше года	Дилер	Несоблюдение договора
ЗАО ТоргМебель	15	4	От трех до десяти лет	Меньше года	Дистрибьютор	Прекращение контрагентом
ЗАО ТоргМебель	1	10	От трех до десяти лет	От года до трех	Дистрибьютор	Прекращение контрагентом
ИЧП Дубрава	1	1	От года до трех	Меньше года	Дилер	Прекращение контрагентом
Магазин № 15	1	1	Свыше десяти лет	От трех до десяти лет	Постоянный партнер	Не прекращены
ООО Гросс	3	2	Меньше года	Меньше года	Постоянный партнер	Не прекращены
ООО Интарис	7	3	От трех до десяти лет	От года до трех	Постоянный партнер	Прекращение контрагентом
ООО ТоргТрест	2	2	Свыше десяти лет	От трех до десяти лет	Постоянный партнер	Не прекращены
ПБОЮЛ Курочкин	0	1	Меньше года	Меньше года	Дилер	Не прекращены

Для проведения анализа может использоваться фрагмент кода, аналогичный приведенному ниже:

[Копировать в буфер обмена](#)

```
&НаКлиенте
```

```
Процедура ДеревоРешений(Команда)
```

```
    Результат = АнализДеревоРешений();
```

```
КонецПроцедуры
```

```
&НаСервереБезКонтекста
```

```
Функция АнализДеревоРешений()
```

```
    Анализ = Новый АнализДанных;
```

```
    Анализ.ТипАнализа = Тип("АнализДанныхДеревоРешений");
```

```
    Группа = Справочники.Контрагенты.НайтиПоНаименованию("Юридические лица");
```

```
    Запрос = Новый Запрос;
```

```
    Запрос.Текст = "
```

```
    |ВЫБРАТЬ
```

```
    |Контрагенты.Ссылка,
```

```
    |Контрагенты.КоличествоРозничныхТочек,
```

```
    |Контрагенты.КоличествоАвтомобилей,
```

```
    |Контрагенты.ВремяРаботыОрганизации,
```

```
    |Контрагенты.ВремяЗаключенияДоговора,
```

```
    |Контрагенты.ВидДоговора,
```

```
    |Контрагенты.ПрекращениеОтношений
```

```
    ИЗ
```

```
    |Справочник.Контрагенты КАК Контрагенты
```

```
    |ГДЕ
```

```
    |(НЕ Контрагенты.ЭтоГруппа И Контрагенты.Родитель = &Родитель)";
```

```
    Запрос.УстановитьПараметр("Родитель", Группа);
```

```
    Анализ.ИсточникДанных = Запрос.Выполнить();
```

```
    Анализ.Параметры.ТипУпрощения.Значение = ТипУпрощенияДереваРешений.НеУпрощать;
```

```
    РезультатАнализа = Анализ.Выполнить();
```

```
    Построитель = Новый ПостроительОтчетаАнализаДанных();
```

```
    Построитель.Макет = Неопределено;
```

```

Построитель.ТипАнализа = Тип("АнализДанныхДеревоРешений");
ТабДок = Новый ТабличныйДокумент;
Построитель.Вывести(РезультатАнализа, ТабДок);
Возврат ТабДок;
КонецФункции

```

В результате проведения анализа получено следующее дерево решений:

Дерево решений

Условие	Решение	Вероятность	Количество
Корень	ПреращениеОтношений = Не прекращены	44,44	9
КоличествоАвтомобилей <= 0,5	ПреращениеОтношений = Несоблюдение договора	100,00	1
КоличествоАвтомобилей > 0,5	ПреращениеОтношений = Не прекращены	50,00	8
КоличествоАвтомобилей <= 2,5	ПреращениеОтношений = Не прекращены	80,00	5
ВремяРаботыОрганизации = Свыше десяти лет	ПреращениеОтношений = Не прекращены	100,00	2
ВремяРаботыОрганизации = Меньше года	ПреращениеОтношений = Не прекращены	100,00	2
ВремяРаботыОрганизации = От года до трех	ПреращениеОтношений = Прекращение контрагентом	100,00	1
КоличествоАвтомобилей > 2,5	ПреращениеОтношений = Прекращение контрагентом	100,00	3

Рис. 316. Дерево решений

Данное дерево можно представить в виде следующей схемы:

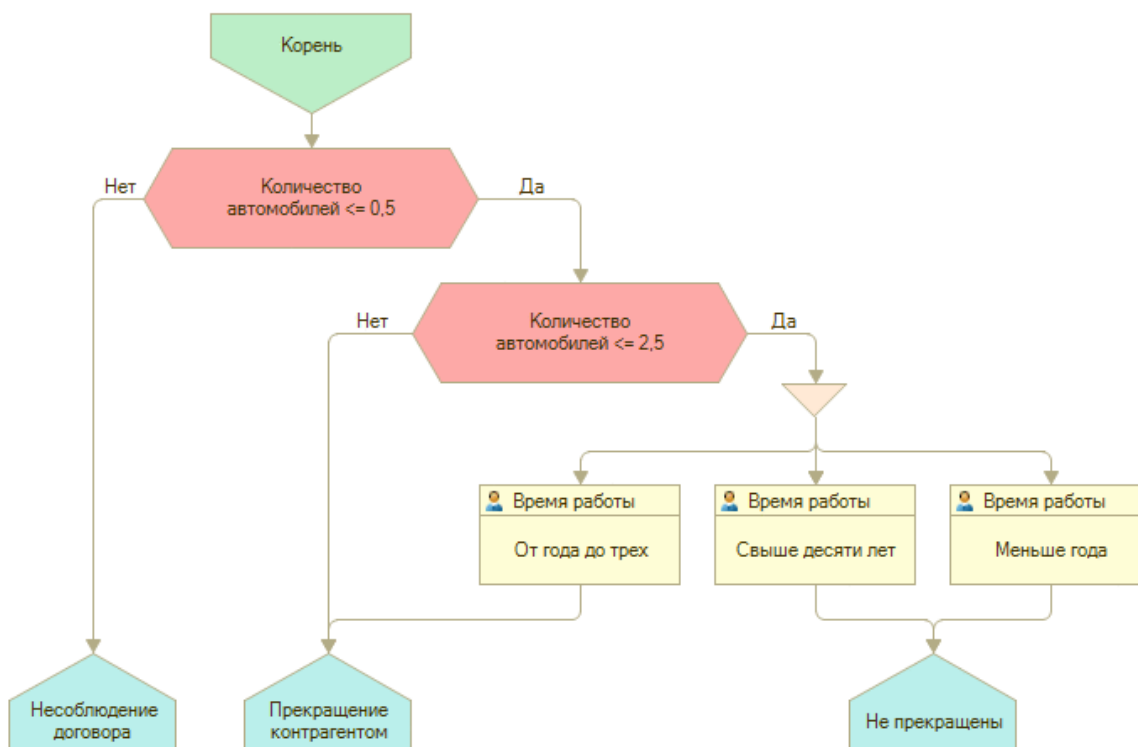


Рис. 317. Представление дерева решения в виде схемы

Ошибки классификации показывают, в каких случаях полученные правила расходятся с действительностью (исходной выборкой данных):

Ошибки классификации

Факт\Классифицировано	Прекращение Отношений = Не прекращены	Прекращение Отношений = Несоблюдение договора	Прекращение Отношений = Прекращение контрагентом	Итого	% Ошибки
Прекращение Отношений = Не пр	4			4	
Прекращение Отношений = Несоб		1		1	
Прекращение Отношений = Прекр			4	4	
Итого	4	1	4	9	

Рис. 318. Ошибки классификации

Исходя из приведенных данных, видно, что ошибок в полученной классификации нет, то есть данные в фактической выборке совпадают с данными классификации.

Предыдущий пример получен, исходя из значения `НеУпрощать` параметра анализа `ТипУпрощения`. Данное значение параметра задано программно в примере выше. Если установить значение `Упрощать` параметра, дерево решений примет такой вид:

Дерево решений

Условие	Решение	Вероятность	Количество
Корень	Прекращение Отношений = Не прекращены	44,44	9
КоличествоАвтомобилей <= 0,5	Прекращение Отношений = Несоблюдение договора	100,00	1
КоличествоАвтомобилей > 0,5	Прекращение Отношений = Не прекращены	50,00	8
КоличествоАвтомобилей <= 2,5	Прекращение Отношений = Не прекращены	80,00	5
КоличествоАвтомобилей > 2,5	Прекращение Отношений = Прекращение контрагентом	100,00	3

Рис. 319. Дерево решений

Упрощение дерева заключается в том, что по определенным правилам (формулам, которые будут рассмотрены ниже) узлы дерева превращаются в листья (отсекается лишнее ветвление).

При принятии решения о том, будет ли произведено преобразование узла в лист, учитываются следующие показатели:

- **Ошибок** – количество ошибок в узле;
- **ОшибокДочерних** – количество ошибок в дочерних узлах;
- **Листов** – количество листьев в узле;
- **Случаев** – количество случаев.

$$СКОшибки = \sqrt{\frac{(\text{ОшибокДочерних} + \frac{\text{Листов}}{2}) * (\text{Случаев} * (\text{ОшибокДочерних} + \frac{\text{Листов}}{2}))}{\text{Случаев}}}$$

Решение о превращении узла в лист принимается в случае выполнения условия:

$$\text{Ошибок} + 0,5 < \text{ОшибокДочерних} + \frac{\text{Листов}}{2} + СКОшибки$$

В нашем примере для узлов [Время работы организации](#) условие выполняется ($0,5 < 1$).

В связи с использованием упрощений появились и ошибки в классификации, что показано в результате анализа:

Ошибки классификации

Факт\Классифицировано	ПреграшениеОтношений = Не прекращены	ПреграшениеОтношений = Несоблюдение договора	ПреграшениеОтношений = Прекращение контрагентом	Итого	% Ошибки
ПреграшениеОтношений = Не пр	4			4	
ПреграшениеОтношений = Несоб		1		1	
ПреграшениеОтношений = Прекр	1		3	4	25,00
Итого	5	1	3	9	11,11

Рис. 320. Ошибки классификации

Например, существует один случай, когда в реальной выборке данных было значение [Прекращение контрагентом](#), а по полученной классификации должно быть значение [Не прекращены](#) и т. п.

14.6. Тип анализа «Кластеризация»

Кластерный анализ – математическая процедура многомерного анализа, позволяющая на основе множества показателей, характеризующих ряд объектов, сгруппировать их в кластеры таким образом, чтобы объекты, входящие в один кластер, были более однородными, сходными, по сравнению с объектами, входящими в другие кластеры.

В основе данного анализа лежит вычисление расстояния между объектами. Именно исходя из расстояний между объектами и производится их группировка по кластерам. Определение расстояния может проводиться разными способами (по разным метрикам). Поддерживаются следующие метрики:

- Евклидова метрика,
- Евклидова метрика в квадрате,
- Метрика города,
- Метрика доминирования.

После определения расстояний между объектами может использоваться один из нескольких алгоритмов распределения объектов по кластерам. Поддерживаются следующие методы кластеризации:

- Ближняя связь,
- Дальняя связь,
- k-средних,
- Центр тяжести.

Схематично механизм проведения кластерного анализа можно представить следующим образом:

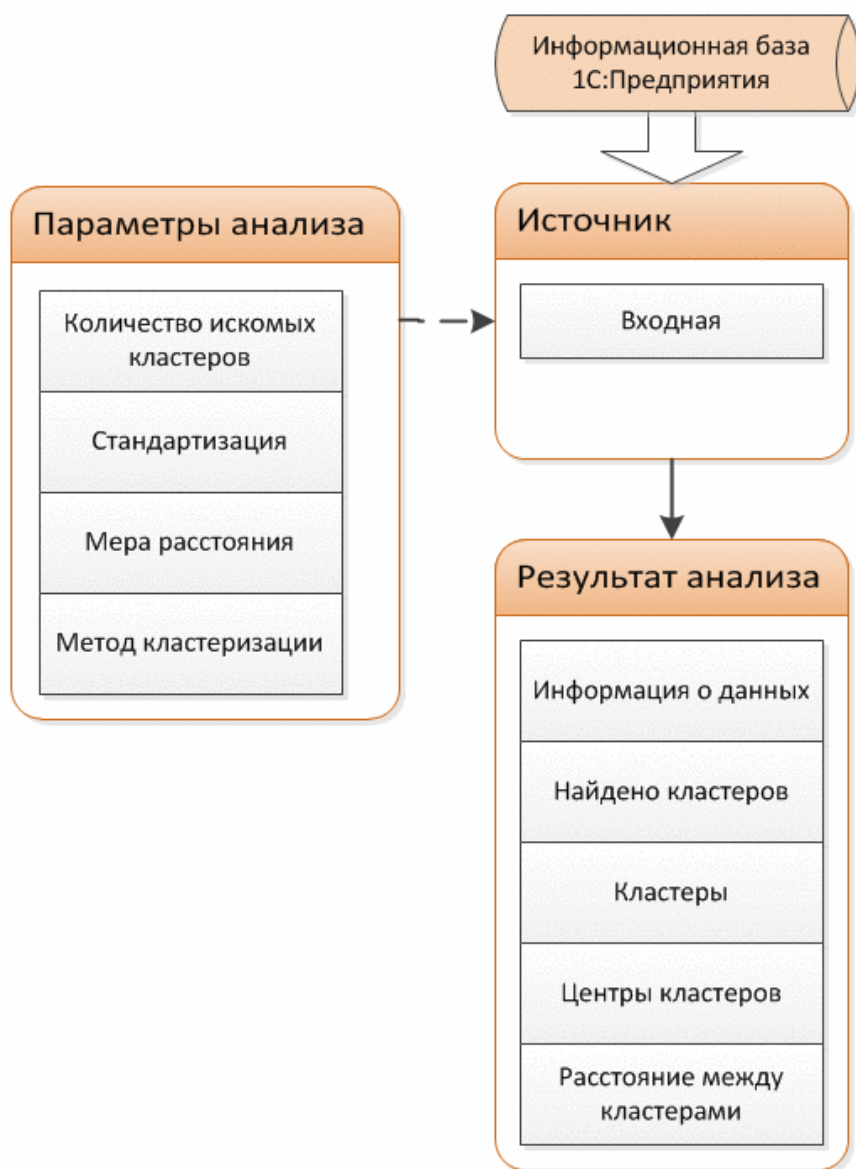


Рис. 321. Схема выполнения кластерного анализа

На вход объекту `АнализДанных` подается источник данных. В качестве источника может выступать результат запроса, таблица значений, область ячеек табличного документа. Колонки источника определяются как входные либо неиспользуемые. Следует отметить, что все значения колонок содержатся в системном перечислении `ТипКолонкиАнализаДанныхКластеризация`. В этом перечислении значений больше (не только неиспользуемые и входные), но другие значения используются при построении прогнозов.

Анализ производится в соответствии с установленными параметрами анализа.

В качестве примера, иллюстрирующего возможность проведения кластерного анализа, будем использовать следующий фрагмент кода:

```

&НаКлиенте
Процедура КластерныйАнализ(Команда)
    Результат = АнализКластеризация();
КонецПроцедуры
&НаСервереБезКонтекста
Функция АнализКластеризация()
    Анализ = Новый АнализДанных;
    Анализ.ТипАнализа = Тип("АнализДанныхКластеризация");
    Группа = Справочники.Контрагенты.НайтиПоНаименованию("Юридические лица");
    Запрос = Новый Запрос;
    Запрос.Текст = "
        ВЫБРАТЬ
        | Контрагенты.Ссылка,
        | Контрагенты.КоличествоРозничныхТочек,
        | Контрагенты.КоличествоАвтомобилей,
        | Контрагенты.ВремяРаботыОрганизации,
        | Контрагенты.ВремяЗаклученияДоговора,
        | Контрагенты.ВидДоговора,
        | Контрагенты.ПрекращениеОтношений
        ИЗ
        | Справочник.Контрагенты КАК Контрагенты
  
```

```

ГДЕ
(НЕ Контрагенты.ЭтоГруппа И Контрагенты.Родитель = &Родитель)";
Запрос.УстановитьПараметр("Родитель", Группа);
Анализ.ИсточникДанных = Запрос.Выполнить();
// Выбор метрики.
Анализ.Параметры.МераРасстояния.Значение =
ТипМерыРасстоянияАнализаДанных.ЕвклидоваМетрикаВКвадрате;
// Выбор метода кластеризации.
Анализ.Параметры.МетодКластеризации.Значение = МетодКластеризации.КСредних;
РезультатАнализа = Анализ.Выполнить();
Построитель = Новый ПостроительОтчетаАнализаДанных();
Построитель.Макет = Неопределено;
Построитель.ТипАнализа = Тип("АнализДанныхКластеризация");
ТабДок = Новый ТабличныйДокумент;
Построитель.Вывести(РезультатАнализа, ТабДок);
Возврат ТабДок;
КонецФункции

```

Запрос выполняется по справочнику **Контрагенты**. По условию запроса выбираются только детальные записи справочника из группы **Юридические лица**.

Выполнение указанного кода приведет к тому, что в качестве начальных установок анализа данных будут определены следующие значения (часть установлена явно, часть – по умолчанию):

Параметры анализа

Количество искомых кластеров:	3
Стандартизация:	Стандартизировать
Мера расстояния:	Евклидова метрика в квадрате
Метод кластеризации:	Метод центра тяжести

Колонки источника данных

Входные колонки

Имя колонки	Тип данных	Вес
Ссылка	Дискретный	1
КоличествоРозничныхТочек	Непрерывный	1
КоличествоАвтомобилей	Непрерывный	1
ВремяРаботыОрганизации	Дискретный	1
ВремяЗаключенияДоговора	Дискретный	1
ВидДоговора	Дискретный	1
ПрекращениеОтношений	Дискретный	1

Рис. 322. Параметры анализа

Состав колонок определился, исходя из состава полей выборки запроса. По умолчанию они определены с равным весом. Для типов **Число** и **Дата** определен вид данных **Непрерывные**, для остальных типов – **Дискретные**. При необходимости изменить параметры колонок это можно сделать по аналогии с приведенным фрагментом:

Копировать в буфер обмена

```
Анализ.НастройкаКолонок.КоличествоАвтомобилей.ДополнительныеПараметры.Вес = 2;
```

В данной строке для колонки **КоличествоАвтомобилей** увеличен вес.

Выборка данных, для которых будет произведен анализ, имеет следующее наполнение:

Контрагент	Количество розничных точек	Количество автомобилей	Время работы организации	Время заключения договора	Вид договора	Состояние взаимоотношений
ЗАО Игорь	1	0	Меньше года	Меньше года	Дилер	Несоблюдение договора
ЗАО ТоргМебель	15	4	От трех до десяти лет	Меньше года	Дистрибьютор	Прекращение контрагентом
ЗАО ТоргМебель	1	10	От трех до десяти лет	От года до трех	Дистрибьютор	Прекращение контрагентом
ИЧП Дубрава	1	1	От года до трех	Меньше года	Дилер	Прекращение контрагентом
Магазин № 15	1	1	Свыше десяти лет	От трех до десяти лет	Постоянный партнер	Не прекращены
ООО Гросс	3	2	Меньше года	Меньше года	Постоянный партнер	Не прекращены
ООО Интарис	7	3	От трех до десяти лет	От года до трех	Постоянный партнер	Прекращение контрагентом

ООО ТоргТрест	2	2	Свыше десяти лет	От трех до десяти лет	Постоянный партнер	Не прекращены
ПБОЮЛ Курочкин	0	1	Меньше года	Меньше года	Дилер	Не прекращены

Результат анализа будет получен в следующем виде:

Информация о данных

Количество объектов: 9

Результат анализа

Найдено кластеров: 3

Кластеры

№ кластера	Количество объектов	Процент
1	6	66,67
2	2	22,22
3	1	11,11

Центры кластеров

Поле \ Кластер	1	2	3
Ссылка = ЗАО ТоргМебель			1,0000
Ссылка = ЗАО МебельКрафт		0,5000	
Ссылка = ООО Интарис		0,5000	
Ссылка = ПБОЮЛ Курочкин	0,1667		
Ссылка = ООО ТоргТрест	0,1667		
Ссылка = Магазин № 15	0,1667		
Ссылка = ЗАО Игорьь	0,1667		
Ссылка = ООО Гросс	0,1667		
Ссылка = ИЧП Дубрава	0,1667		
КоличествоРозничныхТочек	1,3333	11,0000	1,0000
КоличествоАвтомобилей	1,1667	3,5000	10,0000
ВремяРаботыОрганизации = От года до трех	0,1667		
ВремяРаботыОрганизации = Меньше года	0,5000		
ВремяРаботыОрганизации = Свыше десяти лет	0,3333		
ВремяРаботыОрганизации = От трех до десяти лет		1,0000	1,0000
ВремяЗаклученияДоговора = От трех до десяти лет	0,3333		
ВремяЗаклученияДоговора = Меньше года	0,6667	0,5000	
ВремяЗаклученияДоговора = От года до трех		0,5000	1,0000
ВидДоговора = Постоянный партнер	0,5000	0,5000	
ВидДоговора = Дилер	0,5000		
ВидДоговора = Дистрибьютор		0,5000	1,0000
ПрекращениеОтношений = Не прекращены	0,6667		
ПрекращениеОтношений = Несоблюдение договора	0,1667		
ПрекращениеОтношений = Прекращение контрагентом	0,1667	1,0000	1,0000

Расстояния между кластерами

Кластеры	1	2	3
1		8,7740	15,4524
2	8,7740		11,5370
3	15,4524	11,5370	

Рис. 323. Результат кластерного анализа

Отметим тот факт, что в результате анализа получают данные именно о найденных кластерах (их количество, центры, расстояния между ними). В результате анализа не получают данные о том, какие объекты (в нашем случае контрагенты) в какие кластеры входят. Подобное поведение наблюдается в случае, если настройка параметров проводимого анализа не проводится явным образом (а именно параметра `ТипЗаполненияТаблицы`).

Для того чтобы в результате анализа увидеть распределение объектов по кластерам, необходимо перед выполнением анализа (но после определения его типа) определить следующую строку кода:

[Копировать в буфер обмена](#)

`Анализ.Параметры.ТипЗаполненияТаблицы.Значение = ТипЗаполненияТаблицыРезультатаАнализаДанных.ИспользуемыеПоля;`

14.6.1. Используемые метрики

Сразу отметим такой факт: несмотря на то, что в предыдущем примере входные колонки имели непрерывный тип (для этого типа очевидно понятие «расстояния»), при анализе могут использоваться и колонки дискретных типов (ссылки на справочники, значения перечислений и т. п.).

Разберемся с метриками, которые могут использоваться при проведении кластерного анализа.

14.6.1.1. Евклидова метрика

В данной метрике расстояние между двумя объектами вычисляется по формуле:

$$\text{Расстояние} = \sqrt{\sum (W_i * (X_i - Y_i)^2)}$$

Где:

- X_i, Y_i – значения атрибутов двух объектов (между которыми определяется расстояние);
- W_i – весовой коэффициент атрибута (устанавливается в колонке анализа);
- i – номер атрибута, от 1 до n ;
- n – число атрибутов.

Предположим, что объекты характеризуются одним свойством, которое у одного объекта имеет значение 9, у другого – 5. Весовой коэффициент данного атрибута равен единице. Расстояние между объектами будет равно:

$$\sqrt{\sum (1_1 * (9_1 - 5_1)^2)} = \sqrt{\sum (1 * (4)^2)} = 4$$

14.6.1.2. Евклидова метрика в квадрате

В данной метрике расстояние между двумя объектами вычисляется по формуле:

$$\text{Расстояние} = \sum (W_i * (X_i - Y_i)^2)$$

Где:

- X_i, Y_i – значения атрибутов двух объектов (между которыми определяется расстояние);
- W_i – весовой коэффициент атрибута (устанавливается в колонке анализа);
- i – номер атрибута, от 1 до n ;
- n – число атрибутов.

Предположим, что объекты характеризуются одним свойством, которое у одного объекта имеет значение 5, у другого – 3. Весовой коэффициент данного атрибута равен двум. Расстояние между объектами будет равно:

$$\sum (2_1 * (5_1 - 3_1)^2) = \sum (2 * 1 * (2)^2) = 8$$

14.6.1.3. Метрика города

В данной метрике расстояние между двумя объектами вычисляется по формуле:

$$\text{Расстояние} = \sum (W_i * |X_i - Y_i|)$$

Где:

- X_i, Y_i – значения атрибутов двух объектов (между которыми определяется расстояние);
- W_i – весовой коэффициент атрибута (устанавливается в колонке анализа);
- i – номер атрибута, от 1 до n ;
- n – число атрибутов.

Предположим, что объекты характеризуются двумя атрибутами, которые имеют значения 3 и 5, 7 и 3. Вес первого равен 2, вес второго равен 1:

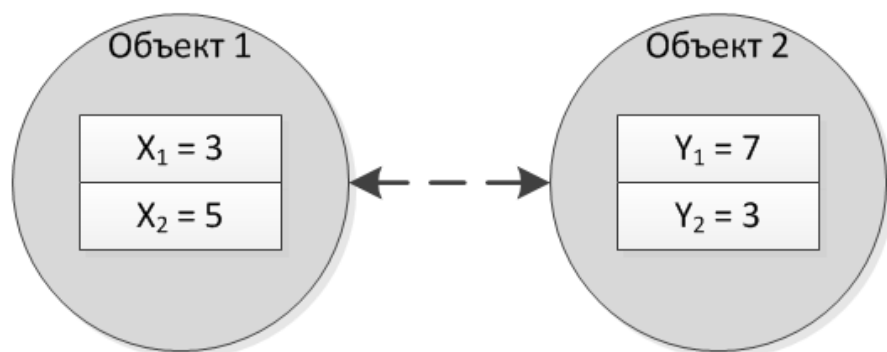


Рис. 324. Характеристики объектов

$$(W_1 * |X_1 - Y_1|) + (W_2 * |X_2 - Y_2|) = (2 * |3 - 7|) + (1 * |5 - 3|) = 10$$

14.6.1.4. Метрика доминирования

В данной метрике расстояние между двумя объектами вычисляется по формуле:

$$\text{Расстояние} = \text{MAX}(W_i * |X_i - Y_i|)$$

Где:

- X_i, Y_i – значения атрибутов двух объектов (между которыми определяется расстояние);
- W_i – весовой коэффициент атрибута (устанавливается в колонке анализа);
- i – номер атрибута, от 1 до n ;
- n – число атрибутов.

Предположим, что объекты характеризуются двумя атрибутами, которые имеют значения 3 и 5, 7 и 3. Вес первого равен 2, вес второго равен 1 (см. [рис. 324](#)).

$$\text{MAX}(W_1 * |X_1 - Y_1|, W_2 * |X_2 - Y_2|) = \text{MAX}(2 * |3 - 7|, 1 * |5 - 3|) = 8$$

14.6.2. Методы кластеризации

Вариант метода кластеризации определяет, исходя из каких принципов объект соотносится к той или иной группе, по какому алгоритму производится формирование кластеров.

Можно сказать, что целью любого алгоритма кластеризации является:

- минимизация изменчивости внутри кластеров,
- максимизация изменчивости между кластерами.

Различия между методами будем рассматривать на объектах, представленных на рисунке (см. [рис. 325](#)).

Будем считать, что объекты образуют две группы. Первая состоит из объектов 1, 2 и 3. Вторая группа состоит из объектов 4, 5 и 6.

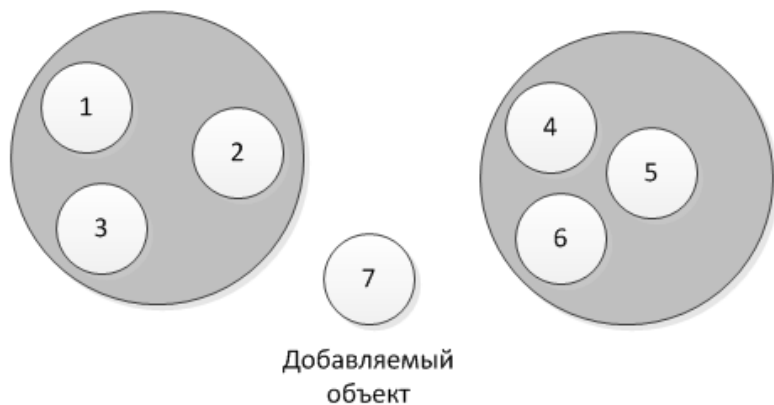


Рис. 325. Группы объектов

14.6.2.1. Ближняя связь

Метод кластеризации, в котором объект присоединяется к той группе, для которой расстояние до ближайшего объекта минимально.

В рассматриваемом примере объект 7 будет включен в группу, в которой находится объект 4. Самыми близкими объектами двух групп являются объекты 4 и 3. Расстояние до объекта 4 минимально.

14.6.2.2. Дальняя связь

Метод кластеризации, в котором объект присоединяется к той группе, для которой расстояние до наиболее дальнего объекта минимально.

В рассматриваемом примере объект 7 будет включен в группу, в которой находится объект 5. Самыми дальними объектами двух групп являются объекты 1 и 5. Расстояние до объекта 5 меньше.

14.6.2.3. Центр тяжести

Метод кластеризации, в котором объект присоединяется к той группе, для которой расстояние до центра тяжести минимально:

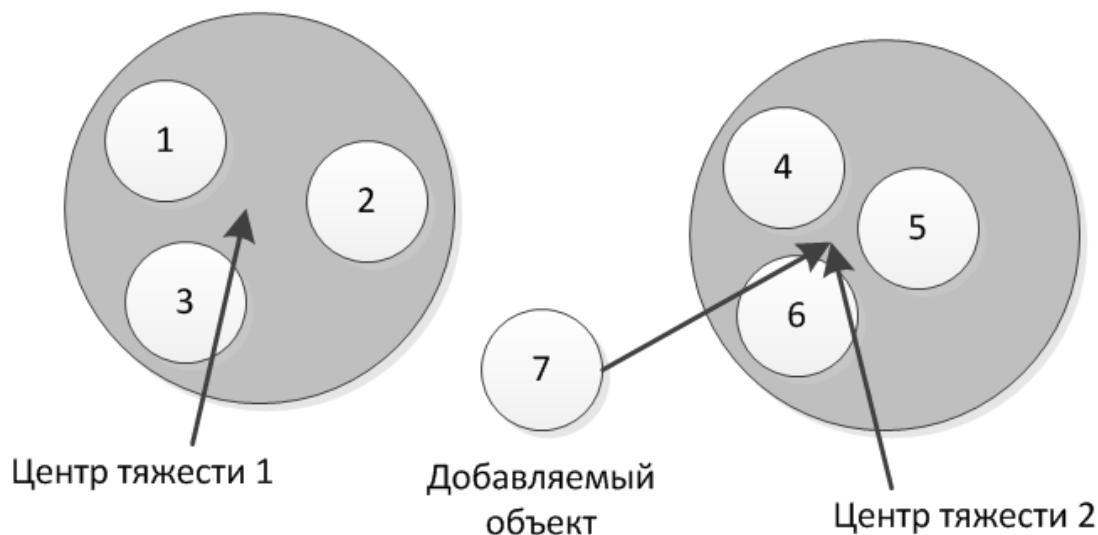


Рис. 326. Группы объектов

В примере, рассмотренном на рисунке, объект 7 добавится в группу, содержащую объекты 4, 5 и 6. Расстояние до центра тяжести (некоего мифического объекта со средними значениями атрибутов) минимально.

14.6.2.4. k-средних

В данном методе выбираются объекты, находящиеся первыми в выборке. Они считаются центрами кластеров. Далее выбирается следующий объект и, в соответствии с расстоянием до центров кластеров, относится к тому или иному кластеру. Центр кластера, к которому был добавлен объект, пересчитывается.

Процедура повторяется до полного перебора всех объектов. Далее опять производится новая выборка объектов (начиная с первого). Процедура повторяется до тех пор, пока изменяются центры кластеров:

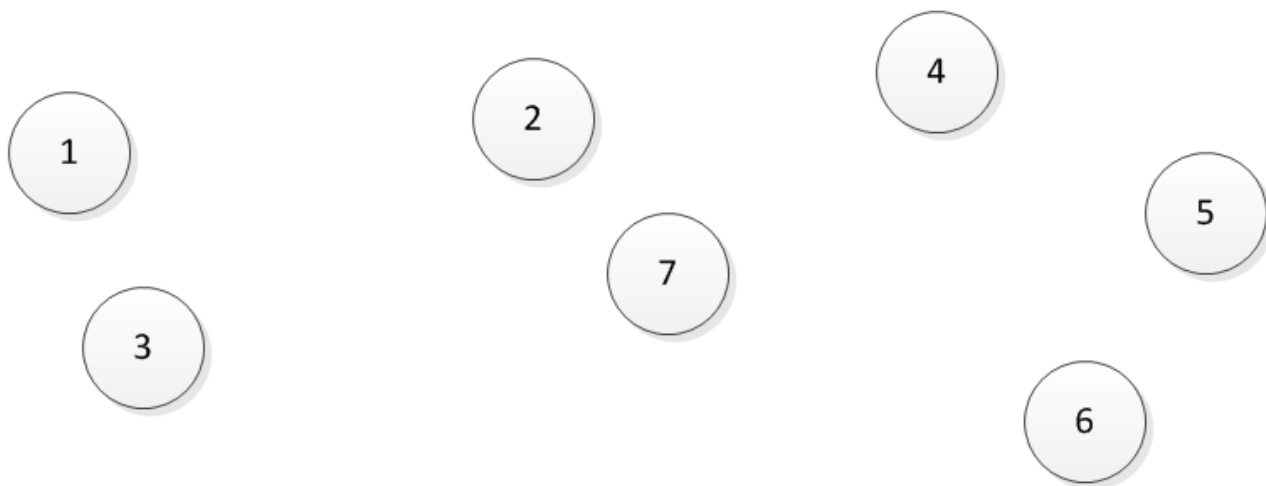


Рис. 327. Пример расположения объектов

Предположим, что произвольно выбраны в качестве центров кластеров объекты 1 и 2. Объект 3 добавляется к кластеру, центром которого является объект 1. Центр первого кластера перерасчитывается (он находится между объектом 1 и 3). Объект 4 добавляется ко второму кластеру (его центр также перерасчитывается).

После перебора всех анализируемых объектов к первому кластеру относятся объекты 1 и 3, ко второму – остальные объекты (его центр предположительно находится в центре треугольника из объектов 4, 7, 6).

Далее опять производится выборка объектов и распределение их по кластерам (относительно постоянно рассчитываемых центров кластеров).

Где-то на третьей выборке объектов, скорее всего, объект 2, который изначально был центром второго кластера, станет относиться к первому кластеру.

В конце алгоритма к первому кластеру будут относиться объекты 1, 2, 3. Ко второму – объекты 4, 5, 6, 7.

14.6.2.5. Вывод данных в дендрограмму

При выводе данных кластерного анализа, если используется алгоритм, отличный от алгоритма k-средних, результаты кластерного анализа выводятся в виде дендрограммы (алгоритм анализа должен предусматривать вывод распределения анализируемых объектов по кластерам):

Дендрограмма связей

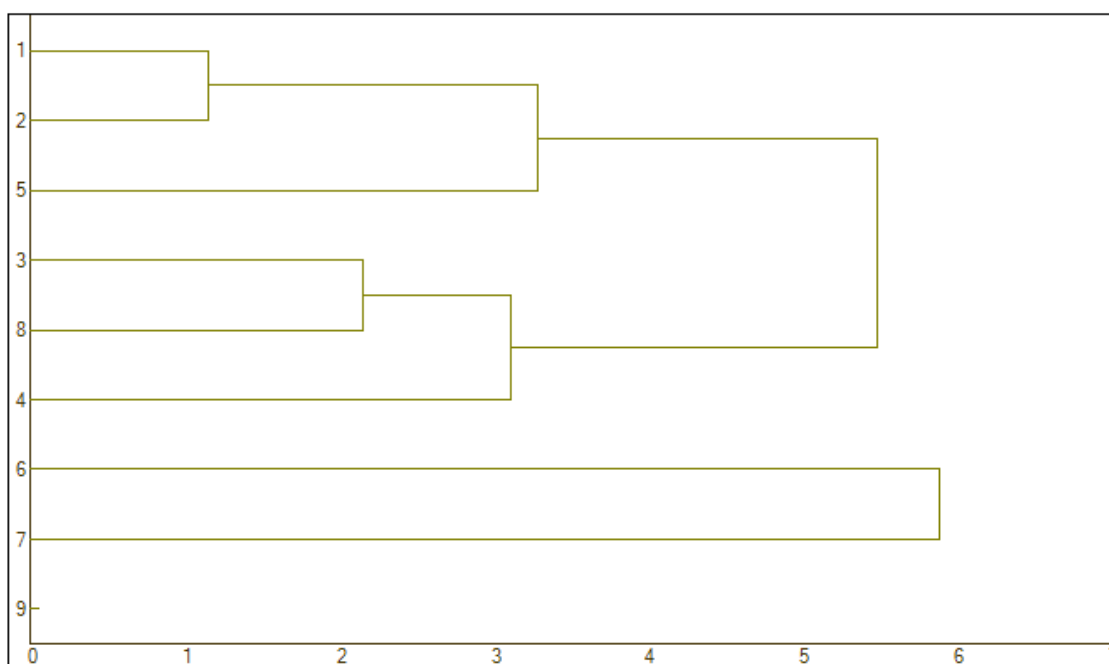


Рис. 328. Дендрограмма

Глава 15. Механизмы обмена данными

15.1. Цели и задачи

Механизмы обмена данными – это набор средств системы «1С:Предприятие», предназначенных для организации обмена данными между различными информационными базами, а также информационными базами и внешними программными системами. Механизмы обмена данными могут быть условно разделены на два уровня:

- универсальные механизмы обмена данными,
- распределенные информационные базы.

15.1.1. Универсальные механизмы обмена данными

Универсальные механизмы обмена данными могут использоваться как вместе, так и по отдельности, в различных комбинациях, для организации обмена данными информационных баз системы «1С:Предприятие» с различными программными системами. В качестве программных систем, с которыми организуется обмен, могут выступать другие информационные базы системы «1С:Предприятие». При этом обменивающиеся между собой информационные базы могут в общем случае иметь разные конфигурации.

Кроме того, универсальные механизмы обмена данными могут использоваться для организации обмена с программами, не основанными на системе «1С:Предприятие». Этому способствуют следующие факторы:

- формат обмена данными основан на языке XML, являющемся на сегодняшний день общепринятым средством представления данных;
- средства обмена данными, благодаря своей модульной организации и высокой гибкости, могут быть использованы для организации разнообразных схем обмена данными;
- протоколы, предлагаемые механизмами обмена данными, несложны и могут быть воспроизведены во внешних программных системах.

15.1.2. Распределенные информационные базы

Распределенная информационная база представляет собой иерархическую структуру, состоящую из отдельных информационных баз системы «1С:Предприятие» – узлов распределенной информационной базы, между которыми организован обмен данными с целью синхронизации конфигурации и данных.

Механизмы управления распределенными информационными базами базируются на универсальных механизмах обмена данными, но содержат некоторые дополнительные возможности, недоступные через универсальные механизмы.

Главное отличие распределенных информационных баз от универсальных механизмов обмена данными заключается в том, что универсальные механизмы обмена данными позволяют выстраивать достаточно произвольные схемы обмена данными, в то время как распределенные информационные базы имеют более узкую специализацию, а также выполняют передачу изменений конфигурации в подчиненные узлы.

15.2. Универсальные механизмы обмена данными

К универсальным механизмам обмена данными могут быть отнесены:

- средства чтения и записи документов XML,
- XML-сериализация,
- планы обмена.

15.2.1. Средства чтения и записи документов XML

Предполагается, что участники обмена данными обмениваются сообщениями в формате XML. Таким образом, средства чтения и записи документов XML образуют базовый уровень обмена данными.

Средства чтения и записи документов XML обеспечивают работу с документами XML в самом общем виде. Данный набор средств не определяет способов представления данных системы «1С:Предприятие» в формате XML.

К средствам чтения и записи документов XML, предоставляемым системой «1С:Предприятие», относятся объекты:

ЧтениеXML, ЗаписьXML и ПреобразованиеXSL. Также платформа предоставляет возможность работать с XML-данными в формате FastInfoset, для чего существуют объекты ЧтениеFastInfoset и ЗаписьFastInfoset.

15.2.2. XML-сериализация

Основная задача XML-сериализации – поддержка чтения/записи объектов данных системы «1С:Предприятие» в/из XML.

Базовые средства чтения и записи документов XML не предоставляют достаточной основы для решения данной задачи. Они не определяют форматов представления данных системы «1С:Предприятие» в XML и не предоставляют средств для чтения/записи объектов данных в/из XML в принятом формате как единого целого.

15.2.2.1. Представление данных в XML-сериализации

В конечном счете каждый объект данных системы «1С:Предприятие» представляется как элемент XML, содержащий значение объекта данных.

С точки зрения представления в XML типы значений делятся на простые и сложные.

К простым типам данных относятся типы, значения которых представляются подсистемой XML-сериализации в виде элементов XML только с текстовым содержимым.

Значения сложных типов представляются в виде элементов XML, содержащих вложенные элементы.

Каждому из типов данных системы «1С:Предприятие», значения которых могут быть представлены в XML, ставится в соответствие тип данных XML.

Каждый тип данных XML характеризуется именем типа и пространством имен, к которому относится тип.

Тип данных XML может быть следующим:

- одним из типов, определенных в документе XML Schema Part 2: Datatypes консорциума W3C (пространство имен – <http://www.w3.org/2001/XMLSchema>);
- предопределенным типом системы «1С:Предприятие» (пространство имен – <http://v8.1c.ru/data>);
- типом, производным от метаданных конфигурации системы «1С:Предприятие» (не относится ни к какому пространству имен).

В представлении объекта данных в XML тип данных XML может быть задан в явном виде. Для задания типа данных XML элемент XML, содержащий представление значения, должен содержать атрибут `type`, относящийся к пространству имен <http://www.w3.org/2001/XMLSchema-instance>, значение которого содержит тип данных XML.

Другим возможным способом задания типа данных XML является имя корневого элемента XML, содержащего представление значения. Имя корневого элемента, представляющего объект данных, жестко не специфицируется и может быть произвольным. Однако если при записи значения в XML имя корневого элемента не задано, то оно будет установлено в соответствии с типом записываемого значения. При чтении данных из XML тип значения, если он не задан в атрибуте `type`, может быть установлен по имени корневого элемента.

При рассмотрении примеров представления различных значений в XML и при дальнейшем изложении будем исходить из предположения, что определены следующие соответствия пространств имен:

[Копировать в буфер обмена](#)

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:v8="http://v8.1c.ru/data"
```

15.2.2.1.1. Представление значений простых типов в XML

К простым типам с точки зрения представления в XML относятся следующие типы системы «1С:Предприятие»:

- Число;
- Строка;
- Дата;
- Булево;

918

- **ДвоичныеДанные**;
- **NULL**;
- **УникальныйИдентификатор**;
- **ХранилищеЗначения**;
- все ссылки на объекты базы данных;
- ссылки на перечисления, определяемые в метаданных.

Число

Описание:

Типу **Число** соответствует тип данных XML **decimal** из пространства имен <http://www.w3.org/2001/XMLSchema>.

Правила представления значений данного типа определены в документе XML Schema Part 2: Datatypes.

Пример:

[Копировать в буфер обмена](#)

```
<!-- Не задано явно имя корневого элемента -->
<decimal>45684.087</decimal>
<!-- Явно задано имя корневого элемента XML -->
<Amount>523</Amount>
<!-- Явно указан тип данных XML -->
<Data xsi:type="xsd:decimal">64793.01</Data>
```

Строка

Описание:

Типу **Строка** соответствует тип данных **string** из пространства имен <http://www.w3.org/2001/XMLSchema>. **Строка** записывается в XML как есть.

Пример:

[Копировать в буфер обмена](#)

```
<!-- Не задано явно имя корневого элемента -->
<string>Это такая строка</string>
<!-- Явно задано имя корневого элемента XML -->
<Name>Иванов</Name>
<!-- Явно указан тип данных XML -->
<Data xsi:type="xsd:string">Это такая строка</Data>
```

Дата

Описание:

Значения типа **Дата** представляются в виде **YYYY-MM-DDTНН:ММ:SS**, где:

- **YYYY** – год, представленный в виде четырех цифр;
- **MM** – месяц, представленный двумя цифрами;
- **DD** – день месяца двумя цифрами;
- **T** – латинская буква **T**;
- **НН** – час суток;
- **ММ** – минута;
- **SS** – секунда.

Такой формат даты определен как допустимый в документе XML Schema Part 2: Datatypes.

Пример:

[Копировать в буфер обмена](#)

```
<!-- Не задано явно имя корневого элемента -->
<dateTime>2008-11-21T12:00:00</dateTime>
<!-- Явно задано имя корневого элемента XML -->
<Started>2001-10-30T19:00:00</Started>
<!-- Явно указан тип данных XML -->
<Data xsi:type="xsd:dateTime">1980-08-25T10:00:00</Data>
```

Булево

Описание:

Типу **Булево** соответствует тип данных `boolean` из пространства имен <http://www.w3.org/2001/XMLSchema>.

Значение **Ложь** представляется строкой `false`, а значение **Истина** – строкой `true`. Такой формат предусмотрен в документе XML Schema Part 2: Datatypes.

Пример:

[Копировать в буфер обмена](#)

```
<!-- Не задано явно имя корневого элемента -->
<boolean>>false</boolean>
<!-- Явно задано имя корневого элемента XML -->
<Posted>>true</Posted>
<!-- Явно указан тип данных XML -->
<Data xsi:type="xsd:boolean">>true</Data>
```

ДвоичныеДанные

Описание:

Типу **ДвоичныеДанные** соответствует тип данных XML `base64Binary` из пространства имен <http://www.w3.org/2001/XMLSchema>.

Значения данного типа представляются как двоичные данные, закодированные с использованием алгоритма `Base64`, описанного в [RFC 2045](http://tools.ietf.org/html/rfc2045) (<http://tools.ietf.org/html/rfc2045.html>).

Пример:

[Копировать в буфер обмена](#)

```
<!-- Не задано явно имя корневого элемента -->
<base64Binary>YWJjZGVm</base64Binary>
<!-- Явно задано имя корневого элемента XML -->
<BinaryData>YWJjZGVm</BinaryData>
<!-- Явно указан тип данных XML -->
<Data xsi:type="xsd:base64Binary">YWJjZGVm</Data>
```

NULL

Описание:

Типу **NULL** соответствует тип данных XML `Null` из пространства имен <http://v8.1c.ru/data>. Данный тип имеет единственное значение, которое представляется пустой строкой.

Пример:

[Копировать в буфер обмена](#)

```
<!-- Не задано явно имя корневого элемента -->
<v8:Null/>
<!-- Явно задано имя корневого элемента XML -->
<Selected/>
<!-- Явно указан тип данных XML -->
<Data xsi:type="v8:Null"/>
```

УникальныйИдентификатор

Описание:

Типу **УникальныйИдентификатор** соответствует тип данных XML **UUID** из пространства имен <http://v8.1c.ru/data>.

Значения данного типа представляются в XML в соответствии с общепринятой практикой и стандартами (**ISO-11578, DCE 1.1: Remote Procedure Call – Universal Unique Identifier**).

Пример:

[Копировать в буфер обмена](#)

```
<!-- Не задано явно имя корневого элемента -->
<v8:UUID>3294be0f-c039-41a9-bd65-596da0dcfe68</v8:UUID>
<!-- Явно задано имя корневого элемента XML -->
<Id>da035e32-3f7a-4d87-accf7db8cb4b</Id>
<!-- Явно указан тип данных XML -->
<Data xsi:type="v8:UUID">08839b0b-5ec3-4a53-a9f5-173312316919</Data>
```

ХранилищеЗначения

Описание:

Типу **ХранилищеЗначения** соответствует тип данных XML **ValueStorage** из пространства имен <http://v8.1c.ru/data>.

Значения данного типа представляются в XML как данные **ХранилищеЗначения**, сохраненные в файл, а затем закодированные с использованием алгоритма **Base64**.

Пример:

[Копировать в буфер обмена](#)

```
<!-- Не задано явно имя корневого элемента -->
<v8:ValueStorage>AQEOAAAAAAAAAAO+7v3siUyIsIjHQoSJ9</v8:ValueStorage>
<!-- Явно задано имя корневого элемента XML -->
<Data>AQEOAAAAAAAAAAO+7v3siUyIsIjHQoSJ9</Data>
<!-- Явно указан тип данных XML -->
<Data xsi:type="v8:ValueStorage">AQEOAAAAAAAAAAO+7v3siUyIsIjHQoSJ9</Data>
```

Ссылки на объекты базы данных

Описание:

Каждому из типов ссылок на объекты базы данных соответствует свой собственный тип данных XML. Имя типа данных XML для ссылок на объекты базы данных соответствует англоязычному имени типа значения ссылки системы «1С:Предприятие».

Так, например, для справочника **Валюты** англоязычное имя типа ссылки будет выглядеть как **CatalogRef.Валюты**. Так же будет выглядеть и имя типа данных XML.

Типы данных XML для ссылок на объекты базы данных не относятся ни к какому пространству имен.

Значения ссылок представляются в XML как значения типа **УникальныйИдентификатор**, полученные из ссылок.

Пример:

[Копировать в буфер обмена](#)

```
<!-- Не задано явно имя корневого элемента -->
<CatalogRef.Банки>911b5b8b-11f5-4993-9673-2c9a7a8995d5</CatalogRef.Банки >
<!-- Явно задано имя корневого элемента XML -->
<Ref>911b5b8b-11f5-4993-9673-2c9a7a8995d5</Ref>
<!-- Явно указан тип данных XML -->
<Data xsi:type="CatalogRef.Банки">
    911b5b8b-11f5-4993-9673-2c9a7a8995d5</Data>
```

Ссылки на перечисления, определяемые в метаданных

Описание:

Каждому из типов ссылок на значения перечислений, определенных в конфигурации, соответствует свой собственный тип данных XML. Имя типа данных XML для ссылок на значения перечисления соответствует англоязычному имени типа системы «1С:Предприятие».

Так, например, для перечисления **ВидыАдресов** англоязычное имя типа ссылки на значение будет выглядеть как **EnumRef.ВидыАдресов**. Так же будет выглядеть и имя типа данных XML.

Типы данных XML для ссылок на значения перечислений не относятся ни к какому пространству имен. В XML ссылки на значения перечислений представляются в виде имени соответствующего значения перечисления.

Пример:

[Копировать в буфер обмена](#)

```
<!-- Не задано явно имя корневого элемента -->
<EnumRef.ВидыАдресов>Юридический</EnumRef.ВидыАдресов>
<!-- Явно задано имя корневого элемента XML -->
<Ref>Юридический</Ref>
<!-- Явно указан тип данных XML -->
<Data xsi:type="EnumRef.ВидыАдресов">Физический</Data>
```

15.2.2.1.2. Представление значений сложных типов в XML

К сложным типам, значения которых могут быть представлены в XML, относятся следующие типы системы «1С:Предприятие»:

- **Тип**;
- **ОписаниеТипов**;
- **КонстантаМенеджерЗначения**. <Имя константы>;
- все объекты базы данных;
- наборы записей регистров, последовательностей, перерасчетов;
- **УдалениеОбъекта**.

Тип

Описание:

Типу **Тип** соответствует тип данных XML **Type** из пространства имен <http://v8.1c.ru/data>. Элемент XML, представляющий значение данного типа, содержит текст, в котором записано имя типа XML, соответствующего типу данных системы «1С:Предприятие».

На первый взгляд тип **Тип** относится не к сложным, а к простым типам данных, так как элемент, представляющий значение данного типа, не содержит вложенных элементов. Однако это не так. Вложенных элементов действительно нет. Но при этом текст элемента, содержащий имя типа данных XML, содержит префикс пространства имен типа, который должен быть определен в данном элементе или одном из родительских элементов, что делает текст элемента не вполне самостоятельным. Поэтому данный тип не отнесен к простым типам.

Пример:

[Копировать в буфер обмена](#)

```
<!-- Не задано явно имя корневого элемента -->
<v8:Type>v8:ValueStorage</v8:Type>
<!-- Явно задано имя корневого элемента XML -->
<Tp>xsd:string</Tp>
<!-- Явно указан тип данных XML -->
<Data xsi:type="v8:Type">v8:ValueStorage<Data>
```

ОписаниеТипов

Описание:

Типу **ОписаниеТипов** соответствует тип данных XML **TypeDescription** из пространства имен <http://v8.1c.ru/data>. Корневой элемент, представляющий значение типа **ОписаниеТипов**, включает в себя ряд вложенных элементов, каждый из которых содержит некоторую составляющую часть описания типов.

Вложенный элемент **Types** из пространства имен <http://v8.1c.ru/data> содержит представления отдельных типов, входящих в описание типов. Элемент с именем **NumberQualifiers** из пространства имен <http://v8.1c.ru/data> содержит квалификаторы числового значения. А элементы с именами **StringQualifiers** и **DateQualifiers** из того же пространства имен содержат квалификаторы строки и даты соответственно.

Пример:

```
<v8:TypeDescription>
  <v8:Types>
    <v8:Type>v8:UUID</v8:Type>
    <v8:Type>CatalogRef.Банки</v8:Type>
    <v8:Type>xsd:boolean</v8:Type>
    <v8:Type>xsd:decimal</v8:Type>
  </v8:Types>
  <v8:NumberQualifiers>
    <v8:Digits>10</v8:Digits>
    <v8:FractionDigits>2</v8:FractionDigits>
    <v8:AllowedSign>Any</v8:AllowedSign>
  </v8:NumberQualifiers>
  <v8:StringQualifiers>
    <v8:Length>30</v8:Length>
    <v8:AllowedLength>Variable</v8:AllowedLength>
  </v8:StringQualifiers>
  <v8>DateQualifiers>
    <v8>DateFractions>Date</v8>DateFractions>
  </v8>DateQualifiers>
</v8:TypeDescription>
```

КонстантаМенеджерЗначения. <Имя константы >

Описание:

Каждому из типов **КонстантаМенеджерЗначения. <Имя константы >** соответствует тип данных XML **ConstantValueManager. <Имя константы >**, не относящийся ни к какому пространству имен.

Пример:

```
<ConstantValueManager.НазваниеОрганизации>
<Value>000 "Мебиус" </Value>
</ConstantValueManager.НазваниеОрганизации>
```

[Копировать в буфер обмена](#)

Объекты базы данных

Описание:

Объекты базы данных представляются в XML как совокупность значений реквизитов и табличных частей. Имя типа данных XML, соответствующего объекту базы данных, определяется как англоязычное имя типа значения системы «1С:Предприятие». Типы данных XML для объектов базы данных не относятся ни к какому пространству имен. Состав элементов XML, вложенных в корневой элемент, определяется типом объекта, а также составом реквизитов и табличных частей.

Каждый из реквизитов представляется элементом XML, имя которого соответствует имени реквизита. Если тип значения реквизита не может быть однозначно определен из метаданных, то элемент XML, представляющий реквизит, содержит атрибут **xsi:type**, в котором указан тип значения XML.

Каждая из табличных частей представляется элементом XML, имя которого совпадает с именем табличной части.

Каждая из строк табличной части представляется элементом XML с именем **Row**. Реквизиты табличной части представлены элементами XML, вложенными в элемент **Row**.

Пример:

Представление в XML объекта типа **Документ.ЗаказПокупателя**:

```
<DocumentObject.ЗаказПокупателя>
<Ref>8d106783-9726-11d7-9334-0050ba8480bd</Ref>
<DeletionMark>false</DeletionMark>
<Date>2008-04-15T12:00:00</Date>
<Number>00000006</Number>
<Posted>true</Posted>
<ПодразделениеКомпании>
317f130d-5a08-11d7-9324-0050ba8480bd</ПодразделениеКомпании>
<СтруктурнаяЕдиница xsi:type="CatalogRef.КассыКомпании">
317f12f4-5a08-11d7-9324-0050ba8480bd</СтруктурнаяЕдиница>
<Контрагент>12952ac7-5a08-11d7-9324-0050ba8480bd</Контрагент>
<ЮрФизЛицоКонтрагента xsi:type="CatalogRef.ЮридическиеЛица">
```

[Копировать в буфер обмена](#)

```

0aadfe81-5a08-11d7-9324-0050ba8480bd</ЮрФизЛицоКонтрагента>
<ВалютаДокумента>
029156b4-5a08-11d7-9324-0050ba8480bd</ВалютаДокумента>
<КурсДокумента>1</КурсДокумента>
<УчитыватьНДС>true</УчитыватьНДС>
<УчитыватьНП>false</УчитыватьНП>
<СуммаВключаетНДС>false</СуммаВключаетНДС>
<СуммаВключаетНП>false</СуммаВключаетНП>
<Комментарий/>
<СуммаДокумента>44077.14</СуммаДокумента>
<ВидОперации>СчетНаОплату</ВидОперации>
<ДоговорВзаиморасчетов>
b0401f23-6e84-11d7-932c-0050ba8480bd</ДоговорВзаиморасчетов>
<СкладКомпании>
317f1317-5a08-11d7-9324-0050ba8480bd</СкладКомпании>
<ТипЦен>317f1311-5a08-11d7-9324-0050ba8480bd</ТипЦен>
<ДатаОплаты>2008-04-15T00:00:00</ДатаОплаты>
<АвтоРезервирование>false</АвтоРезервирование>
<АвтоРазмещение>false</АвтоРазмещение>
<КурсВзаиморасчетов>33.4209</КурсВзаиморасчетов>
<ТипСкидкиНаценки>
00000000-0000-0000-0000-000000000000</ТипСкидкиНаценки>
<Организация>317f1308-5a08-11d7-9324-0050ba8480bd</Организация>
<ДатаОтгрузки>2008-04-15T00:00:00</ДатаОтгрузки>
<Ответственный>
4ff40e0b-5ac5-11d7-9325-0050ba8480bd</Ответственный>
<КратностьДокумента>1</КратностьДокумента>
<КратностьВзаиморасчетов>1</КратностьВзаиморасчетов>
<Товары>
<Row>
<Номенклатура>
297c6534-5a08-11d7-9324-0050ba8480bd</Номенклатура>
<ЕдиницаИзмерения>
297c6535-5a08-11d7-9324-0050ba8480bd
</ЕдиницаИзмерения>
<Цена>4537.56</Цена>
<Сумма>13612.68</Сумма>
<СтавкаНДС>НДС20</СтавкаНДС>
<СуммаНДС>2722.54</СуммаНДС>
<СтавкаНП>1ac73736-5a08-11d7-9324-0050ba8480bd</СтавкаНП>
<СуммаНП>0</СуммаНП>
<ХарактеристикаНоменклатуры>
00000000-0000-0000-0000-000000000000
</ХарактеристикаНоменклатуры>
<Размещение xsi:nil="true"/>
<Коэффициент>1</Коэффициент>
<Количество>3</Количество>
<ПроцентСкидкиНаценки>0</ПроцентСкидкиНаценки>
</Row>
<Row>
<Номенклатура>
317f12d8-5a08-11d7-9324-0050ba8480bd</Номенклатура>
<ЕдиницаИзмерения>
317f12d9-5a08-11d7-9324-0050ba8480bd
</ЕдиницаИзмерения>
<Цена>4915.55</Цена>
<Сумма>19662.2</Сумма>
<СтавкаНДС>НДС20</СтавкаНДС>
<СуммаНДС>3932.44</СуммаНДС>
<СтавкаНП>1ac73736-5a08-11d7-9324-0050ba8480bd</СтавкаНП>
<СуммаНП>0</СуммаНП>
<ХарактеристикаНоменклатуры>
00000000-0000-0000-0000-000000000000
</ХарактеристикаНоменклатуры>
<Размещение xsi:nil="true"/>
<Коэффициент>1</Коэффициент>
<Количество>4</Количество>
<ПроцентСкидкиНаценки>0</ПроцентСкидкиНаценки>
</Row>
</Товары>
<ВозвратнаяТара/>
</DocumentObject.ЗаказПокупателя>

```

ПРИМЕЧАНИЕ. В используемом примере, а также в других примерах данной главы присутствуют длинные строки (например, `<ЕдиницаИзмерения>317f12d9-5a08-11d7-9324-0050ba8480bd</ЕдиницаИзмерения>`). В связи с ограничением, накладываемым форматом книги, такие строки приводятся с переносом на следующую

Набор записей

Описание:

Представление в XML набора записей включает отбор, по которому получен набор записей, и сами записи, входящие в отбор. Значения отбора представлены во вложенном элементе XML с именем **Filter**, не относящимся ни к какому пространству имен. А все записи, составляющие набор записей, представлены во вложенном элементе с именем **Records**, также не относящимся ни к какому пространству имен. Записи представлены элементами XML с именем **Record**, вложенными в элемент **Records**. Имя элемента **Record** также не относится ни к какому пространству имен.

Пример:

Представление в XML набора записей регистра накопления **ОстаткиТоваровКомпании**:

[Копировать в буфер обмена](#)

```
<AccumulationRegisterRecordSet.ОстаткиТоваровКомпании>
<Filter>
<Recorder xsi:type="DocumentRef.РеализацияТоваров">
1725f36e-6f35-11d7-932d-0050ba8480bd</Recorder>
</Filter>
<Records>
<Record>
<Recorder xsi:type="DocumentRef.РеализацияТоваров">
1725f36e-6f35-11d7-932d-0050ba8480bd</Recorder>
<Period>2008-04-13T17:45:39</Period>
<MovementType>Expense</MovementType>
<Active>true</Active>
<Номенклатура>
297c6556-5a08-11d7-9324-0050ba8480bd</Номенклатура>
<СкладКомпании>
317f1317-5a08-11d7-9324-0050ba8480bd</СкладКомпании>
<Заказ xsi:nil="true"/>
<ЦенаВРознице>0</ЦенаВРознице>
<ХарактеристикаНоменклатуры>
00000000-0000-0000-0000-000000000000
</ХарактеристикаНоменклатуры>
<Количество>2</Количество>
<ПодразделениеКомпании>
317f130d-5a08-11d7-9324-0050ba8480bd
</ПодразделениеКомпании>
</Record>
<Record>
<Recorder xsi:type="DocumentRef.РеализацияТоваров">
1725f36e-6f35-11d7-932d-0050ba8480bd</Recorder>
<Period>2008-04-13T17:45:39</Period>
<MovementType>Expense</MovementType>
<Active>true</Active>
<Номенклатура>
297c6558-5a08-11d7-9324-0050ba8480bd</Номенклатура>
<СкладКомпании>
317f1317-5a08-11d7-9324-0050ba8480bd</СкладКомпании>
<Заказ xsi:nil="true"/>
<ЦенаВРознице>0</ЦенаВРознице>
<ХарактеристикаНоменклатуры>
ac47d77e-5ec7-11d7-9329-0050ba8480bd
</ХарактеристикаНоменклатуры>
<Количество>2</Количество>
<ПодразделениеКомпании>
317f130d-5a08-11d7-9324-0050ba8480bd
</ПодразделениеКомпании>
</Record>
</Records>
</AccumulationRegisterRecordSet.ОстаткиТоваровКомпании>
```

УдалениеОбъекта

Описание:

Типу **УдалениеОбъекта** соответствует тип данных XML **ObjectDeletion** из пространства имен <http://v8.1c.ru/data>. Корневой элемент XML-представления значения типа **УдалениеОбъекта** содержит один вложенный элемент с

именем `Ref` из пространства имен <http://v8.1c.ru/data>, в котором находится представление ссылки на объект базы данных.

Пример:

Представление в XML объекта типа `УдалениеОбъекта`:

[Копировать в буфер обмена](#)

```
<v8:ObjectDeletion xmlns="http://v8.1c.ru/data">
<v8:Ref xsi:type="CatalogRef.Банки">
60c5cec3-7f6f-4ec3-9620-e757fe3614ca</v8:Ref>
</v8:ObjectDeletion>
```

15.2.2.2. Доступ к средствам XML-сериализации из встроенного языка

Для работы с XML-представлениями значений простых типов предназначены два метода глобального контекста – `XMLСтрока()` и `XMLЗначение()`.

Метод `XMLСтрока()` имеет единственный параметр – значение, для которого нужно получить XML-представление. Это значение должно относиться к типу, являющемуся простым с точки зрения XML-сериализации. В противном случае будет вызвано исключение. При нормальном завершении функция возвращает строку, которая может быть использована как текст элемента XML, представляющего значение простого типа.

Метод `XMLЗначение()` выполняет противоположную задачу. У этого метода два параметра:

- тип значения, которое нужно получить из строки;
- сама строка.

Для преобразования типа данных системы «1С:Предприятие» в тип данных XML и наоборот предназначены методы `XMLТип()` и `ИзXMLТипа()`. Метод `XMLТип()` имеет один параметр – тип, для которого нужно получить соответствующий тип данных XML. Если соответствующий тип данных XML определен, то метод возвращает значение типа `ТипДанныхXML`. Если же соответствующего типа данных XML нет, то метод возвращает значение `Неопределено`.

Метод `ИзXMLТипа()` имеет два варианта вызова. В первом варианте метод имеет единственный параметр типа `ТипДанныхXML`. Во втором варианте параметра два: имя типа XML и пространство имен. В обоих случаях метод возвращает соответствующий типу данных XML тип данных системы «1С:Предприятие», если таковой имеется, или `Неопределено` в противном случае.

Для записи и чтения различных значений в/из XML предназначены методы глобального контекста `ЗаписатьXML()` и `ПрочитатьXML()`.

Метод `ЗаписатьXML()` имеет два обязательных параметра. Первый параметр – это объект типа `ЗаписьXML`, через который осуществляется запись XML; а второй – значение, которое должно быть записано в XML. Необязательные параметры образуют три различных варианта вызова метода.

В простейшем случае параметра три, и в качестве третьего параметра указывается значение перечисления `НазначениеТипаXML`, определяющее необходимость явного указания типа данных XML в атрибуте `xsi:type` корневого элемента XML.

У следующего варианта вызова в качестве третьего параметра используется строковое значение, указывается имя корневого элемента XML. При этом подразумевается, что пространство имен не определено. Четвертый параметр – значение типа `НазначениеТипаXML`, определяющее необходимость явного указания типа данных XML.

И, наконец, у последнего варианта вызова после параметра, указывающего имя корневого элемента XML, появляется еще один параметр – строковое значение, обозначающее пространство имен, к которому относится корневой элемент. Последний параметр по-прежнему имеет тип `НазначениеТипаXML`.

[Копировать в буфер обмена](#)

```
...
Знач = "Строка такая";
ЗаписатьXML(Зп, Знач);
ЗаписатьXML(Зп, Знач, "Root", НазначениеТипаXML.Явное);
ЗаписатьXML(Зп, Знач, "Root", "urn:some-namespace");
...
```

В результате выполнения приведенного выше фрагмента будет получен следующий XML-фрагмент.

```

...
<string>Строка такая</string>
<Root xsi:type="xsd:string">Строка такая</Root>
<dlp1:Root xmlns:dlp1="urn:some-namespace">Строка такая</dlp1:Root>
...

```

Если в качестве значения, помещаемого в XML, будет передано значение типа, который не может быть представлен в XML, то будет вызвано исключение.

Метод `ПрочитатьXML()` предназначен для чтения значений из XML. Данный метод имеет один обязательный параметр – объект `ЧтениеXML`, из которого должно быть прочитано значение. В качестве второго параметра может быть указан тип значения, которое должно быть прочитано из XML. Если тип значения явно указан в XML, то в качестве второго параметра может быть указано значение `Неопределенно`, или же он может быть вообще опущен. В этом случае метод `ПрочитатьXML()` пытается определить тип читаемого значения по содержимому атрибута `xsi:type`, а если атрибут `xsi:type` отсутствует, то по имени элемента. Если не удалось установить тип или значение указанного типа не может быть прочитано из XML, то вызывается исключение. При удачном завершении метод `ПрочитатьXML()` возвращает считанное значение.

Следует обратить внимание на то, как считываются менеджеры значений констант, объекты базы данных и наборы записей. После успешного выполнения чтения метод `ПрочитатьXML()` возвращает считанное из XML значение, но это значение еще не записано в базу данных. Если, например, считан элемент справочника, то для того, чтобы считанный элемент справочника оказался записанным в базу данных, необходимо обратиться к его методу `Записать()`, как и при «обычной» записи измененного состояния объекта. Это же относится и к другим объектам базы данных, менеджерам записи констант и наборам записей.

При чтении объекта базы данных из XML в базе данных производится поиск объекта с таким же значением ссылки. Если такой объект найден, то считывание из XML выглядит так, как будто объект был прочитан из базы данных, после чего значения его реквизитов, табличных частей и т. п. перезаписываются полученными из XML значениями. Если же объект по ссылке не найден, то считывание из XML выглядит как создание нового объекта, установка ему значения ссылки и заполнение его содержимого значениями, прочитанными из XML.

Метод `ВозможностьЧтенияXML()` определяет, возможно ли считывание значения из объекта `ЧтениеXML`, находящегося в текущей позиции документа XML. Объект `ЧтениеXML` передается данному методу в качестве параметра. Если метод возвращает `Истина`, то чтение возможно; если `Ложь` – значение не может быть считано.

Метод `ПолучитьXMLТип()` позволяет получить из объекта `ЧтениеXML` тип данных XML, соответствующий текущей позиции документа XML. Данный метод также имеет один параметр – `ЧтениеXML`.

15.2.3. Планы обмена

Планы обмена являются центром, вокруг которого группируются прочие механизмы, связанные с обменом данными. В одной конфигурации может быть определено произвольное количество планов обмена. Каждый из планов обмена определяет набор данных, которыми предполагается обмениваться в рамках данного плана обмена. Вместе с набором данных могут определяться и специфические форматы представления этих данных.

Предполагается, что форматы данных основаны на XML, но благодаря гибкости языка XML и наличию развитых средств работы с XML в системе «1С:Предприятие» остается достаточно большое пространство для творчества в области способов представления данных.

В планах обмена можно выделить две значимые составляющие:

- инфраструктура сообщений,
- служба регистрации изменений.

Элементами данных плана обмена являются узлы плана обмена, подобно тому, как элементами данных справочника являются элементы справочника. Каждый из узлов плана обмена обозначает участника обмена данными по данному плану обмена. Один из узлов соответствует данной информационной базе, а остальные – другим участникам, с которыми данная информационная база может обмениваться данными.

Данные переносятся между узлами с помощью сообщений. Средства работы с сообщениями образуют **инфраструктуру сообщений**. Каждое сообщение относится к определенному плану обмена, имеет определенный **узел-отправитель** и определенный **узел-получатель**. Сообщение не может быть отправлено неизвестному узлу и не может быть принято от неизвестного узла. Каждое сообщение имеет свой собственный целочисленный номер.

Служба регистрации изменений предназначена для регистрации изменений данных, производимых системой «1С:Предприятие», чтобы при обмене данными иметь возможность передавать не все данные, а только

измененные.

Таким образом, планы обмена определяют набор механизмов, предназначенных для организации обмена данными. Рассмотрим эти механизмы подробнее.

15.2.3.1. Узлы планов обмена

При создании нового плана обмена в нем автоматически создается один узел – этот узел или узел плана обмена, соответствующий данной информационной базе. Остальные узлы, то есть узлы, с которыми данный узел может обмениваться данными, в рамках плана обмена автоматически не создаются.

Для каждого узла должен быть определен уникальный код, так как при обмене данными узел идентифицируется по коду. Коды узлов задаются таким образом, чтобы обменивающиеся стороны «узнали» друг друга.

ПРИМЕЧАНИЕ. План обмена не может быть без кода и наименования. Другими словами, для стандартных реквизитов плана обмена **Код** и **Наименование** нельзя установить длину, равной 0.

Предположим, требуется организовать обмен данными между двумя информационными базами по плану обмена с именем **УдаленныеСклады**. Одна из этих информационных баз выполняет функции центрального офиса, а другая – удаленного склада.

В этом случае было бы целесообразно в качестве значения кода этого узла плана обмена **УдаленныеСклады** в первой информационной базе задать значение **Офис** (если, конечно, позволяет длина кода), а во второй – **Склад1**. Таким образом, эти узлы будут поименованы. Но этого недостаточно, ведь нужно еще задать узлы, с которыми будет производиться обмен данными. Для этого в первой информационной базе в плане обмена **УдаленныеСклады** следует создать узел с кодом **Склад1**, а во второй информационной базе – узел с кодом **Офис**.

Таким образом, первая информационная база будет «знать», что в рамках плана обмена **УдаленныеСклады** ее саму «зовут» **Офис** и она будет вести обмен с узлом по имени **Склад1**; а вторая – что ее «зовут» **Склад1**, а обмениваться данными она будет с узлом **Офис**.

15.2.3.2. Инфраструктура сообщений

Важнейшей составляющей инфраструктуры сообщений являются сами сообщения. Как уже отмечалось, сообщения передаются в рамках плана обмена от одного узла другому. То есть каждое сообщение точно ассоциировано с планом обмена, имеет одного отправителя и одного получателя.

Рассмотрим, что такое сообщение. Сообщение оформляется как документ XML, имеющий определенную структуру. В качестве примера приведем следующее сообщение:

[Копировать в буфер обмена](#)

```
<v8msg:Message xmlns:v8msg="http://v8.1c.ru/messages">
  <v8msg:Header>
    <v8msg:ExchangePlan>УдаленныеСклады</v8msg:ExchangePlan>
    <v8msg:To>Склад1</v8msg:To>
    <v8msg:From>Офис</v8msg:From>
    <v8msg:MessageNo>20</v8msg:MessageNo>
    <v8msg:ReceivedNo>15</v8msg:ReceivedNo>
  </v8msg:Header>
  <v8msg:Body>
    <!-- Тело сообщения -->
  </v8msg:Body>
</v8msg:Message>
```

Все сообщение находится внутри элемента XML с именем **Message**, относящимся к пространству имен <http://v8.1c.ru/messages>. Сообщение делится на заголовок и тело сообщения. Соответственно, элемент **Message** содержит два вложенных элемента с именами **Header** и **Body**. Оба относятся к пространству имен <http://v8.1c.ru/messages>.

Элемент **Header** содержит заголовок сообщения. Структура заголовка жестко задана. Информация заголовка представлена в нескольких элементах XML, вложенных в элемент **Header**. Все элементы, вложенные в элемент **Header**, относятся к пространству имен <http://v8.1c.ru/messages>:

- **ExchangePlan** содержит имя плана обмена, к которому относится сообщение.
- **To** содержит код узла-отправителя.
- **From** содержит код узла, для которого предназначено сообщение.

- **MessageNo** содержит номер данного сообщения. Номер сообщения является положительным целым числом и присваивается узлом-отправителем. Номер каждого последующего сообщения равен номеру предыдущего отправленного сообщения плюс 1.

- **ReceivedNo** содержит максимальный номер сообщения, которое узел-отправитель данного сообщения принял от узла-получателя данного сообщения. Данное значение включено в состав заголовка сообщения для подтверждения приема сообщений.

Тело сообщения содержится в элементе XML с именем **Body**, относящимся к пространству имен <http://v8.1c.ru/messages>. Данный элемент может иметь произвольное содержимое, определяемое прикладными потребностями. Инфраструктурой сообщений содержимое тела сообщения никак не регламентируется.

15.2.3.3. Служба регистрации изменений

Суть регистрации изменений состоит в том, чтобы иметь перечень измененных элементов данных. Эти элементы данных должны быть переданы в очередном сообщении тому или иному узлу, с которым производится обмен данными. При каждом изменении данных должно быть зарегистрировано, что имеются изменения и их предстоит передать во все узлы, с которыми поддерживается обмен этими данными. При получении подтверждения приема сообщения, в котором были отправлены изменения, записи регистрации изменений должны быть удалены.

Регистрация изменений может выполняться для следующих элементов данных:

- **КонстантаМенеджерЗначения.<Имя константы>**;
- **Объекты базы данных:**
 - **СправочникОбъект.<Имя справочника>**;
 - **ДокументОбъект.<Имя документа>**;
 - **ПланСчетовОбъект.<Имя плана счетов>**;
 - **ПланВидовХарактеристикОбъект.<Имя плана видов характеристик>**;
 - **ПланВидовРасчетаОбъект.<Имя плана видов расчета>**;
 - **БизнесПроцессОбъект.<Имя бизнес-процесса>**;
 - **ЗадачаОбъект.<Имя задачи>**.
- **Наборы записей:**
 - **РегистрСведенийНаборЗаписей.<Имя регистра сведений>**;
 - **РегистрБухгалтерииНаборЗаписей.<Имя регистра бухгалтерии>**;
 - **РегистрНакопленияНаборЗаписей.<Имя регистра накопления>**;
 - **ПоследовательностьНаборЗаписей.<Имя последовательности>**;
 - **РегистрРасчетаНаборЗаписей.<Имя регистра расчета>**.
 - **ПерерасчетНаборЗаписей.<Имя перерасчета>**.

Для каждого из приведенных элементов данных ведется своя таблица регистрации изменений. Таблицы имеют разную структуру, в зависимости от того, для каких элементов данных регистрируются изменения, но все-таки структуры таблиц подобны. В структуре можно выделить три составляющих:

- ключ элемента данных, для которого регистрируются изменения;
- ссылка на узел, в который изменение должно быть передано;
- номер сообщения, в котором изменение передано в первый раз.

Структуры таблиц регистрации изменений для разных данных отличаются ключом, так как ключи у разных данных разные:

- для константы ключом является идентификатор константы;
- для объектов базы данных в качестве ключа используется ссылка на объект;
- для наборов записей, для которых определен регистратор, в качестве ключа используется ссылка на объект-регистратор;
- для набора записей регистра сведений, если регистратор не определен, в качестве ключа используется совокупность измерений, входящих в основной отбор; а если регистр сведений является периодическим и включен основной отбор по периоду, то в ключ входит еще и период.

Изменение элемента данных должно быть зарегистрировано для всех узлов, в которые изменение должно быть передано. Таким образом, в результате изменения элемента данных в таблице регистрации изменений должно появиться N записей, где N – количество узлов, для которых регистрируются изменения. В каждой из этих записей указано одно и то же значение ключа элемента данных и различные значения ссылки на узел.

Непосредственно после выполнения регистрации изменения номер сообщения имеет значение **NULL**. При первой отправке изменения в данное поле помещается номер сообщения, в котором изменение отправлено.

При конфигурировании плана обмена определяется так называемый состав плана обмена. В состав плана обмена включаются объекты метаданных. Вхождение объекта метаданных в состав плана обмена показывает, что изменения данных, соответствующих объекту метаданных, могут регистрироваться для узлов данного плана обмена. Если объект метаданных не входит в состав ни одного плана обмена, то для данного объекта не создается таблица регистрации изменений и регистрация изменений данных не выполняется.

При определении вхождения объекта метаданных в состав плана обмена указывается свойство **Авторегистрация**. Авторегистрацию можно разрешить или запретить. Если авторегистрация разрешена, то при изменении данных регистрация будет выполнена автоматически. Если запрещена, то регистрацию изменения можно выполнить вручную.

15.2.3.4. Доступ к механизмам планов обмена средствами встроенного языка

Для работы с планами обмена из встроенного языка предусмотрен ряд объектов.

Объект **ПланыОбменаМенеджер** помимо традиционной для такого рода менеджеров функциональности содержит методы для работы со службой регистрации изменений **ЗарегистрироватьИзменения()**, **УдалитьРегистрациюИзменений()**, **ИзменениеЗарегистрировано()**, **ВыбратьИзменения()**, а также методы для создания объектов, читающих и записывающих сообщения: **СоздатьЧтениеСообщения()**, **СоздатьЗаписьСообщения()**.

Для объекта **ПланОбменаМенеджер. <Имя плана обмена>** наиболее важной особенностью является наличие метода **ЭтотУзел()**, возвращающего ссылку на узел данного плана обмена, соответствующий данной информационной базе (далее используется термин «этот узел»). Для определения узла плана обмена, соответствующего данной информационной базе, также служит стандартный реквизит **ЭтотУзел**. Значение этого реквизита (**ЭтотУзел**) можно изменять. Основным вариантом применения этой операции – восстановление корректного состояния информационной базы после того, как произошло восстановление данных из файла XML. В этом случае необходимо назначить в восстановленной базе в качестве «этого узла» тот элемент плана обмена, который был установлен в базе-источнике. Это особенно актуально в том случае, когда ссылки на «этот узел» плана обмена были запомнены в каких-либо объектах информационной базы. Рассмотрим процесс замены «этого узла» более подробно.

Обычной записью элемента плана обмена с установленным (или сброшенным) свойством **ЭтотУзел** выполнить данную операцию невозможно (при наличии в системе элемента плана обмена с установленным свойством **ЭтотУзел**) – система выдаст исключение. Для того чтобы выполнить эту операцию, необходимо выполнить следующие действия:

1. Для элемента, который планируется установить новым элементом «этот узел», следует установить свойство **ЭтотУзел** в режиме **ОбменДанными.Загрузка = Истина** (см. [здесь](#)).
2. Для элемента, который ранее был элементом «этот узел», следует сбросить свойство **ЭтотУзел** (обычным образом).
3. Рекомендуется эти действия (установку нового и отключение старого узла) выполнять в транзакции.

Пример смены элемента «этот узел»:

```

ТекущийЭтотУзел = ПланыОбмена.ОбменДанными.ЭтотУзел();
НовыйОбъект = НовыйЭтотУзел.ПолучитьОбъект();
НовыйОбъект.ОбменДанными.Загрузка = Истина;
НовыйОбъект.ЭтотУзел = Истина;
НовыйОбъект.Записать();
ТекущийОбъект = ТекущийЭтотУзел.ПолучитьОбъект();
ТекущийОбъект.ЭтотУзел = Ложь;
ТекущийОбъект.Записать();
ЗафиксироватьТранзакцию();

```

При смене `ЭтотУзел` следует иметь ввиду несколько особенностей поведения системы:

1. Если завершить работу системы в ситуации, когда у плана обмена отсутствует элемент «этот узел», то при старте система автоматически создаст новый элемент плана обмена с установленным свойством `ЭтотУзел`.
2. Если в системе создана два элемента со значением свойства `ЭтотУзел` равным значением `Истина`, то поведение системы в этом случае становится неопределенным.

`ПланОбменаОбъект.<Имя плана обмена>` соответствует узлу плана обмена. Особого внимания заслуживают свойства `НомерОтправленного` и `НомерПринятого`. Свойство `НомерОтправленного` содержит номер последнего сообщения, отправленного из данной информационной базы в адрес узла, которому соответствует объект `ПланОбменаОбъект.<Имя плана обмена>`. Свойство `НомерПринятого` содержит максимальный из номеров сообщений, принятых данной информационной базой от узла, которому соответствует объект `ПланОбменаОбъект.<Имя плана обмена>`.

15.2.3.4.1. Регистрация изменений

Как уже отмечалось, регистрация изменений может выполняться автоматически при записи или удалении элемента данных. Рассмотрим, как это происходит. У каждого из объектов, чьи изменения могут быть зарегистрированы планом обмена, имеется свойство `ОбменДанными` с типом `ПараметрыОбменаДанными`. Данное свойство может быть использовано только для чтения и предназначено для управления различными параметрами при обмене данными.

У объекта `ПараметрыОбменаДанными` есть свойство `Получатели`, имеющее тип `НаборУзлов`. В данном свойстве хранится перечень узлов, для которых будет выполняться регистрация изменений при записи или удалении данных. Список получателей заполняется автоматически перед тем, как будет вызван обработчик `ПередЗаписью()` при выполнении записи данных или `ПередУдалением()` при выполнении удаления. Однако автоматическое заполнение будет выполнено только в том случае, если свойство `Автозаполнение` объекта `НаборУзлов` имеет значение `Истина` (`Истина` является значением по умолчанию для свойства `Автозаполнение`). При автоматическом заполнении в список получателей попадают ссылки на все узлы всех планов обмена, в состав которых входит соответствующий объект метаданных, при условии, что значением свойства `Авторегистрация` является `Разрешить`. Само собой разумеется, что узлы, соответствующие данной информационной базе, при этом в список получателей не попадут. При выполнении автоматического заполнения список получателей предварительно очищается.

В обработчике `ПередЗаписью()` (и/или `ПередУдалением()`) в список получателей можно внести изменения: добавить или удалить ссылки на узлы. Однако следует помнить, что список получателей может содержать только ссылки на узлы, относящиеся к планам обмена, в состав которых входит соответствующий объект метаданных.

В приведенном ниже примере обработчик `ПередЗаписью()` исключает из списка получателей узел с кодом `Особый` плана обмена `УдаленныеСклады`.

[Копировать в буфер обмена](#)

```

Процедура ПередЗаписью()
    Узел = ПланыОбмена.УдаленныеСклады.НайтиПоКоду("Особый");
    ОбменДанными.Получатели.Удалить(Узел);
КонецПроцедуры

```

Присвоив свойству `Автозаполнение` значение `Ложь`, можно добиться того, что автоматическое заполнение списка получателей выполняться не будет. В этом случае действия со списком получателей можно производить не только в обработчике `ПередЗаписью()`, но и в любом фрагменте кода, как показано в примере.

[Копировать в буфер обмена](#)

```

Объект = Ссылка.ПолучитьОбъект();
Узел = ПланыОбмена.УдаленныеСклады.НайтиПоКоду("Склад1");
Объект.ОбменДанными.Получатели.Автозаполнение = Ложь;
Объект.ОбменДанными.Получатели.Добавить(Узел);
Объект.Записать();

```

Ряд методов для регистрации изменений содержит объект `ПланыОбменаМенеджер`. Прежде всего, это метод `ЗарегистрироватьИзменения()`. Данный метод позволяет выполнять регистрацию изменений одиночных элементов

данных или целых групп для одного или нескольких узлов. Первый параметр данного метода – ссылка на узел плана обмена или массив ссылок на узлы, для которых выполняется регистрация изменений. Если первый параметр представляет собой одиночную ссылку на узел, то второй параметр может быть опущен. При этом выполняется регистрация изменений всех элементов данных, которые на данный момент присутствуют в базе данных и изменения которых могут быть зарегистрированы для данного узла.

Это может быть полезно для организации начальной передачи данных вновь созданному узлу.

[Копировать в буфер обмена](#)

```
Узел = ПланыОбмена.УдаленныеСклады.НайтиПоКоду( "Новый" );
ПланыОбмена.ЗарегистрироватьИзменения(Узел);
```

Если же первый параметр представляет собой массив ссылок на узлы, то второй параметр обязательно должен быть указан. Впрочем, второй параметр может присутствовать и в том случае, если первый параметр – одиночная ссылка на узел. В зависимости от способа задания второго параметра можно зарегистрировать изменения одного элемента данных или же всех данных, относящихся к одному объекту метаданных.

Для регистрации изменений одного элемента в качестве второго параметра может быть указан сам элемент данных, ссылка на объект базы данных или объект типа [УдалениеОбъекта](#).

Если указан элемент данных, то регистрируется его изменение. Если указана ссылка на объект базы данных, то регистрируется изменение этого объекта.

Если второй параметр имеет тип [УдалениеОбъекта](#), то регистрируется изменение объекта базы данных, ссылку на который содержит [УдалениеОбъекта](#).

[Копировать в буфер обмена](#)

```
Узлы = Новый Массив(2);
Узлы[0] = ПланыОбмена.УдаленныеСклады.НайтиПоКоду( "Склад1" );
Узлы[1] = ПланыОбмена.УдаленныеСклады.НайтиПоКоду( "Склад2" );
Данные = Справочники.Номенклатура.НайтиПоКоду( "ТП00127" );
ПланыОбмена.ЗарегистрироватьИзменения(Узлы, Данные);
```

Для регистрации изменений всех данных, относящихся к объекту метаданных, в качестве второго параметра должен быть указан соответствующий объект метаданных.

[Копировать в буфер обмена](#)

```
Узлы = Новый Массив(2);
Узлы[0] = ПланыОбмена.УдаленныеСклады.НайтиПоКоду( "Склад1" );
Узлы[1] = ПланыОбмена.УдаленныеСклады.НайтиПоКоду( "Склад2" );
ПланыОбмена.ЗарегистрироватьИзменения(Узлы, Метаданные.Справочники.Номенклатура);
```

Для удаления записей регистрации изменений у объекта [ПланыОбменаМенеджер](#) имеется метод [УдалитьРегистрациюИзменений\(\)](#). С его помощью можно выполнить удаление записей регистрации изменений для всех данных, которые зарегистрированы для узла.

[Копировать в буфер обмена](#)

```
Узел = ПланыОбмена.УдаленныеСклады.НайтиПоКоду( "Склад1" );
ПланыОбмена.УдалитьРегистрациюИзменений(Узел);
```

Можно удалить записи регистрации изменений конкретного элемента данных для одного или нескольких узлов.

[Копировать в буфер обмена](#)

```
Узлы = Новый Массив(2);
Узлы[0] = ПланыОбмена.УдаленныеСклады.НайтиПоКоду( "Склад1" );
Узлы[1] = ПланыОбмена.УдаленныеСклады.НайтиПоКоду( "Склад2" );
Данные = Справочники.Номенклатура.НайтиПоКоду( "ТП00127" );
ПланыОбмена.УдалитьРегистрациюИзменений(Узлы, Данные);
```

Также можно удалить записи регистрации изменений всех данных, относящихся к объекту метаданных для одного или нескольких узлов.

[Копировать в буфер обмена](#)

```
Узлы = Новый Массив(2);
Узлы[0] = ПланыОбмена.УдаленныеСклады.НайтиПоКоду( "Склад1" );
Узлы[1] = ПланыОбмена.УдаленныеСклады.НайтиПоКоду( "Склад2" );
ПланыОбмена.УдалитьРегистрациюИзменений(Узлы, Метаданные.Справочники.Номенклатура);
```

Кроме того, если в качестве первого параметра указан одиночный узел, то в качестве второго параметра может быть указан номер сообщения. В этом случае метод [УдалитьРегистрациюИзменений\(\)](#) удаляет из всех таблиц регистрации изменений все записи, относящиеся к указанному узлу, у которых номер сообщения меньше или

равен значению второго параметра (но не `NULL`). Данная форма метода предназначена для удаления записей регистрации изменений, по которым получено подтверждение приема изменений от узла, указанного в первом параметре.

Для проверки, зарегистрировано ли изменение элемента данных для того или иного узла, служит метод `ИзменениеЗарегистрировано()`. Первый параметр данного метода – ссылка на узел, а второй параметр – элемент `Данные`, ссылка на объект базы данных или `УдалениеОбъекта`.

15.2.3.4.2. Запись сообщений обмена данными

Для записи сообщений предназначен объект `ЗаписьСообщенияОбмена`.

Объект `ЗаписьСообщенияОбмена` создается при обращении к методу `СоздатьЗаписьСообщения()` объекта `ПланыОбменаМенеджер`.

Объект `ЗаписьСообщенияОбмена` имеет три метода:

- `НачатьЗапись()`,
- `ЗакончитьЗапись()`,
- `ПрерватьЗапись()`.

У метода `НачатьЗапись()` два параметра: объект типа `ЗаписьXML`, через который будет записываться сообщение, и ссылка на узел, которому адресовано сообщение. Метод `НачатьЗапись()` вычисляет номер сообщения путем прибавления 1 к номеру предыдущего сообщения; производит запись начала элемента XML, содержащего все сообщение, заголовка сообщения целиком, а также начала элемента XML, содержащего тело сообщения. После этого можно приступить к записи содержимого тела сообщения.

Для нормального завершения записи сообщения предназначен метод `ЗакончитьЗапись()`. При обращении к этому методу производится запись конца элемента XML, содержащего тело сообщения, и конца элемента XML, содержащего все сообщение. После успешной записи элементов XML, завершающих сообщение, сообщение считается отправленным, и его номер запоминается как номер последнего отправленного сообщения от данного узла узлу-получателю.

Если есть необходимость прервать запись сообщения и не считать его отправленным, то следует обратиться к методу `ПрерватьЗапись()`.

Ниже приведен типовой фрагмент кода, выполняющий запись сообщения обмена данными.

[Копировать в буфер обмена](#)

```
ЗаписьXML = Новый ЗаписьXML();
ЗаписьXML.ОткрытьФайл(ИмяФайлаСообщения);
Узел = ПланыОбмена.УдаленныеСклады.НайтиПоКоду(КодУзла);
ЗаписьСообщения = ПланыОбмена.СоздатьЗаписьСообщения();
ЗаписьСообщения.НачатьЗапись(ЗаписьXML, Узел);
// Запись тела сообщения
ЗаписьСообщения.ЗакончитьЗапись();
```

Тело сообщения может содержать любые данные, представленные в XML, в зависимости от того, какой формат принят для того или иного плана обмена. Здесь же будет рассмотрен случай, когда сообщение содержит данные, изменения которых зарегистрированы службой регистрации изменений. Для реализации этого следует выбрать зарегистрированные изменения, обойти их и поместить XML-представления измененных данных в сообщение.

Для выборки изменений предназначен метод `ВыбратьИзменения()` объекта `ПланыОбменаМенеджер`. Данный метод имеет два параметра: ссылка на узел, для которого выбираются изменения, и номер сообщения, в которое изменения должны быть помещены. Метод `ВыбратьИзменения()` возвращает объект типа `ВыборкаДанных`, содержащий ключи данных, изменения которых были зарегистрированы для узла, переданного в качестве первого параметра. Кроме того, метод `ВыбратьИзменения()` помещает номер сообщения, переданный в качестве второго параметра, в соответствующие поля отобранных записей таблиц регистрации изменений, если в этих полях содержалось значение `NULL`.

Если изменение уже было выбрано при предыдущем обращении, то номер сообщения в соответствующей записи таблицы регистрации изменений не модифицируется; а если изменение еще не выбиралось, то при выборке запоминается номер первого сообщения, для которого изменения были выбраны.

Для обхода выбранных изменений объект `ВыборкаДанных` имеет методы `Следующий()` и `Получить()`. При обращении к методу `Следующий()` происходит переход к следующему ключу в выборке. При первом обращении к

методу `Следующий()` происходит переход к первому ключу. При обращении к методу `Получить()` происходит выборка из базы данных элемента данных, соответствующего текущему ключу выборки. Здесь следует сделать отдельное замечание относительно удаленных объектов базы данных. Если регистрация изменения объекта базы данных была выполнена в результате его удаления, то метод `Получить()` вернет не объект базы данных, а объект типа `УдалениеОбъекта`.

Помещение XML-представления элементов данных в сообщение выполняется при помощи метода `ЗаписатьXML()` глобального контекста.

Ниже приведен фрагмент кода, в котором выполняется формирование сообщения и в тело которого помещаются зарегистрированные изменения данных.

[Копировать в буфер обмена](#)

```
ЗаписьXML = Новый ЗаписьXML();
ЗаписьXML.ОткрытьФайл(ИмяФайлаСообщения);
Узел = ПланыОбмена.УдаленныеСклады.НайтиПоКоду(КодУзла);
ЗаписьСообщения = ПланыОбмена.СоздатьЗаписьСообщения();
ЗаписьСообщения.НачатьЗапись(ЗаписьXML, Узел);
Выборка = ПланыОбмена.ВыбратьИзменения(Узел, ЗаписьСообщения.НомерСообщения);
Пока Выборка.Следующий() Цикл
    Данные = Выборка.Получить();
    ЗаписатьXML(ЗаписьXML, Данные);
КонецЦикла;
ЗаписьСообщения.ЗакончитьЗапись();
```

15.2.3.4.3. Чтение сообщений обмена данными

Для чтения сообщений обмена данными предназначен объект `ЧтениеСообщенияОбмена`.

Объект создается при обращении к методу `СоздатьЧтениеСообщения()` объекта `ПланыОбменаМенеджер`.

Объект `ЧтениеСообщенияОбмена` имеет три метода:

- `НачатьЧтение()`,
- `ЗакончитьЧтение()`,
- `ПрерватьЧтение()`.

У метода `НачатьЧтение()` два параметра: объект типа `ЧтениеXML`, через который производится чтение сообщения, и значение перечисления `ДопустимыйНомерСообщения`.

Метод `НачатьЧтение()` считывает начало элемента XML, содержащего все сообщение, читает заголовок сообщения и проверяет его приемлемость:

- определен ли план обмена, к которому относится сообщение;
- правильно ли заданы отправитель и получатель сообщения;
- допустимый ли номер имеет сообщение.

Допустимость номера сообщения определяется с учетом значения второго параметра. Если второй параметр имеет значение `ДопустимыйНомерСообщения.Любой`, то читаемое сообщение может иметь любой номер. Если значение второго параметра `ДопустимыйНомерСообщения.Следующий`, то номер сообщения должен быть в точности на 1 больше максимального из номеров ранее принятых сообщений. А если второй параметр принимает значение `ДопустимыйНомерСообщения.Большой`, то номер сообщения просто должен быть больше максимального из номеров ранее принятых сообщений. Значением по умолчанию для второго параметра является `ДопустимыйНомерСообщения.Большой`.

После этого считывается начало элемента XML, содержащего тело сообщения. Если сообщение не является допустимым или при чтении произошла ошибка, то вызывается исключение. Если же все нормально, то можно приступить к чтению тела сообщения.

Для нормального завершения чтения сообщения используется метод `ЗакончитьЧтение()`. Данный метод считывает конец элемента XML, содержащего тело элемента, и конец элемента XML, содержащего все сообщение. Если все хорошо, то сообщение считается принятым, и номер сообщения, если он больше максимального из номеров ранее принятых сообщений, запоминается как максимальный номер принятого сообщения.

Метод `ПрерватьЧтение()` позволяет прервать чтение сообщения в любой момент.

Ниже приведен типовой фрагмент кода, в котором показано использование объекта `ЧтениеСообщенияОбмена`.

[Копировать в буфер обмена](#)

```
ЧтениеXML = Новый ЧтениеXML();
ЧтениеXML.ОткрытьФайл(ИмяФайлаСообщения);
ЧтениеСообщения = ПланыОбмена.СоздатьЧтениеСообщения();
ЧтениеСообщения.НачатьЧтение(ЧтениеXML, ДопустимыйНомерСообщения.Большой);
// Чтение тела сообщения
ЧтениеСообщения.ЗакончитьЧтение();
```

Как уже отмечалось, тело сообщения может в принципе содержать любую информацию, но здесь будет рассмотрен случай считывания из тела сообщения измененных данных. Фрагмент кода, в котором производится чтение сообщения, содержащего измененные данные, выглядит следующим образом.

[Копировать в буфер обмена](#)

```
ЧтениеXML = Новый ЧтениеXML();
ЧтениеXML.ОткрытьФайл(ИмяФайлаСообщения);
ЧтениеСообщения = ПланыОбмена.СоздатьЧтениеСообщения();
ЧтениеСообщения.НачатьЧтение(ЧтениеXML);
ПланыОбмена.УдалитьРегистрациюИзменений(ЧтениеСообщения.Отправитель, ЧтениеСообщения.НомерСообщения);
Пока ВозможностьЧтенияXML(ЧтениеXML) Цикл
    Данные = ПрочитатьXML(ЧтениеXML);
    Данные.ОбменДанными.Отправитель = ЧтениеСообщения.Отправитель;
    Данные.ОбменДанными.Загрузка = Истина;
    Данные.Записать();
КонецЦикла;
ЧтениеСообщения.ЗакончитьЧтение();
```

Обращение к методу `УдалитьРегистрациюИзменений()` объекта `МенеджерПлановОбмена` предназначено для удаления записей регистрации, по которым произведено подтверждение приема. Далее из тела сообщения с помощью метода `ПрочитатьXML()` считываются данные, пока имеется возможность считывания. Перед записью данных в базу данных производится изменение двух свойств объекта `ПараметрыОбменаДанными`, который принадлежит объекту, представляющему считанные из XML данные. В свойстве `Отправитель` указывается узел-отправитель данных, чтобы не производить регистрацию изменений для отправки в узел, откуда эти данные только что были получены. Установка свойству `Загрузка` значения `Истина` означает, что запись производится в рамках загрузки данных, а не обычной записи. В этом случае при записи не будут производиться некоторые проверки.

15.2.3.4.4. Гарантированная доставка сообщений

В рассмотренных примерах чтения и записи сообщений обмена данными предполагалось, что отправленные сообщения могут быть потеряны по тем или иным причинам. Поэтому при записи сообщений в них помещались все зарегистрированные изменения, в том числе и те, которые уже были отправлены, но по ним еще не получено подтверждение приема.

При приеме допустимым считается сообщение с номером большим, чем максимальный из номеров принятых сообщений. Удаление регистрации изменений производится только для номеров сообщений, прием которых подтвержден.

[Копировать в буфер обмена](#)

```
ЧтениеСообщения.НачатьЧтение(ЧтениеXML);
ПланыОбмена.УдалитьРегистрациюИзменений(ЧтениеСообщения.Отправитель, ЧтениеСообщения.НомерПринятого);
```

Такая логика считается типовой при реализации обмена данными. Однако можно реализовать и вариант, при котором доставка сообщений считается гарантированной. В этом случае удаление регистрации изменений должно производиться непосредственно после удачного завершения записи сообщения, чтобы отправка уже отправленных изменений не повторялась. А приниматься может только то сообщение, номер которого на 1 больше максимального из номеров ранее принятых сообщений.

В этом случае ранее приведенный фрагмент кода, в котором выполняется запись сообщения, будет выглядеть следующим образом.

[Копировать в буфер обмена](#)

```
ЗаписьXML = Новый ЗаписьXML();
ЗаписьXML.ОткрытьФайл(ИмяФайлаСообщения);
Узел = ПланыОбмена.УдаленныеСклады.НайтиПоКоду(КодУзла);
ЗаписьСообщения = ПланыОбмена.СоздатьЗаписьСообщения();
ЗаписьСообщения.НачатьЗапись(ЗаписьXML, Узел);
Выборка = ПланыОбмена.ВыбратьИзменения(Узел, ЗаписьСообщения.НомерСообщения);
Пока Выборка.Следующий() Цикл
```

```

Данные = Выборка.Получить();
ЗаписатьXML(ЗаписьXML, Данные);
КонецЦикла;
НомерСообщения = ЗаписьСообщения.НомерСообщения;
ЗаписьСообщения.ЗакончитьЗапись();
ПланыОбмена.УдалитьРегистрациюИзменений(Узел, НомерСообщения);

```

Фрагмент кода, в котором происходит прием сообщения, будет выглядеть следующим образом:

[Копировать в буфер обмена](#)

```

ЧтениеXML = Новый ЧтениеXML();
ЧтениеXML.ОткрытьФайл(ИмяФайлаСообщения);
ЧтениеСообщения = ПланыОбмена.СоздатьЧтениеСообщения();
ЧтениеСообщения.НачатьЧтение(ЧтениеXML, ДопустимыйНомерСообщения.Следующий);
Пока ВозможностьЧтенияXML(ЧтениеXML) Цикл
    Данные = ПрочитатьXML(ЧтениеXML);
    Данные.ОбменДанными.Отправитель = ЧтениеСообщения.Отправитель;
    Данные.ОбменДанными.Загрузка = Истина;
    Данные.Записать();
КонецЦикла;
ЧтениеСообщения.ЗакончитьЧтение();

```

Следует учитывать, что организация гарантированной доставки может быть достаточно сложной, и в большинстве случаев предпочтительно повторно посылать изменения до получения подтверждения их приема.

15.2.3.4.5. Разрешение коллизий

В приведенных выше примерах чтения и записи сообщений не учитывалось, что при обмене данными один и тот же элемент данных может быть изменен одновременно в двух обменивающихся данными узлах. В этом случае непонятно, какое из изменений должно быть в конечном счете принято. Такая ситуация называется **коллизией**.

Одним из способов разрешения коллизий может быть определение, какой из узлов является главным, а какой – подчиненным. При этом должно быть принято изменение, сделанное в главном узле; а изменение, сделанное в подчиненном узле, должно быть отвергнуто.

Для реализации этого при приеме сообщения перед записью данных необходимо установить, зарегистрировано ли изменение этих данных, и, в зависимости от роли узла в данной паре получатель-отправитель, принять решение: записывать или не записывать данные.

Ниже приведен пример реализации стратегии «главный – подчиненный» при чтении сообщения. Предполагается, что для хранения роли узла в плане обмена был определен реквизит **Главный**, имеющий тип **Булево**.

[Копировать в буфер обмена](#)

```

ЧтениеXML = Новый ЧтениеXML();
ЧтениеXML.ОткрытьФайл(ИмяФайлаСообщения);
ЧтениеСообщения = ПланыОбмена.СоздатьЧтениеСообщения();
ЧтениеСообщения.НачатьЧтение(ЧтениеXML);
ПланыОбмена.УдалитьРегистрациюИзменений(ЧтениеСообщения.Отправитель, ЧтениеСообщения.НомерСообщения);
Отправитель = ЧтениеСообщения.Отправитель;
Главный = Отправитель.Главный;
Пока ВозможностьЧтенияXML(ЧтениеXML) Цикл
    Данные = ПрочитатьXML(ЧтениеXML);
    Если Главный Или Не ПланыОбмена.ИзменениеЗарегистрировано(Отправитель, Данные) Тогда
        Данные.ОбменДанными.Отправитель = ЧтениеСообщения.Отправитель;
        Данные.ОбменДанными.Загрузка = Истина;
        Данные.Записать();
    КонецЕсли;
КонецЦикла;
ЧтениеСообщения.ЗакончитьЧтение();

```

15.2.3.4.6. Задание соответствий пространств имен

При записи сообщения, после выполнения метода **НачатьЗапись()** объекта **ЗаписьСообщенияОбмена**, не определено никаких соответствий пространств имен. В то же время ряд пространств имен может многократно использоваться при записи отдельных элементов данных. В этом случае определения соответствий одних и тех же пространств имен будут многократно встречаться в теле сообщения. Поэтому целесообразно после начала записи сообщения, но до начала записи тела сообщения поместить несколько строк кода, в которых будут определяться соответствия для наиболее часто используемых пространств имен.

Например, как это сделано в приведенном ниже фрагменте записи сообщения.

[Копировать в буфер обмена](#)

```

ЗаписьXML = Новый ЗаписьXML();
ЗаписьXML.ОткрытьФайл(ИмяФайлаСообщения);
Узел = ПланыОбмена.УдаленныеСклады.НайтиПоКоду(КодУзла);
ЗаписьСообщения = ПланыОбмена.СоздатьЗаписьСообщения();
ЗаписьСообщения.НачатьЗапись(ЗаписьXML, Узел);
ЗаписьXML.ЗаписатьСоответствиеПространстваИмен("xsd", "http://www.w3.org/2001/XMLSchema");
ЗаписьXML.ЗаписатьСоответствиеПространстваИмен("xsi", "http://www.w3.org/2001/XMLSchema-instance");
ЗаписьXML.ЗаписатьСоответствиеПространстваИмен("v8", "http://v8.1c.ru/data");
Выборка = ПланыОбмена.ВыбратьИзменения(Узел, ЗаписьСообщения.НомерСообщения);
Пока Выборка.Следующий() Цикл
    Данные = Выборка.Получить();
    ЗаписатьXML(ЗаписьXML, Данные);
КонецЦикла;
НомерСообщения = ЗаписьСообщения.НомерСообщения;
ЗаписьСообщения.ЗакончитьЗапись();
ПланыОбмена.УдалитьРегистрациюИзменений(Узел, НомерСообщения);

```

В ряде случаев такой прием поможет ощутимо уменьшить размер документа XML, в котором находится сообщение обмена данными.

15.3. Распределенные информационные базы

15.3.1. Общие принципы

Распределенная информационная база – это совокупность информационных баз системы «1С:Предприятие» (узлов распределенной информационной базы), в которых поддерживается синхронизация конфигурации и данных. Распределенная информационная база имеет иерархическую структуру. У каждого узла распределенной информационной базы может быть один **главный** и произвольное число **подчиненных** узлов. **Самый главный узел**, или узел, у которого нет главного узла, называется **корневым** узлом распределенной информационной базы (база **Главная база данных** на [рис. 329](#)). Каждый из узлов может обмениваться данными только со своими «соседями», то есть со своими главными и подчиненными узлами.

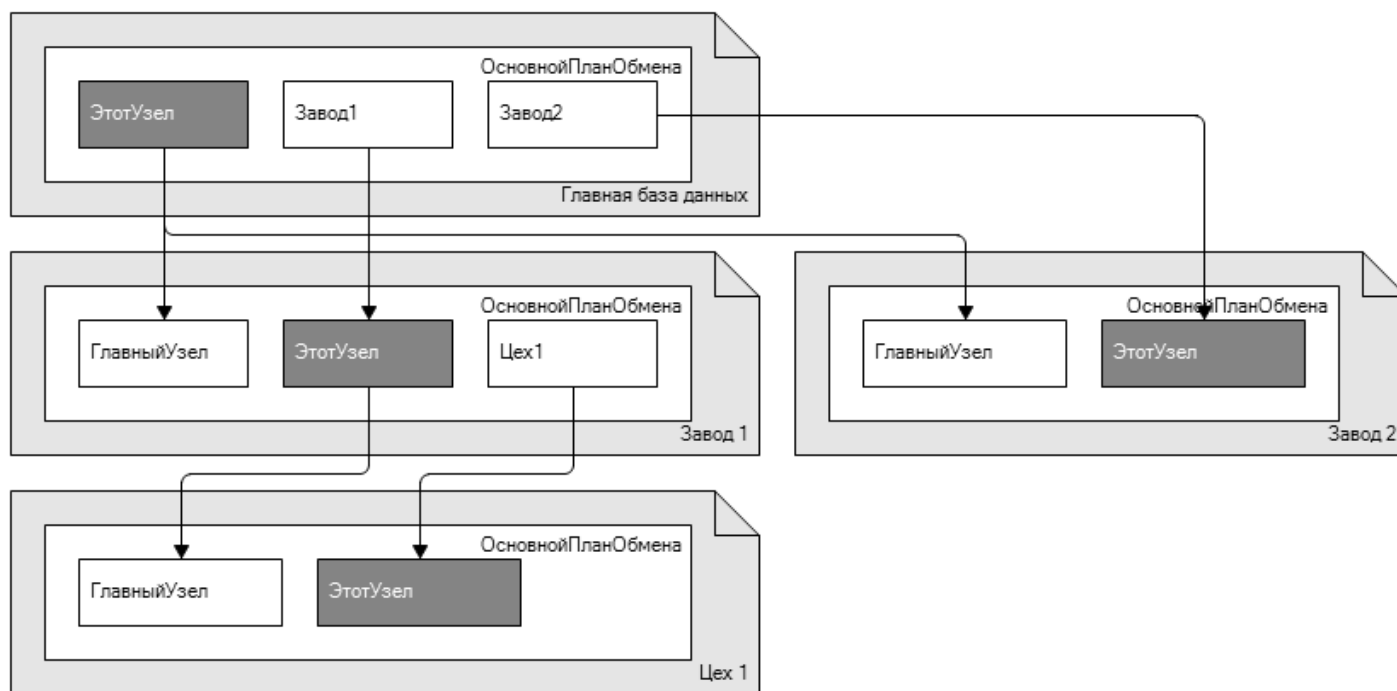


Рис. 329. Иерархическая распределенная информационная база

Изменения конфигурации допускаются только в корневом узле распределенной информационной базы с последующим ее распространением по иерархии от корневого узла к его подчиненным и т. д. Таким образом, механизм управления распределенными информационными базами обеспечивает наличие во всех узлах распределенной информационной базы одной и той же конфигурации.

Изменение данных допускается в любом узле распределенной информационной базы. Синхронизация данных достигается путем распространения изменений данных, произведенных в одном узле, во все структуры распределенной информационной базы.

При организации работы последовательности документов в распределенной информационной базе нужно

учитывать, что участие документа в последовательности имеет смысл только в одном узле распределенной информационной базы. Это может быть либо узел, в котором документ был создан, либо другой узел, но узел должен быть один. Нарушение данного принципа может привести к различным проблемам в процессе работы с системой, например, невозможности восстановления последовательности документов.

Если в рамках всей распределенной информационной базы поддерживается полная идентичность конфигурации, то полная идентичность данных необязательна. Состав данных, изменения которых передаются в рамках распределенной информационной базы, может регулироваться как «по вертикали» (путем определения множества объектов метаданных, данные которых участвуют в обмене), так и «по горизонтали» (путем задания условий на передачу и прием изменений на уровне отдельных элементов данных).

15.3.2. Планы обмена

Планы обмена занимают центральное место и в управлении распределенными информационными базами. Но для того, чтобы тот или иной план обмена оказался пригоден для организации распределенной информационной базы, у него при конфигурировании должно быть установлено свойство [Распределенная информационная база](#).

Данные в распределенной информационной базе переносятся с помощью сообщений, предоставляемых инфраструктурой сообщений. В отличие от универсальных механизмов обмена данными, содержимое сообщений, передаваемых между узлами распределенной информационной базы, не может быть произвольным, а является регламентированным протоколом обмена, принятым для распределенной информационной базы.

Номенклатура данных, изменениями которых будет производиться обмен в рамках распределенной информационной базы, определяется составом плана обмена. Вхождение объекта метаданных в состав плана обмена показывает, что изменения данных, соответствующих объекту метаданных, могут регистрироваться для узлов данного плана обмена. Но в отличие от универсальных механизмов обмена данными, номенклатура данных, обмен которыми может производиться в рамках распределенной информационной базы, строго ограничена составом соответствующего плана обмена.

Для регистрации изменений данных в распределенной информационной базе задействована служба регистрации изменений. Элементы данных помещаются в сообщение с использованием механизмов XML-сериализации. Помимо изменений данных, между узлами распределенной информационной базы передаются изменения конфигурации, а также некоторая дополнительная служебная информация. Регистрация изменений конфигурации и их передача в распределенной информационной базе осуществляются полностью автоматически и недоступны для пользователя и разработчика конфигураций.

В отличие от универсальных механизмов обмена данными, формирование и прием сообщения обмена данными в распределенной информационной базе производятся «в одно действие», то есть все содержимое сообщения формируется путем вызова одного метода встроенного языка. Аналогично и считывание содержимого сообщения производится путем вызова одного метода. Для того чтобы управлять составом данных, помещаемых в сообщение, а также считываемых из сообщения и помещаемых в базу данных, на уровне отдельных элементов данных в модуле плана обмена могут быть определены обработчики событий:

- [ПриОтправкеДанныхПодчиненному](#),
- [ПриОтправкеДанныхГлавному](#),
- [ПриПолученииДанныхОтПодчиненного](#),
- [ПриПолученииДанныхОтГлавного](#).

Таким образом, в распределенной информационной базе практически полностью задействованы универсальные механизмы обмена данными, но имеются и некоторые дополнительные возможности, недоступные вне распределенной информационной базы.

15.3.2.1. Главный и подчиненный узлы

Как было сказано выше, у каждого из узлов распределенной информационной базы может быть один главный и произвольное число подчиненных узлов (см. [рис. 329](#)). Для своего главного узла узел является подчиненным и, соответственно, для своих подчиненных – главным. Узел, у которого нет главного узла, является корневым узлом распределенной информационной базы.

ВНИМАНИЕ! Корневой узел распределенной информационной базы – это единственное место, где разрешено вносить изменения в конфигурацию информационной базы.

Распределенная информационная база может быть построена на основе нескольких планов обмена, с установленным свойством [Распределенная информационная база](#). Взаимодействие в каждой паре узлов «главный – подчиненный» производится в соответствии с одним из определенных в конфигурации планов обмена. Никаких

ограничений на использование того или иного плана обмена в том или ином узле распределенной информационной базы не накладывается.

Каждый из узлов распределенной информационной базы, как и в случае использования универсальных механизмов обмена данными, «знает» только своих «соседей», то есть свой главный и свои подчиненные узлы. Таким образом, полная схема распределенной информационной базы при наличии более чем двух уровней неизвестна никакому из узлов.

15.3.2.2. Разрешение коллизий

На основе отношения «главный – подчиненный» в распределенной информационной базе организована типовая процедура разрешения коллизий, автоматически выполняемая при приеме сообщения. Считается, что изменение элемента данных, произведенное в главном узле, имеет высший приоритет по отношению к изменению, произведенному в подчиненном узле. Таким образом, если сообщение, пришедшее от подчиненного узла, содержит элемент данных, изменения которого зарегистрированы для этого подчиненного узла, то никаких действий предпринято не будет, то есть этот элемент данных не будет помещен в базу данных и запись регистрации изменений не будет удалена.

Если сообщение, пришедшее от главного узла, содержит элемент данных, изменения которого зарегистрированы для главного узла, то элемент данных будет записан в базу данных, а запись регистрации изменения будет удалена.

15.3.2.3. Начальный образ узла распределенной информационной базы

Обычно при работе с распределенной информационной базой подчиненный узел порождается из главного. То есть для подчиненного узла, определенного в плане обмена, на основании конфигурации и данных, содержащихся в главном узле, создается новая информационная база, соответствующая подчиненному узлу. Такая информационная база называется начальным образом узла распределенной информационной базы.

При создании начального образа выполняются следующие действия:

- Из главного узла в начальный образ без изменений переносится конфигурация.
- В начальном образе в плане обмена, в соответствии с которым создается начальный образ, создаются объекты-узлы, проинициализированные таким образом, чтобы не требовалось дополнительной настройки для начала обмена данными между главным и подчиненным узлами.
- Из главного узла в начальный образ переносятся данные в соответствии с правилами, определяемыми планом обмена (составом плана обмена и результатами выполнения обработчика события *ПриОтправкеДанныхПодчиненному*).

Процедура создания начального образа может повторяться для одного и того же узла неоднократно. Это может иметь смысл для тех случаев, когда информационная база подчиненного узла была потеряна безвозвратно. При создании начального образа все записи регистрации изменений для узла, для которого создается начальный образ, удаляются, так как считается, что история данного подчиненного узла начинается сначала.

15.3.2.4. Сообщение обмена данными в распределенной информационной базе

Для передачи изменений данных и конфигурации в распределенной информационной базе используются сообщения обмена данными, предоставляемые инфраструктурой сообщений. Если в случае применения универсальных механизмов обмена данными разработчик конфигурации сам определяет, что и как помещается в тело сообщения, то в случае распределенной информационной базы структура и состав данных, помещаемых в тело сообщения, четко определены.

Рассмотрим структуру сообщения обмена данными, используемого в распределенной информационной базе. В качестве примера приведем следующее сообщение:

[Копировать в буфер обмена](#)

```
<v8msg:Message xmlns:v8msg="http://v8.1c.ru/messages">
  <v8msg:Header>
    <v8msg:ExchangePlan>УдаленныеСклады</v8msg:ExchangePlan>
    <v8msg:To>Склад1</v8msg:To>
    <v8msg:From>Офис</v8msg:From>
    <v8msg:MessageNo>20</v8msg:MessageNo>
    <v8msg:ReceivedNo>15</v8msg:ReceivedNo>
  </v8msg:Header>
  <v8msg:Body>
    <v8de:Changes xmlns:v8de="http://v8.1c.ru/dataexchange">
      <v8de:Signature>
        7b4d5320-f69c-4a7b-9273-ff56607fc8ab</v8de:Signature>
    </v8de:Changes>
  </v8msg:Body>
</v8msg:Message>
```

```

<v8de:Config xmlns:v8md="http://v8.1c.ru/metadata">
  <!-- Измененные объекты конфигурации -->
<v8de:Digest1>88d3f3a6ba3f4df03c7ec00f154837fc</v8de:Digest1>
<v8de:Digest2>00cf636b02a488103a64c7a2cf81069e</v8de:Digest2>
</v8de:Config>
<v8de:Nodes>
  <v8de:Node>
    <!-- Данные главного узла -->
  </v8de:Node>
  <v8de:Node>
    <!-- Данные подчиненного узла -->
  </v8de:Node>
</v8de:Nodes>
<v8de:Data>
  <!-- Измененные элементы данных -->
</v8de:Data>
</v8de:Changes>
</v8msg:Body>
</v8msg:Message>

```

Как видно из примера, все особенности сообщения обмена данными, используемого в распределенной информационной базе, сосредоточены в теле сообщения. Тело сообщения (элемент **Body**, относящийся к пространству имен <http://v8.1c.ru/messages>) содержит один-единственный элемент XML – **Changes**, относящийся к пространству имен <http://v8.1c.ru/dataexchange>. Внутри этого элемента сосредоточены все данные, передаваемые при обмене данными в распределенной информационной базе:

- **Changes** может содержать четыре вложенных элемента, относящихся к тому же пространству имен:
 - **Signature** содержит «подпись» плана обмена, в соответствии с которым получено сообщение.
 - **Config** содержит изменения конфигурации, а также данные, идентифицирующие состояние конфигурации.
 - Необязательные элементы **Metadata**, вложенные в **Config**, содержат изменения отдельных объектов конфигурации. Если изменения конфигурации не передаются в сообщении, то элементы **Metadata** отсутствуют. Такие элементы могут присутствовать только в сообщениях, передаваемых от главного узла подчиненному. Элементы **Digest1** и **Digest2** содержат цифровые подписи передаваемых в данном сообщении изменений конфигурации и всей конфигурации за вычетом изменений. Элементы **Digest1** и **Digest2** присутствуют во всех сообщениях, передаваемых от главного узла подчиненному и наоборот.
 - **Nodes** может присутствовать только в сообщениях, передаваемых от главного узла подчиненному. Этот элемент содержит два вложенных элемента **Node**, первый из которых содержит данные главного узла (отправителя), а второй – подчиненного (получателя).
 - И, наконец, элемент **Data** содержит измененные элементы данных, передаваемые в сообщении. Элементы данных помещаются в сообщение с помощью XML-сериализации.

15.3.2.5. Конфигурирование плана обмена для работы с распределенной информационной базой

Как было отмечено выше, для того чтобы план обмена мог быть использован для организации распределенной информационной базы, у него должно быть установлено свойство **Распределенная информационная база**.

При конфигурировании плана обмена, используемого для организации распределенной информационной базы, необходимо определить состав плана обмена, так как именно состав определяет номенклатуру данных, по которым будет вестись регистрация изменений и которые могут передаваться при обмене данными в распределенной информационной базе.

Следует отметить, что если в конфигурации уже работающей распределенной информационной базы в состав обмена будет включен еще какой-либо объект метаданных, то это не повлечет автоматически никаких действий по регистрации изменений элементов данных, соответствующих этому объекту метаданных. Таким образом, если при включении в состав плана обмена нового объекта есть необходимость передать уже существующие данные другим узлам, об этом следует позаботиться отдельно.

15.3.2.6. Обработчики событий плана обмена

При необходимости у объекта **ПланыОбменаОбъект.<Имя плана обмена>** могут быть определены обработчики событий **ПриОтправкеДанныхПодчиненному**, **ПриОтправкеДанныхГлавному**, **ПриПолученииДанныхОтПодчиненного**, **ПриПолученииДанныхОтГлавного**, позволяющие управлять помещением в сообщение и считыванием из сообщения отдельных элементов данных.

Обработчик события **ПриОтправкеДанныхПодчиненному** вызывается при помещении элемента данных в сообщение, отправляемое подчиненному узлу распределенной информационной базы. Первый параметр содержит сам элемент данных; значением второго параметра при вызове обработчика является **ОтправкаЭлементаДанных.Авто**, но это значение может быть изменено обработчиком. Если значением второго параметра в результате выполнения обработчика останется **ОтправкаЭлементаДанных.Авто**, то это соответствует поведению по умолчанию (элемент данных будет помещен в сообщение). Если второму параметру в обработчике будет присвоено значение **ОтправкаЭлементаДанных.Удалить**, то в сообщении будет помещен объект, соответствующий удалению элемента данных. Для объектов базы данных – это объект типа **УдалениеОбъекта**, проинициализированный ссылкой на удаляемый объект базы данных; а для наборов записей – это пустой набор записей. Только для менеджера записи константы поведение, соответствующее значению **ОтправкаЭлементаДанных.Удалить**, не отличается от поведения, соответствующего значению **ОтправкаЭлементаДанных.Авто**. Если же второму параметру будет присвоено значение **ОтправкаЭлементаДанных.Игнорировать**, то в сообщении не будет помещено ничего соответствующего элементу данных, переданному в первом параметре.

Обработчик события **ПриОтправкеДанныхГлавному** отличается от предыдущего только тем, что вызывается при помещении элемента данных в сообщение, отправляемое главному узлу.

Обработчик **ПриПолученииДанныхОтПодчиненного** вызывается после считывания элемента данных из сообщения и перед записью элемента в базу данных. Первый параметр содержит элемент данных, считанный из сообщения. Второй параметр при вызове обработчика имеет значение **ПолучениеЭлементаДанных.Авто**, что соответствует поведению по умолчанию. То есть если в данном узле не было зарегистрировано изменений элемента данных для узла-отправителя, то элемент будет записан в базу данных. Если же изменения зарегистрированы, то элемент в базу данных записан не будет. Значение второго параметра может быть изменено обработчиком. Если второй параметр получит значение **ПолучениеЭлементаДанных.Принять**, то элемент данных будет записан в базу данных независимо от того, были ли зарегистрированы его изменения для узла-отправителя. Если изменения были зарегистрированы, то соответствующая запись регистрации изменений удаляется. Если же второй параметр получит значение **ПолучениеЭлементаДанных.Игнорировать**, то никаких действий предпринято не будет (элемент не будет записан в базу данных и не будет произведено никаких действий с записями регистрации изменений). Третий параметр позволяет управлять регистрацией изменений элемента данных для узла-отправителя. При вызове обработчика данный параметр имеет значение **Ложь**. Если это значение не будет изменено обработчиком, то никаких дополнительных действий произведено не будет. Если в обработчике события присвоить параметру значение **Истина**, то при отсутствии зарегистрированных изменений элемента данных для узла-отправителя такая регистрация будет выполнена.

Обработчик события **ПриПолученииДанныхОтГлавного** имеет тот же набор параметров, что и предыдущий обработчик, и отличается от него тем, что вызывается при чтении сообщения, принимаемого от главного узла распределенной информационной базы. Соответственно, несколько отличаются действия, выполняемые при получении элемента данных. Значения второго параметра **ПолучениеЭлементаДанных.Авто** и **ПолучениеЭлементаДанных.Принять**, присвоенные обработчиком, производят одинаковый эффект, так как согласно принятой стратегии разрешения коллизий в подчиненном узле считанный из сообщения элемент данных должен быть записан в базу данных независимо от того, зарегистрированы его изменения или нет. В остальном значения параметров и смысл предпринимаемых действий аналогичны описанным выше.

15.3.3. Работа с распределенной информационной базой

В работе с распределенной информационной базой можно выделить следующие основные действия:

- создание начального образа подчиненного узла распределенной информационной базы;
- запись сообщения обмена данными для отправки в другой узел распределенной информационной базы;
- чтение сообщения обмена данными, отправленного из другого узла распределенной информационной базы.

Если для доступа к возможностям, предоставляемым универсальными механизмами обмена данными, без встроенного языка не обойтись, то перечисленные действия могут быть выполнены как из встроенного языка, так и интерактивно с помощью команд меню **Еще (Все действия)** формы списка плана обмена или иными средствами, определенными при конфигурировании.

Кроме того, для каждого из узлов распределенной информационной базы может быть установлен и, соответственно, получен его главный узел. Эта операция не может быть отнесена к основным операциям, производимым в распределенной информационной базе, поэтому интерактивное действие как аналог этой операции не предусмотрено.

15.3.3.1. Работа с распределенной информационной базой из встроенного языка

15.3.3.1.1. Создание начального образа подчиненного узла распределенной информационной базы

Для создания начального образа подчиненного узла распределенной информационной базы объект

ПланыОбменаМенеджер содержит метод СоздатьНачальныйОбраз().

В качестве первого параметра данному методу должно быть передано значение типа ПланыОбменаСсылка.<Имя плана обмена>, представляющее собой ссылку на подчиненный узел распределенной информационной базы, или значение типа ПланыОбменаОбъект.<Имя плана обмена>, представляющее такой узел.

Второй параметр должен содержать строку соединения, идентифицирующую информационную базу, в которую будет помещен начальный образ подчиненного узла. Информационная база, используемая для создания начального образа, должна быть пустой или не должна существовать вовсе.

Как было отмечено выше, конфигурация информационной базы переносится в начальный образ без изменений. Данные, соответствующие объектам метаданных, не входящим в состав плана обмена, не будут помещены в начальный образ. Кроме того, при переносе элементов данных в начальный образ для каждого элемента данных вызывается обработчик события ПриОтправкеДанныхПодчиненному. Соответственно, в начальный образ могут попасть только те данные, которые должны туда попасть в соответствии с правилами, установленными планом обмена.

В плане обмена, в соответствии с которым создается начальный образ, в информационной базе начального образа будут созданы и проинициализированы два узла, соответственно для главного и «этого» узлов. Таким образом, сразу после создания начального образа информационная база подчиненного узла готова к обмену данными со своим главным узлом.

Все операции по переносу данных в начальный образ выполняются в рамках одной транзакции базы данных главного узла. Это необходимо для обеспечения согласованности данных, помещаемых в начальный образ. Однако при интенсивном изменении данных в информационной базе главного узла другими пользователями возможны конфликты между транзакцией, в рамках которой выполняется создание начального образа, и транзакциями других пользователей. В таких случаях перед созданием начального образа целесообразно установить монопольный режим для информационной базы главного узла.

ПРИМЕЧАНИЕ. Работа информационной базы в монопольном режиме не переводит базу данных MS SQL в однопользовательский режим (single user).

15.3.3.1.2. Запись сообщения обмена данными распределенной информационной базы

Как было показано, сообщение обмена данными распределенной информационной базы отличается от сообщения обмена данными в общем виде вполне конкретным наполнением тела сообщения. Таким же образом отличается и процедура записи сообщения обмена данными распределенной информационной базы от записи сообщения в общем виде.

Для записи тела сообщения обмена данными распределенной информационной базы менеджер планов обмена содержит метод ЗаписатьИзменения(). В качестве первого параметра данному методу передается объект типа ЗаписьСообщенияОбмена, через который осуществляется запись сообщения. У данного объекта уже должен быть вызван метод НачатьЗапись(), но еще не должен быть вызван метод ЗакончитьЗапись().

Необязательный второй параметр указывает максимальное число элементов данных, которые будут помещаться в сообщение в рамках одной транзакции. Если в качестве значения параметра указано 0 (значение по умолчанию), то все формирование сообщения будет выполнено в одной транзакции. В таком режиме обеспечивается наилучшая согласованность данных, помещаемых в сообщение, но возможны конфликты с транзакциями, выполняемыми другими пользователями. При меньшем числе элементов, обрабатываемых в одной транзакции, вероятность конфликтов транзакций меньше, но больше вероятность помещения в сообщение несогласованных данных. Поэтому рекомендуется без острой необходимости не использовать для второго параметра значения, отличные от значения по умолчанию.

Метод ЗаписатьИзменения() записывает элемент XML Changes в тело сообщения обмена данными, помещая туда всю требуемую информацию, как это было показано выше. При этом в соответствующие фрагменты сообщения помещаются объекты конфигурации и элементы данных, изменения которых были зарегистрированы для узла-получателя сообщения. Фактическое помещение элемента данных в сообщение определяется результатом выполнения обработчика события ПриОтправкеДанныхГлавному (ПриОтправкеДанныхПодчиненному), вызываемому у объекта типа ПланыОбменаОбъект.<Имя плана обмена>, представляющего узел-получатель сообщения.

Ниже приведен типовой фрагмент кода, выполняющий запись сообщения обмена данными распределенной информационной базы.

```
ЗаписьXML = Новый ЗаписьXML();
ЗаписьXML.ОткрытьФайл(ИмяФайлаСообщения);
Узел = ПланыОбмена.УдаленныеСклады.НайтиПоКоду(КодУзла);
ЗаписьСообщения = ПланыОбмена.СоздатьЗаписьСообщения();
ЗаписьСообщения.НачатьЗапись(ЗаписьXML, Узел);
```

Копировать в буфер обмена

15.3.3.1.3. Чтение сообщения обмена данными распределенной информационной базы

Для чтения тела сообщения обмена данными распределенной информационной базы менеджер планов обмена содержит метод `ПрочитатьИзменения()`.

В качестве первого параметра методу передается объект типа `ЧтениеСообщенияОбмена`, через который осуществляется чтение сообщения в целом. У этого объекта уже должен быть вызван метод `НачатьЧтение()`, но еще не вызван метод `ЗакончитьЧтение()`.

СОВЕТ. Настоятельно рекомендуется, чтобы при обращении к методу `НачатьЧтение()` объекта `ЧтениеСообщенияОбмена` значение второго параметра не указывалось или же указывалось значение по умолчанию – `ДопустимыйНомерСообщения.Больший`.

Это связано с тем, что вся логика обмена данными в распределенной информационной базе построена с учетом того, что отдельные сообщения обмена данными могут быть утеряны, но при этом не допускается повторный прием одного и того же сообщения.

В качестве значения необязательного второго параметра может быть указано максимальное число элементов данных, считываемых из сообщения и помещаемых в базу данных в рамках одной транзакции. Значение параметра по умолчанию 0 означает, что все чтение сообщения будет выполняться в одной транзакции. Если все чтение сообщения выполняется в одной транзакции, то в случае возникновения ошибок не может оказаться так, что часть элементов данных была считана из сообщения и помещена в базу данных, а часть – нет. Но при таком режиме может случиться, что число изменений базы данных, которое нужно выполнить в рамках одной транзакции, окажется слишком большим. Кроме того, повышается вероятность конфликтов между транзакцией, в которой происходит чтение сообщения, и транзакциями, выполняемыми другими пользователями. Для того чтобы избежать неприятностей такого рода, введена возможность ограничения числа элементов данных, обрабатываемых в одной транзакции. Но если нет острой нужды, рекомендуется использовать режим по умолчанию, то есть производить считывание всех элементов данных в одной транзакции.

Последовательность действий, выполняемая методом `ПрочитатьИзменения()`, выглядит примерно следующим образом:

- Считывается и проверяется «подпись» плана обмена, чтобы с максимальной достоверностью убедиться, что сообщение прибыло от ожидаемого плана обмена ожидаемой конфигурации.
- Из сообщения считываются изменения конфигурации. При считывании проводится проверка цифровых подписей конфигурации, чтобы исключить возможность того, что в узле-отправителе и текущем узле распределенной информационной базы находятся несовместимые конфигурации. Для каждого из считанных измененных объектов конфигурации проверяется, является ли этот объект конфигурации фактически измененным. Напомним, что измененные объекты конфигурации могут содержаться только в сообщениях, передаваемых от главного узла подчиненному.
- Удаляются записи регистрации изменений объектов конфигурации и элементов данных, отправленных в сообщениях, для которых получено подтверждение приема. Напомним, что максимальный номер, принятый узлом-отправителем сообщений, содержится в заголовке сообщения и доступен через свойство `НомерПринятого` объекта `ЧтениеСообщенияОбмена`.
- Если в результате проверки цифровых подписей конфигурации удалось установить, что конфигурация в текущем узле распределенной информационной базы отличается от конфигурации узла-отправителя, то возможны два варианта дальнейших действий:
 - Если текущий узел является главным по
 - Если же текущий узел является подчиненным по отношению к узлу-отправителю, то перед продолжением приема сообщения необходимо обновить конфигурацию базы данных, приведя ее в соответствие с конфигурацией узла-отправителя. При обновлении конфигурации базы данных в нее будут перенесены ранее принятые, сохраненные и измененные объекты конфигурации. Далее в обоих случаях вызывается исключение с соответствующим текстом сообщения об ошибке. И если в первом случае сообщение не может быть прочитано, то во втором случае после обновления конфигурации базы данных, которое может быть выполнено в режиме `Конфигуратор`, то же самое сообщение может быть успешно прочитано и принято.
 - Если производится прием сообщения обмена данными, отправленного из главного узла распределенной информационной базы, то данные текущего узла и узла-отправителя приводятся в соответствие с данными об этих узлах, содержащимися в сообщении обмена данными. Если сообщение обмена данными получено от подчиненного узла, то там таких данных быть не должно (подробнее см. [здесь](#)).

- Производится считывание из сообщения и запись в базу данных элементов данных. При этом в сообщении могут содержаться только элементы данных, соответствующие объектам метаданных, входящих в состав плана обмена, к которому относится данное сообщение. Для каждого прочитанного из сообщения элемента данных у объекта типа `ПланыОбменаОбъект.<Имя плана обмена>` вызывается обработчик события `ПриПолученииДанныхОтГлавного (ПриПолученииДанныхОтПодчиненного)`. Дальнейшие действия по каждому из элементов данных определяются результатами действий обработчика (подробнее см. [здесь](#)).

15.3.3.1.4. Получение и установка главного узла распределенной информационной базы

Для корневого узла распределенной информационной базы характерно, что у него нет главного узла. У всех остальных главный узел есть. В типовом сценарии работы с распределенной информационной базой нужды в принудительной установке главного узла не возникает. Но в некоторых случаях такая возможность может все-таки пригодиться. Например, может возникнуть потребность выделить какое-либо из поддеревьев распределенной информационной базы в самостоятельную информационную базу или переподчинить какой-либо из узлов распределенной информационной базы.

Для установки главного узла распределенной информационной базы предназначен метод `УстановитьГлавныйУзел()` объекта `ПланыОбменаМенеджер`. Метод имеет один параметр. Если в качестве параметра данному методу передано значение типа `ПланыОбменаОбъект.<Имя плана обмена>` или `ПланыОбменаСсылка.<Имя плана обмена>`, то у плана обмена, к которому относится ссылка или объект, должно быть установлено свойство `Распределенная информационная база`. В этом случае для данной информационной базы будет установлен главный узел. Если в качестве значения параметра передано `Неопределено`, то назначение главного узла отменится.

Для успешного выполнения данного метода требуется, чтобы у информационной базы не было других активных пользователей, в том числе и в режиме Конфигуратор.

Для получения главного узла предназначен метод `ГлавныйУзел()` объекта `ПланыОбменаМенеджер`. Если текущая информационная база не является узлом распределенной информационной базы или главный узел для нее не определен (она сама является корневым узлом), метод возвращает `Неопределено`. Если же главный узел для информационной базы определен, то метод возвращает значение типа `ПланыОбменаСсылка.<Имя плана обмена>`.

ПРИМЕЧАНИЕ. Для отмены назначения главного узла информационной базы можно применить параметр `/ResetMasterNode` командной строки пакетного запуска конфигуратора.

В том случае, если в распределенной информационной базе используются предопределенные данные, то операции с главным узлом следует выполнять особым образом:

- Отключение информационной базы от распределенной информационной базы:
 1. Для информационных баз, которые использовались в «1С:Предприятии» версий от 8.3.3 до 8.3.5 (независимо от установленного режима совместимости), для всех объектов конфигурации, для которых в конфигураторе созданы предопределенные элементы и свойство `Обновление предопределенных данных` установлено в значение `Авто`, необходимо вызвать метод `УстановитьИнициализациюПредопределенныхДанных (Истина)`;
 2. Отключить информационную базу от главного узла с помощью вызова метода `УстановитьГлавныйУзел (Неопределено)` или с помощью параметра `/ResetMasterNode` командной строки пакетного запуска конфигуратора.
- Восстановление конфигурации в узле распределенной информационной базы:
 1. Установить режим обновления предопределенных данных информационной базы в значение `ОбновлениеПредопределенныхДанных . НеОбновлятьАвтоматически`. Это можно сделать с помощью метода языка `УстановитьОбновлениеПредопределенныхДанныхИнформационнойБазы()` или с помощью параметра `/SetPredefinedDataUpdate -DoNotUpdateAutomatically` командной строки пакетного запуска конфигуратора.
 2. Получить ссылку на главный узел информационной базы с помощью метода `ГлавныйУзел()` и запомнить полученную ссылку.
 3. Отключить информационную базу от главного узла с помощью вызова метода `УстановитьГлавныйУзел (Неопределено)` или с помощью параметра `/ResetMasterNode` командной строки пакетного запуска конфигуратора.
 4. Выполнить действия по восстановлению конфигурации и другие операции по восстановлению работоспособности узла.

5. Восстановить ссылку на главный узел с помощью метода `УстановитьГлавныйУзел()` с указанием в качестве параметра ссылки на узел, который ранее был установлен в качестве главного узла (получен на шаге 2).

6. Установить режим обновления предопределенных данных информационной базы в значение `ОбновлениеПредопределенныхДанных.Авто`. Это можно сделать с помощью метода языка `УстановитьОбновлениеПредопределенныхДанныхИнформационнойБазы()` или с помощью параметра `/SetPredefinedDataUpdate -Auto` командной строки пакетного запуска конфигуратора.

15.3.3.2. Интерактивная работа с распределенной информационной базой

Как было отмечено выше, ряд наиболее часто выполняемых действий с распределенной информационной базой может быть произведен интерактивно через меню **Еще (Все действия)** формы списка плана обмена (или пиктограммы командной панели формы списка). Соответствующие команды вставляются в меню при автоматическом заполнении.

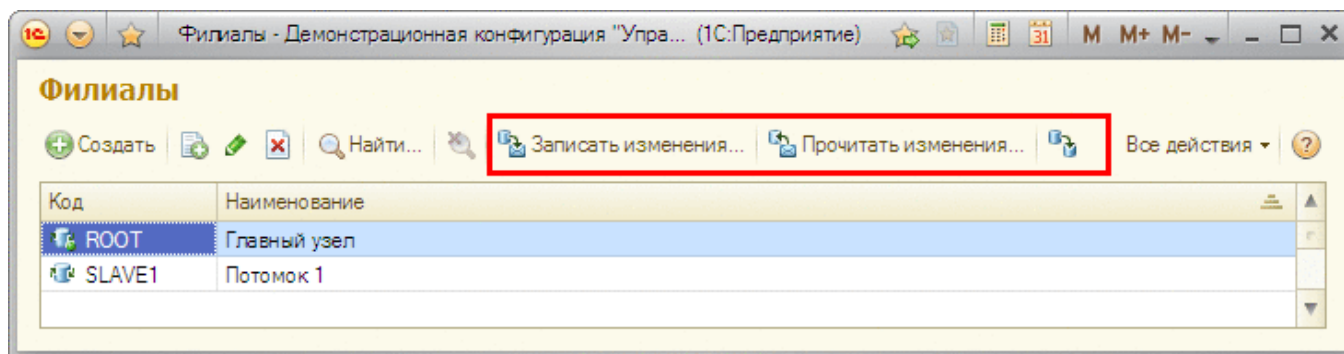


Рис. 330. Список планов обмена

Для выполнения каждой из этих команд следует выделить в списке узел плана обмена, для которого предполагается выполнить команду, а затем выбрать соответствующий пункт меню **Еще (Все действия)**.

Команда **Еще – Создать начальный образ... (Все действия – Создать начальный образ...)** предназначена для создания начального образа подчиненного узла распределенной информационной базы. После обращения к данному пункту меню на экране появляется диалог, в котором предлагается выбрать тип расположения информационной базы и ее параметры (в случае создания образа в клиент-серверном варианте). После нажатия кнопки **Создать начальный образ** в этом диалоге начинается создание начального образа. Сама процедура создания начального образа при интерактивном выполнении ничем не отличается от процедуры, выполняемой при обращении к методу `СоздатьНачальныйОбраз()` объекта `ПланыОбменаМенеджер`.

Команда **Еще – Записать изменения... (Все действия – Записать изменения...)** предназначена для записи в файл сообщения обмена данными. После обращения к этому пункту меню на экране появляется диалог. В нем необходимо задать число элементов данных, обрабатываемых в одной транзакции. Затем нажать кнопку **Записать и сохранить в файл**. Откроется диалог выбора файла обмена. После выбора следует нажать кнопку **Открыть**, начнется выгрузка данных.

Команда **Еще – Прочитать изменения... (Все действия – Прочитать изменения...)** предназначена для чтения из файла сообщения обмена данными. После обращения к данному пункту меню на экране появляется диалог. В нем необходимо задать число элементов данных, обрабатываемых в одной транзакции. Затем нажать кнопку **Выбрать файл и прочитать изменения**. Откроется диалог выбора файла обмена. После выбора следует нажать кнопку **Открыть**, начнется загрузка данных.

Все операции по загрузке/выгрузке данных (включая создание начального образа информационной базы) выполняются на стороне сервера «1С:Предприятия».

15.3.4. Сценарии обмена данными в распределенной информационной базе

Обработчики событий `ПриОтправкеДанныхПодчиненному`, `ПриОтправкеДанныхГлавному`, `ПриПолученииДанныхОтПодчиненного`, `ПриПолученииДанныхОтГлавного` позволяют достаточно гибко управлять обменом данными в распределенной информационной базе. С использованием этих обработчиков может быть построено большое разнообразие сценариев обмена данными. В этом разделе в качестве примера будет рассмотрена организация нескольких сценариев.

15.3.4.1. Поведение по умолчанию

Данный сценарий является наиболее простым и соответствует поведению распределенной информационной базы по умолчанию. Для этого сценария характерно следующее:

- каждое изменение элемента данных, произведенное в любом из узлов распределенной информационной базы, стремится распространиться по всем узлам;
- разрешение коллизий производится на основании отношения узлов «главный – подчиненный».

Для реализации такого сценария все обработчики не должны изменять значения переданных им параметров, или же обработчики могут быть не определены вовсе.

15.3.4.2. Распределение данных по подчиненным узлам

Данный сценарий подразумевает, что для некоторых элементов данных, для которых он реализуется, выполняется следующее:

- вся совокупность элементов данных присутствует в главном узле;
- присутствие того или иного элемента данных в том или ином подчиненном узле определяется на основе сравнения значений некоторых реквизитов элемента данных с реквизитами подчиненного узла плана обмена;
- разрешение коллизий производится на основании отношения узлов «главный – подчиненный».

Для реализации данного сценария нужно обеспечить, чтобы при записи сообщения обмена данными в главном узле в сообщение не попадали элементы данных, которые не должны присутствовать в подчиненном узле.

Кроме того, если значения реквизитов элемента данных могут быть изменены в подчиненном узле, то необходимо обеспечить, чтобы при получении сообщения обмена данными в главном узле производилась регистрация изменений для тех объектов, которых в соответствии со значениями их реквизитов в подчиненном узле быть не должно.

Для более детального рассмотрения примера предположим, что в качестве типа элементов данных, для которых реализуется сценарий, выступает документ *РасходнаяНакладная*. У данного документа имеется реквизит *Склад* типа *СправочникСсылка.Склады*. Обмен данными организован в соответствии с планом обмена *Склады*. У этого плана обмена также определен реквизит *Склад* типа *СправочникСсылка.Склады*. В соответствии с этим планом обмена организована распределенная информационная база, в которой корневым узлом является центральный офис, а его подчиненными узлами – склады. У каждого из подчиненных узлов плана обмена значение реквизита *Склад* установлено так, чтобы обозначать, какому складу соответствует этот узел. Все документы *РасходнаяНакладная* должны присутствовать в корневом узле, а условием присутствия документов в подчиненных узлах является равенство значений реквизитов *Склад* в документе и узле плана обмена.

В этом случае для того, чтобы документы *РасходнаяНакладная* не попадали в те подчиненные узлы, куда они попадать не должны, обработчик события *ПриОтправкеДанныхПодчиненному* должен иметь следующий вид:

[Копировать в буфер обмена](#)

```
Процедура ПриОтправкеДанныхПодчиненному(ЭлементДанных, ОтправкаЭлемента)
    ТипДанных = ТипЗнч(ЭлементДанных);
    Если ТипДанных = Тип("ДокументОбъект.РасходнаяНакладная") Тогда
        Если ЭлементДанных.Склад <> Склад Тогда
            ОтправкаЭлемента = ОтправкаЭлементаДанных.Удалить;
        КонецЕсли;
    КонецЕсли;
КонецПроцедуры
```

В приведенном примере обработчика анализируется тип элемента данных, и если он равен *ДокументОбъект.РасходнаяНакладная*, то значение реквизита *Склад* документа сравнивается со значением реквизита *Склад* узла плана обмена. Если значения реквизитов равны, то значение параметра *ОтправкаЭлемента* можно не изменять (при вызове параметр имеет значение *ОтправкаЭлементаДанных.Авто*). При этом в сообщение будет помещено XML-представление документа. Если же значения реквизитов не равны, то параметру *ОтправкаЭлемента* присваивается значение *ОтправкаЭлементаДанных.Удалить*. В этом случае в сообщение будет помещено XML-представление объекта *УдалениеОбъекта*, проинициализированного ссылкой на соответствующий документ *РасходнаяНакладная*.

Может показаться странным, что в случае неравенства значений реквизитов *Склад* параметру *ОтправкаЭлемента* присваивается значение *ОтправкаЭлементаДанных.Удалить*, а не *ОтправкаЭлементаДанных.Игнорировать*, так как в случае значения *ОтправкаЭлементаДанных.Удалить* XML-представление объекта *УдалениеОбъекта* будет помещаться в сообщения, отправляемые всем подчиненным узлам, кроме того узла, в который будет отправлен сам документ. Таким образом, в значительной части случаев объект *УдалениеОбъекта* будет отправлен тем узлам, где никогда не было документа, который требуется удалить.

Это действительно так, но в данном примере рассмотрен наиболее общий случай. Если же, например, известно, что значение реквизита *Склад* документа *РасходнаяНакладная* может быть установлено только при создании

документа и в дальнейшем не может быть изменено, то параметру `ОтправкаЭлемента` в данном обработчике действительно могло бы быть присвоено значение `ОтправкаЭлементаДанных.Игнорировать`.

Если же значение реквизита `Склад` документа `РасходнаяНакладная` может быть изменено в подчиненном узле, то в плане обмена необходимо определить обработчик события `ПриПолученииДанныхОтПодчиненного` следующего вида:

[Копировать в буфер обмена](#)

```
Процедура ПриПолученииДанныхОтПодчиненного(ЭлементДанных, ПолучениеЭлемента, ОтправкаНазад)
    ТипДанных = ТипЗнч(ЭлементДанных);
    Если ТипДанных = Тип("ДокументОбъект.РасходнаяНакладная") Тогда
        Если ЭлементДанных.Склад <> Склад Тогда
            ОтправкаНазад = Истина;
        КонецЕсли;
    КонецЕсли;
КонецПроцедуры
```

В приведенном примере обработчика анализируется тип элемента данных, и если он равен `ДокументОбъект.РасходнаяНакладная`, то значение реквизита `Склад` документа сравнивается со значением реквизита `Склад` узла плана обмена. Если значения реквизитов равны, то значения параметров `ПолучениеЭлемента` и `ОтправкаНазад` можно не изменять, обеспечив этим поведение по умолчанию при приеме элемента данных. Если же значения реквизитов не равны, то параметру `ОтправкаНазад` присваивается значение `Истина`. Тем самым гарантируется, что изменения документа будут зарегистрированы и при отправке сообщения подчиненному узлу будет отправлен объект `УдалениеОбъекта`, если, конечно, реквизит `Склад` документа не будет изменен в главном узле так, что он окажется равен значению реквизита `Склад` соответствующего узла плана обмена.

Если же значение реквизита `Склад` документа `РасходнаяНакладная` не может быть изменено после создания документа, то обработчик `ПриПолученииДанныхОтПодчиненного` можно не определять.

15.3.4.3. Нестандартное разрешение коллизий

Данный сценарий подразумевает, что для некоторых элементов данных, для которых он реализуется, выполняется следующее:

- каждое изменение элемента данных, произведенное в любом из узлов распределенной информационной базы, стремится распространиться по всем узлам;
- разрешение коллизий производится на основании отношения узлов «главный – подчиненный», но более высокий приоритет имеет подчиненный узел.

Для рассмотрения данного случая воспользуемся приведенным выше примером с документом `РасходнаяНакладная` и планом обмена `Склады`.

В данном случае требуется определить обработчики событий `ПриПолученииДанныхОтПодчиненного` и `ПриПолученииДанныхОтГлавного`. Обработчик `ПриПолученииДанныхОтПодчиненного` будет иметь следующий вид:

[Копировать в буфер обмена](#)

```
Процедура ПриПолученииДанныхОтПодчиненного(ЭлементДанных, ПолучениеЭлемента, ОтправкаНазад)
    ТипДанных = ТипЗнч(ЭлементДанных);
    Если ТипДанных = Тип("ДокументОбъект.РасходнаяНакладная") Тогда
        ПолучениеЭлемента = ПолучениеЭлементаДанных.Принять;
    КонецЕсли;
КонецПроцедуры
```

Приведенный обработчик весьма прост: проверяется тип элемента данных, и если элемент данных относится к интересующему нас типу, то параметру `ПолучениеЭлемента` присваивается значение `ПолучениеЭлементаДанных.Принять`, что приводит к безусловному приему элемента данных, независимо от того, зарегистрированы его изменения или нет.

Обработчик события `ПриПолученииДанныхОтГлавного` выглядит следующим образом:

[Копировать в буфер обмена](#)

```
Процедура ПриПолученииДанныхОтГлавного(ЭлементДанных, ПолучениеЭлемента, ОтправкаНазад)
    ТипДанных = ТипЗнч(ЭлементДанных);
    Если ТипДанных = Тип("ДокументОбъект.РасходнаяНакладная") Тогда
        Если ПланыОбмена.ИзменениеЗарегистрировано(Ссылка, ЭлементДанных) Тогда
            ПолучениеЭлемента = ПолучениеЭлементаДанных.Игнорировать;
        КонецЕсли;
    КонецЕсли;
КонецПроцедуры
```

Этот обработчик несколько сложнее. Если элемент данных относится к интересующему нас типу, то производится проверка: зарегистрированы ли изменения элемента данных для узла-отправителя сообщения. Если изменения зарегистрированы, то параметру `ПолучениеЭлемента` присваивается значение `ПолучениеЭлементаДанных.Игнорировать`. В результате прочитанный элемент данных не записывается в базу данных, а регистрация изменений сохраняется, что позволит поместить элемент данных в сообщение, отправляемое главному узлу.

15.3.4.4. Другие сценарии

Рассмотренные сценарии являются только некоторыми из возможных сценариев организации обмена данными в распределенной информационной базе. На базе описанных выше обработчиков можно реализовать большое разнообразие различных сценариев. Какие-то из них могут быть комбинацией рассмотренных выше, а какие-то – принципиально другими.

Глава 16. Механизм XDTO

16.1. Общая информация

Механизм XDTO является универсальным способом представления данных для взаимодействия с различными внешними источниками данных и программными системами.

Аббревиатура XDTO обозначает [XML Data Transfer Objects](#).

Механизм XDTO позволяет создать модель представления данных (модель типов и значений), которая, с одной стороны, обеспечивает возможность просто и естественно манипулировать данными в среде системы «1С:Предприятие», а с другой стороны, хорошо приспособлена для прозрачного преобразования данных в другие форматы, главным образом XML.



Рис. 331. Общая схема XDTO

Можно выделить несколько задач, для решения которых используется механизм XDTO:

- обмен данными между конфигурациями системы «1С:Предприятие» с разными схемами данных;
- обмен данными на основе схем XML, не привязанных к той или иной конфигурации (например, обмен с информационными системами, построенными не на основе системы «1С:Предприятие»);
- организация работы с Web-сервисами. Механизм XDTO позволяет описывать типы параметров и возвращаемых значений Web-сервисов, а также манипулировать передаваемыми и возвращаемыми данными.

Механизм XDTO обладает следующими ключевыми свойствами:

- обеспечивает работу с XML,
- привычная модель работы с данными.

В настоящее время обмен данными с различными программными платформами и системами реализуется с использованием XML: XML-документы служат для представления данных, а схема XML используется для описания форматов и структур данных. Механизм XDTO позволяет создавать требуемые для обмена схемы XML и формировать XML-документы, удовлетворяющие этим схемам.

В то же время использование механизма XDTO позволяет выполнять эти действия в привычной для большинства разработчиков системы «1С:Предприятие» манере.

Разработчик имеет дело с типами и объектами данных, объекты данных содержат свойства, свойствам присваиваются значения и т. д. При манипулировании данными с помощью XDTO разработчик максимально изолирован от подробностей, связанных с тем, как эти данные представлены в XML. Конечно, совсем избавиться от этих подробностей невозможно, но важно, что они проявляются только там, где это действительно нужно.

16.2. Фабрика XDTO

Ключевым понятием механизма XDTO является фабрика XDTO. Фабрика XDTO содержит описание всех типов, с которыми оперирует некоторая система. В частности, для любой конфигурации системы «1С:Предприятие» существует глобальная фабрика XDTO, которая описывает все типы, используемые в конфигурации, в терминах XDTO (эта фабрика XDTO доступна через свойство глобального контекста [ФабрикаXDTO](#)).

Все описания типов, которые содержит фабрика XDTO, сгруппированы в один или несколько пакетов XDTO. Если проводить аналогию между XDTO и XML, то можно сказать, что пакет XDTO соответствует схеме XML. Таким образом, фабрика XDTO может соответствовать нескольким схемам XML.

Фабрика XDTO является полностью самодостаточной. То есть любой из типов, зарегистрированных в фабрике XDTO, может ссылаться только на типы из той же самой фабрики XDTO.

В общем случае фабрика XDTO создается одновременно на основании описаний всех типов, которые должны быть зарегистрированы в фабрике. Для создания фабрики XDTO средствами встроенного языка используется конструктор объекта [ФабрикаXDTO](#), которому передается набор схем XML, содержащийся в объекте [НаборСхемXML](#). Сценарий, при котором типы XDTO добавляются в фабрику по одному или группами, не поддерживается.

Ниже приведен пример создания фабрики XDTO на основе схемы XML, содержащейся в файле XML. Так как механизм XDTO представляет собой абстракцию, построенную «над» XML, то для получения схемы XML из файла XML необходимо последовательно «пройти» несколько уровней работы с данными XML:

- сначала низкоуровневое чтение/запись файлов XML;
- затем объектную модель XML, из которой уже может быть получен объект встроенного языка *СхемаXML*, содержащий данные схемы XML.

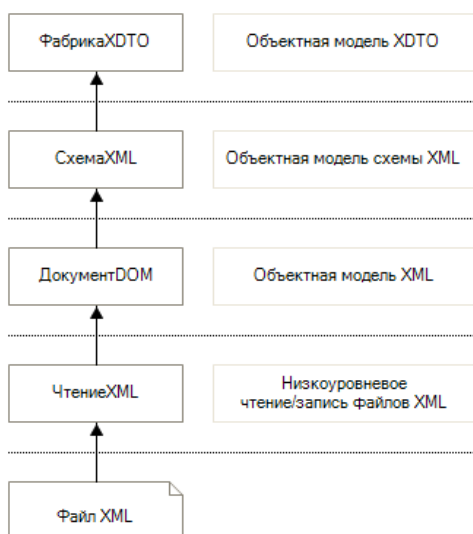


Рис. 332. Создание фабрики XDTO

Пример:

```
// Создать фабрику XDTO на основе схемы XML,
// содержащейся в файле XML
// Создать объект чтения XML по умолчанию
НовоеЧтениеXML = Новый ЧтениеXML;
// Открыть файл XML
НовоеЧтениеXML.ОткрытьФайл("D:/MySchema.xsd");
// Создать построитель документа DOM по умолчанию
НовыйПостроительDOM = Новый ПостроительDOM;
// Прочитать файл XML в документ DOM
НовыйДокументDOM = НовыйПостроительDOM.Прочитать(НовоеЧтениеXML);
// Создать построитель схемы XML по умолчанию
НовыйПостроительСхемыXML = Новый ПостроительСхемXML;
// Получить схему XML из документа DOM
НоваяСхемаXML = НовыйПостроительСхемыXML.СоздатьСхемуXML(НовыйДокументDOM);
// Создать набор схем XML по умолчанию
НовыйНаборСхемXML = Новый НаборСхемXML;
// Добавить схему XML в набор схем XML
НовыйНаборСхемXML.Добавить(НоваяСхемаXML);
// Создать фабрику XDTO на основе набора схем XML
НоваяФабрикаXDTO = Новый ФабрикаXDTO(НовыйНаборСхемXML);
```

[Копировать в буфер обмена](#)

В приведенном примере сначала создается объект *ЧтениеXML* и открывается файл XML, расположенный на диске. После этого с помощью построителя документа DOM создается объект *ДокументDOM*, содержащий данные файла XML. Затем с помощью построителя схемы XML на основе документа DOM создается новый объект *СхемаXML*, содержащий данные схемы XML. В заключение создается пустой набор схем XML, в который добавляется имеющаяся схема XML, и на основании этого набора создается фабрика XDTO.

16.2.1. Получение фабрики XDTO из файла-схемы XSD

```
Схемы = Новый Массив;
Схемы.Добавить("путь1");
Схемы.Добавить("путь2");
Пакеты = Новый Массив;
Пакеты.Добавить(ФабрикаXDTO.Пакеты.Получить("URI пространства имен пакета из конфигурации 1"));
Пакеты.Добавить(ФабрикаXDTO.Пакеты.Получить("URI пространства имен пакета из конфигурации 2"));
МояФабрика = СоздатьФабрикуXDTO(Схемы, Пакеты);
```

[Копировать в буфер обмена](#)

В отличие от произвольной фабрики XDTO, которую может создать разработчик, глобальная фабрика XDTO создается системой автоматически, при создании новой информационной базы, и допускает добавление типов XDTO по одному или группами. Для этого используются средства визуального конструирования, позволяющие добавлять пакеты XDTO в ветку дерева метаданных *Общие – XDTO*. Все пакеты, содержащиеся в глобальной фабрике XDTO, можно разделить на три вида:

- Один пакет XDTO, содержащий описание типов платформы. Этот пакет является одинаковым для всех конфигураций системы «1С:Предприятие».
- Один пакет XDTO, содержащий описание типов конфигурации, созданных в результате редактирования метаданных (создания и изменения свойств справочников, документов и пр.).
- Один или несколько пакетов XDTO, описанных непосредственно в дереве конфигурации, в ветке *Общие – XDTO*.

Пакет XDTO содержит описание некоторого множества типов, принадлежащих одному пространству имен – пространству имен пакета. Кроме

непосредственно описаний типов пакет XDTO может содержать ссылки на пакеты, которые используются данным пакетом, а также список определений глобальных свойств пакета.

Ссылки на другие пакеты содержатся в свойстве **Зависимости** пакета XDTO и представляют собой объект **КоллекцияПакетовXDTO**. Пакеты этой коллекции содержат типы из пространства имен, на которые имеются ссылки в данном пакете.

В пакете можно ссылаться на глобальные свойства из других пакетов.

Отсутствует поддержка циклов в директивах импорта и включения XSD-схем.

16.3. Типы данных XDTO

Каждый из типов данных XDTO является либо типом значения XDTO, либо типом объекта XDTO. Соответственно, для описания типа значения используется объект **ТипЗначенияXDTO**, а для описания типа объекта – **ТипОбъектаXDTO**.

Объект **ТипЗначенияXDTO** используется для описания типов простых неделимых значений, в которых не могут быть выделены отдельные составляющие. Примерами простых значений являются разнообразные строки, числа, даты и т. п.

Объект **ТипОбъектаXDTO** используется для описания типов экземпляров данных, имеющих некоторое состояние, представляемое как совокупность значений свойств этого экземпляра данных. При этом типы свойств этого экземпляра данных могут являться как типами значений XDTO, так и типами объектов XDTO.

И **ТипЗначенияXDTO**, и **ТипОбъектаXDTO** имеют два одинаковых свойства:

- **Имя** – имя типа;
- **URIПространстваИмен** – URI пространства имен, в котором определен данный тип.

Значения этих свойств совпадают с аналогичными параметрами, с которыми тип определяется в схеме XML. Имя типа и URI пространства имен образуют уникальный идентификатор типа. Имя типа должно быть обязательно определено. При этом свойство **URIПространстваИмен** может содержать пустую строку, хотя это и нежелательно.

16.3.1. Тип значения XDTO

Тип значения XDTO в соответствии с правилами для **simple type** из схемы XML может определяться тремя способами:

- ограничением, когда задается базовый тип (свойство **БазовыйТип**) и набор ограничений на множество возможных значений (свойство **Фасеты**);
- объединением, когда тип получается в результате объединения нескольких типов значений (объединяемые типы перечисляются в свойстве **ТипыЧленовОбъединения**);
- списком, когда значение представляет собой список значений (тип значения элементов, составляющих список значений, задается в свойстве **ТипЭлементаСписка**).

Помимо свойств **Имя** и **URIПространстваИмен** тип значения XDTO содержит следующие свойства:

- **БазовыйТип** – базовый тип для данного типа значения XDTO. Базовые типы могут наследоваться, но только от других типов значений XDTO. Допустимое множество значений унаследованного типа представляет собой подмножество возможных значений базового типа. Верхним уровнем в иерархии простых типов является предопределенный тип **anySimpleType** из пространства имен <http://www.w3.org/2001/XMLSchema>. Все типы значений прямо или опосредованно унаследованы от этого типа. Типы, образованные объединением или списком, всегда непосредственно унаследованы от **anySimpleType**.
- **Фасеты** – список фасетов, ограничивающих множество допустимых значений по отношению к базовому типу. Список фасетов задается только для типов значений XDTO, определенных ограничением базового типа. Каждый отдельный фасет представляет собой пару: имя фасета и значение. Определен список имен допустимых фасетов. Причем не любой из допустимых фасетов может быть применен к любому типу. Список фасетов и применимость их к тому или иному типу определяются по правилам XML Schema (<http://www.w3.org/TR/xmlschema-2/>).
- **ТипыЧленовОбъединения** – список типов, образующих объединение. Объединяться могут только типы значений XDTO. Если тип образован объединением, то список **ТипыЧленовОбъединения** содержит по крайней мере один тип. При этом список **Фасеты** должен быть пустым, а свойство **ТипЭлементаСписка** должно возвращать неопределенное значение.
- **ТипЭлементаСписка** – в случае, когда тип значения XDTO определяется списком, данное свойство показывает тип элемента списка. При этом списки **Фасеты** и **ТипыЧленовОбъединения** должны быть пустыми.
- Список имен допустимых фасетов (определяется системным перечислением **ВидФасетаXDTO**):
 - **Длина** – фасет длины. Содержит количество единиц длины, причем единица длины имеет различный смысл для различных типов. Для типов **string** и **anyURI** длина содержит количество символов. Для типов **hexBinary** и **base64Binary** длина содержит количество байт двоичных данных. Для типов, определяемых списком, длина содержит количество элементов списка.
 - **МаксВключающее** – фасет максимума, включающего границу. Ограничивает пространство значений данного типа максимальным значением. Любое значение данного типа меньше либо равно указанному значению.
 - **МаксДлина** – фасет максимальной длины. Содержит максимальное количество единиц длины, причем единица длины имеет различный смысл для различных типов. Для типа **string** максимальная длина содержит максимальное количество символов. Для типов **hexBinary** и **base64Binary** максимальная длина содержит максимальное количество байт двоичных данных. Для типов, определяемых списком, максимальная длина содержит максимальное количество элементов списка.
 - **МаксИсключающее** – фасет максимума, не включающего границу. Ограничивает пространство значений данного типа максимальным значением. Любое значение данного типа меньше указанного значения.

- **МинВключающее** – фасет минимума, включающего границу. Ограничивает пространство значений данного типа минимальным значением. Любое значение данного типа больше либо равно указанному значению.
- **МинДлина** – фасет минимальной длины. Содержит минимальное количество единиц длины, причем единица длины имеет различный смысл для различных типов. Для типа `string` минимальная длина содержит минимальное количество символов. Для типов `hexBinary` и `base64Binary` минимальная длина содержит минимальное количество байт двоичных данных. Для типов, определяемых списком, минимальная длина содержит минимальное количество элементов списка.
- **МинИсключающее** – фасет минимума, не включающего границу. Ограничивает пространство значений данного типа минимальным значением. Любое значение данного типа больше указанного значения.
- **Образец** – фасет образца. Содержит регулярное выражение, определяющее пространство значений данного типа.
- **Перечисление** – фасет перечисления. Определяет набор допустимых значений данного типа.
- **ПробельныеСимволы** – фасет пробельных символов. Может принимать одно из трех значений:
 - **Сохранять** – строка может содержать любые пробельные символы.
 - **Заменять** – строка не должна содержать `#x9` (табуляция), `#xA` (перевод строки) и `#xD` (возврат каретки). Если они существуют, то должны быть заменены символом `#x20` (пробел).
 - **Сворачивать** – дополнительно к требованиям, указанным для значения `replace`, строка не должна содержать парных символов `#x20` (пробел), а также лидирующих и завершающих символов `#x20` (пробел).
- **ЦифрВсего** – фасет общего количества цифр. Содержит общее количество разрядов числа (целая часть плюс дробная часть).
- **ЦифрДробнойЧасти** – фасет количества цифр дробной части. Содержит количество разрядов дробной части числа.

Инфраструктура XDTO определяет набор предопределенных типов значений XDTO. Этот набор совпадает с набором примитивных типов, определенных в XML Schema Part 2: Datatypes. Предопределенные типы образуют иерархию в соответствии с XML Schema Part 2: Datatypes. Имена типов совпадают с именами типов XML Schema и принадлежат URI пространства имен – <http://www.w3.org/2001/XMLSchema>. Предопределенные типы являются автоматически зарегистрированными в любой фабрике XDTO.

16.3.2. Тип объекта XDTO

Помимо свойств `Имя` и `URIПространстваИмен` тип объекта XDTO содержит следующие свойства:

- **БазовыйТип** – базовый тип для данного типа. Это может быть только тип объекта XDTO. Базовым в иерархии типов объектов XDTO является предопределенный тип `anyType` из пространства имен <http://www.w3.org/2001/XMLSchema>. Все типы объектов XDTO прямо или опосредованно унаследованы от этого типа.
- **Открытый** – признак того, является ли тип объекта XDTO открытым. Данное свойство показывает, может ли экземпляр объекта XDTO содержать дополнительные свойства, не определенные в его типе, то есть реализует модель `open content`. Соответствует появлению в XML-схеме для данного типа описаний: `<anyAttribute>`, `<any>`.
- **Абстрактный** – признак того, является ли тип объекта XDTO абстрактным. Соответствует появлению в схеме для данного элемента атрибута `abstract="true"`.
- **Упорядоченный** – признак того, является ли порядок следования элементов, представляющих значения свойств, строго соответствующим порядку следования свойств в типе объекта XDTO. Если задана модель контента `xsd:all`, то порядок следования элементов XML может быть произвольным. При этом допустимым является порядок, соответствующий порядку следования свойств в типе. То есть если свойство `Упорядоченный` имеет значение `Ложь`, то на входе порядок следования элементов XML не контролируется, а на выходе определяется порядком следования свойств, если только признак `Последовательный` не имеет значение `Истина`.
- **Последовательный** – это свойство показывает, содержит ли экземпляр соответствующего объекта XDTO последовательность XDTO. Данный признак равен значению `Истина` в тех случаях, когда порядок следования вложенных элементов XML не может однозначно определяться порядком следования свойств в типе (например, в схеме XML контент задан как `<sequence ... maxOccurs=10 ... >`) или для соответствующего типа XML в схеме определен атрибут `mixed="true"`. Последовательность XDTO позволяет задать в явном виде порядок следования элементов, как они будут представлены в документе XML. Для объектов типов, у которых свойство `Последовательный` установлено в значение `Ложь`, порядок следования вложенных элементов соответствует порядку следования свойств.
- **Смешанный** – свойство показывает, определен ли в схеме XML для данного типа `mixed content`. Если значение свойства `Смешанный` равно `Истина`, то значение `Последовательный` обязательно равно `Истина`, так как `mixed content` невозможно смоделировать без применения последовательности XDTO.
- **Свойства** – список свойств, определенных для данного типа объекта XDTO. Каждое из свойств представляется в виде экземпляра объекта `СвойствоXDTO`. Список содержит полный список свойств, в том числе свойства, определенные в базовом типе.

Существует предопределенный тип объекта XDTO с именем `anyType` и URI пространства имен <http://www.w3.org/2001/XMLSchema>. Данный тип является базовым для любого типа объекта XDTO, но у него нет базового типа. Он является открытым, не абстрактным, подразумевает наличие последовательности и имеет пустой список свойств.

Данный тип объекта XDTO соответствует типу `anyType`, определенному в XML Schema Part 2: Datatypes.

16.3.3. Свойство XDTO

Отдельно взятое свойство отдельного типа объекта XDTO описывается с помощью экземпляра объекта `СвойствоXDTO`. Это означает, что один и тот же экземпляр объекта `СвойствоXDTO` не может быть использован для описания свойств в различных типах объектов XDTO и двух различных свойств одного типа объекта XDTO.

Объект `СвойствоXDTO` содержит следующие свойства:

- **Имя** – имя свойства. В пределах одного типа объекта XDTO имена свойств должны быть уникальными.

При формировании модели данных XDTO на основе схемы XSD, имена свойств XDTO образуются на основании имен атрибутов и элементов, описанных в схеме. Построение модели типа выполняется последовательно: сначала формируется список свойств на основе атрибутов, затем – на основе элементов, в порядке объявления в схеме.

При этом имя приводится в соответствие с правилами именования, принятыми во встроенном языке. Символы, допустимые в имени XML (например: ".", "-«'"), но недопустимые с точки зрения имени для встроенного языка – заменяются на символ "_". В случае дублирования имени атрибута и элемента, дубликату назначается имя, расширенное числовым суффиксом (начиная от 1).

- **Тип** – тип свойства. Может быть как экземпляром объекта `ТипЗначенияXDTO`, так и экземпляром объекта `ТипОбъектаXDTO`.

- **ВерхняяГраница** – свойство типа объекта XDTO может быть определено как содержащее одно или множество значений. Свойство считается содержащим одно значение, если равно 1. Если же свойство `ВерхняяГраница` больше 1, то считается, что оно может содержать множество значений. Такое свойство в структуре объекта моделируется как список (не путать со списком в описании типа значения XDTO). Свойство `ВерхняяГраница` показывает максимальное количество значений свойства. Значение больше 1 может быть задано только для свойств, представляемых в виде элемента XML. Свойство `ВерхняяГраница` соответствует атрибуту `xsd:maxOccurs` в XML Schema. Значение -1 соответствует `unbounded`.

- **НижняяГраница** – минимальное количество значений свойства. Минимальное количество значений свойства может принимать значения меньше или равно 0. Естественно, значение `НижняяГраница` должно быть меньше или равно значению `ВерхняяГраница` (если, конечно, `ВерхняяГраница` не равно -1);

- **ВозможноПустое** – показывает, может ли свойство принимать неопределенное значение. Неопределенное значение свойства представляется в XML в виде элемента следующего вида: `<elem xsi:nil="true" />`. Таким образом, свойство `ВозможноПустое`, равное `Истина`, может быть определено только для свойств с формой представления `Элемент`. Свойство `ВозможноПустое` соответствует атрибуту `xsd:nilable` в XML Schema. Если значение свойства `ВерхняяГраница` больше 1, неопределенное значение является допустимым для элемента списка значений свойства.

- **ЗначениеПоУмолчанию** – значение свойства по умолчанию. Значением свойства по умолчанию может быть только `ЗначениеXDTO`. При этом данное значение должно быть того же типа, что и тип свойства или же унаследованного типа. При создании объекта XDTO свойство, если оно допускает единственное значение, принимает значение по умолчанию (метод `Установлено()` объекта XDTO возвращает значение `Ложь` для этого свойства). Для свойств с множеством значений список значений изначально пуст, независимо от того, определено или нет значение по умолчанию.

- **Фиксированное** – указывает, является ли значение свойства фиксированным. Если установлено в значение `Истина`, то само фиксированное значение можно получить через свойство `ЗначениеПоУмолчанию`.

- **Форма** – форма представления свойства в XML. Это может быть `Текст`, `Элемент` или `Атрибут`. Если формой представления является `Атрибут` или `Текст`, то значение свойства `ВерхняяГраница` не может быть больше 1. Если свойство принимает значение `Текст`, то значение свойства `НижняяГраница` также должно быть равным 1. У одного типа только одно свойство может иметь форму представления `Текст`, при этом остальные свойства должны иметь форму представления `Атрибут`.

- **ЛокальноеИмя** – локальное имя атрибута или элемента, используемого для представления свойства. Для свойств с формой представления `Текст` – пустая строка.

- **URIПространстваИмен** – URI пространства имен для атрибута или элемента, используемого для представления свойства. Пустая строка, если пространство имен отсутствует.

16.4. Экземпляры данных XDTO

Экземпляры данных XDTO могут являться значениями XDTO (`ЗначениеXDTO`) или объектами XDTO (`ОбъектXDTO`).

16.4.1. Значение XDTO

Значение XDTO представляет собой простое неделимое значение, в котором не могут быть выделены отдельные составляющие. Примерами простых значений являются разнообразные строки, числа, даты и т. п. Экземпляры простых значений являются немутабельными.

Новое значение XDTO может быть создано с помощью метода `Создать()` фабрики XDTO:

- на основе типа значения XDTO и значения;
- на основе типа значения XDTO и лексического представления значения.

Ниже приведены примеры создания значения XDTO.

Пример:

```
ГлобальнаяФабрикаXDTO = ФабрикаXDTO;
// Создать значение XDTO из ссылки
СсылкаНаЭлементСправочника = Справочники.Номенклатура.НайтиПоКоду("0000001");
ТипЗначенияXDTOСоздаваемогоЗначения = ГлобальнаяФабрикаXDTO.Тип("urn:schemas-v8-1c-ru:config-data", "CatalogRef.Номенклатура");
НовоеЗначениеXDTO = ГлобальнаяФабрикаXDTO.Создать(ТипЗначенияXDTOСоздаваемогоЗначения, СсылкаНаЭлементСправочника);
// Создать значение XDTO из лексического представления значения
ТипЗначенияXDTOСоздаваемогоЗначения = ГлобальнаяФабрикаXDTO.Тип("http://www.w3.org/2001/XMLSchema", "dateTime");
НовоеЗначениеXDTO = ГлобальнаяФабрикаXDTO.Создать(ТипЗначенияXDTOСоздаваемогоЗначения, "2006-04-20T12:00:30");
```

Новое значение XDTO может быть получено также путем чтения файла XML.

Пример:

```
ГлобальнаяФабрикаXDTO = ФабрикаXDTO;
// Прочитать данные значения XDTO из файла XML
```



```

НовоеЧтениеXML = Новый ЧтениеXML;
НовоеЧтениеXML.ОткрытьФайл("D:/Exchange.xml");
...
НовоеЗначениеXDTO = ГлобальнаяФабрикаXDTO.ПрочитатьXML(НовоеЧтениеXML);
Значение XDTO может быть записано в файл XML.
ГлобальнаяФабрикаXDTO = ФабрикаXDTO;
// Записать данные значения XDTO в файл XML
НоваяЗаписьXML = Новый ЗаписьXML;
НоваяЗаписьXML.ОткрытьФайл("D:/Exchange.xml");
...
ГлобальнаяФабрикаXDTO.ЗаписатьXML(НоваяЗаписьXML, НовоеЗначениеXDTO);

```

16.4.2. Объект XDTO

В противовес простому значению состояние объекта XDTO представляется как совокупность значений его свойств. Экземпляры объекта XDTO являются мутабельными, то есть во время жизни объекта XDTO его состояние может быть изменено путем изменения значений отдельных его свойств. В качестве значений свойств могут фигурировать любые экземпляры данных XDTO, как значение XDTO, так и объект XDTO. Когда значением свойства является объект XDTO, говорят, что значением свойства является ссылка на объект.

Новый объект XDTO может быть создан с помощью метода `Создать()` фабрики XDTO, на основе типа объекта XDTO. После этого следует присвоить соответствующие значения свойствам объекта XDTO. Ниже приведен пример создания объекта XDTO и заполнения его свойств.

Пример:

[Копировать в буфер обмена](#)

```

ГлобальнаяФабрикаXDTO = ФабрикаXDTO;
// Создать "пустой" объект XDTO
ТипОбъектаXDTOСоздаваемогоОбъекта = ГлобальнаяФабрикаXDTO.Тип("http://www.lc.ru/demos/products", "Номенклатура");
НовыйОбъектXDTO = ГлобальнаяФабрикаXDTO.Создать(ТипОбъектаXDTOСоздаваемогоОбъекта);
// Заполнить значения свойств объекта XDTO
ОбъектСправочника = СсылкаНаЭлементСправочника.ПолучитьОбъект();
НовыйОбъектXDTO.Наименование = ОбъектСправочника.Наименование;
НовыйОбъектXDTO.ПолноеНаименование = ОбъектСправочника.ПолноеНаименование;
НовыйОбъектXDTO.ЗакупочнаяЦена = ОбъектСправочника.ЗакупочнаяЦена;
НовыйОбъектXDTO.ШтрихКод = ОбъектСправочника.ШтрихКод;

```

Так же как и значение XDTO, данные объекта XDTO могут быть прочитаны из файла XML или записаны в файл XML.

Пример:

[Копировать в буфер обмена](#)

```

ГлобальнаяФабрикаXDTO = ФабрикаXDTO;
// Прочитать данные объекта XDTO из файла XML
НовоеЧтениеXML = Новый ЧтениеXML;
НовоеЧтениеXML.ОткрытьФайл("D:/Exchange.xml");
...
НовыйОбъектXDTO = ГлобальнаяФабрикаXDTO.ПрочитатьXML(НовоеЧтениеXML);
...
// Записать данные объекта XDTO в файл XML
НоваяЗаписьXML = Новый ЗаписьXML;
НоваяЗаписьXML.ОткрытьФайл("D:/Exchange.xml");
...
ГлобальнаяФабрикаXDTO.ЗаписатьXML(НоваяЗаписьXML, НовыйОбъектXDTO);

```

При чтении нетипизированных данных, считываемый элемент, в случае наличия атрибутов или дочерних элементов, считывается в `ОбъектXDTO` типа `xsd:anyType`. Данный тип является открытым типом со смешанным содержимым, поэтому текст в элементе интерпретируется не как значение свойства `__content`, а как текст и помещается в последовательность объекта.

Например, если выполняет чтение элемента вида `<element attr="attr_value">element value</element>`, то текст `element value` можно получить следующим образом `ОбъектXDTO.Последовательность().ПолучитьТекст(0)`.

Объект `ОбъектXDTO` содержит следующие методы:

- `Тип()` – возвращает тип данного объекта XDTO (`ТипОбъектаXDTO`);
- `Установить(<Выражение>)`, `Установить(<Свойство>, <Значение>)` – позволяет установить значение свойства;
- `Выражение` – выражение на XPath, указывающее свойство;
- `Свойство` – имя свойства;
- `Значение` – устанавливаемое значение свойства.

Если `Выражение` задано неправильно или `Значение` не может быть присвоено свойству (например, тип несовместим с типом свойства), то вызывается исключение. Если свойству присваивается неопределенное значение, а свойство `ВозможноПустое` равно `Ложь`, то выдается исключение. Если свойству присваивается ссылка на объект XDTO и ссылка на этот объект XDTO уже является значением какого-либо другого свойства, то данная ссылка перестает быть значением этого другого свойства.

Цепочки ссылок на объекты XDTO, содержащиеся в свойствах объектов, не могут образовывать циклов. Поэтому при присваивании ссылки на объект XDTO, вызывающий образование цикла, вызывается исключение. Если свойство допускает множество значений, то для него применение метода `Установить()` недопустимо и приводит к вызову исключения. При присваивании значения свойству производится проверка на допустимость присваивания данного типа значения свойству. Значение может быть присвоено в том случае, если его тип совпадает с типом свойства, является унаследованным от типа свойства или является одним из типов, входящих в объединение. При присваивании, если формой представления значения свойства в XML является `Текст` или `Атрибут`, производится приведение значения к типу свойства. Если формой представления является `Элемент`, значение присваивается как есть.

- `Получить(<Свойство>)`, `Получить(<Выражение>)` – получение значения свойства:

- **Свойство** – имя свойства.
- **Выражение** – выражение на XPath, указывающее свойство. Для свойств с множеством значений данный метод возвращает список значений свойства – **СписокXDTO**. Все операции модификации значений свойства должны выполняться через этот список.
- **Сбросить(<Свойство>)**, **Сбросить (<Выражение>)** – сброс значения свойства:
 - **Свойство** – имя свойства.
 - **Выражение** – выражение на XPath, указывающее свойство. Действие метода **Сбросить()** для различных свойств различно. Для свойств, допускающих множество значений (**ВерхняяГраница > 1**), выполнение метода **Сбросить()** приводит к очистке списка значений.
- **Установлено()** – проверяет, установлено ли значение свойства. Непосредственно после создания объекта у всех свойств результатом выполнения метода **Установлено()** будет значение **Ложь**.
- **Последовательность()** – возвращает объект-последовательность (**ПоследовательностьXDTO**), принадлежащий данному объекту XDTO. С помощью последовательности XDTO также можно модифицировать состояние объекта. Данный метод возвращает последовательность XDTO только в том случае, если у типа объекта установлено свойство **Последовательный**.
- **Проверить()** – данный метод позволяет проверить правильность заполнения значений свойств объекта XDTO. При проверке проверяются также объекты, ссылки на которые являются значениями свойств. Предметом проверки является соответствие количества значений свойствам **НижняяГраница** и **ВерхняяГраница**, правильность следования значений свойств в последовательности XDTO, если свойство **Упорядоченный** имеет значение **Истина**. Проверка прекращается при нахождении первой же ошибки. При этом выдается исключение.

16.4.3. Последовательность XDTO

С помощью объекта **ПоследовательностьXDTO** моделируется порядок следования элементов и фрагментов текста, как они выглядят в XML-представлении объекта. Последовательность состоит из пар «свойство – значение». В качестве свойств могут выступать только свойства с формой представления **Элемент**, так как порядок следования атрибутов не важен. Свойство в паре «свойство – значение» может также иметь неопределенное значение. В этом случае считается, что данный элемент последовательности представляет фрагмент текста. Появление элементов последовательности, представляющих фрагменты текста, допустимо только для объектов типов, у которых значение свойства **Смешанный** равно **Истина**.

При формировании содержимого объекта XDTO с помощью присваивания значений свойствам порядок присваивания отражается в последовательности XDTO.

Последовательность XDTO содержит следующие методы:

- **Количество()** – возвращает число элементов последовательности.
- **ПолучитьСвойство(<Индекс>)** – возвращает свойство, которому соответствует значение, находящееся по индексу **Индекс**. Если **Индекс** находится за пределами допустимых значений, выдается исключение. Метод может вернуть неопределенное значение, если элементу последовательности соответствует фрагмент текста из смешанного содержания (текст и элементы).
- **ПолучитьЗначение(<Индекс>)** – возвращает значение, находящееся по индексу **Индекс**. Если **Индекс** находится за пределами допустимых значений, выдается исключение.
- **УстановитьЗначение(<Индекс>, <Элемент>)** – устанавливает значение **Элемент** по индексу **Индекс**. **Индекс** должен иметь значение в диапазоне допустимых индексов. **Элемент** должен иметь допустимое значение для свойства, для которого он устанавливается, или для текста.
- **Добавить(<Свойство>, <Элемент>)** – добавляет пару «свойство – значение» к последовательности. Значение должно быть допустимым для свойства.
- **Добавить(<Текст>)** – добавляет фрагмент текста к последовательности. Если у типа объекта свойство **Смешанный** имеет значение **Ложь**, то выдается исключение.
- **Вставить(<Индекс>, <Свойство>, <Элемент>)** – вставляет пару «свойство – значение» в позицию **Индекс** последовательности. **Индекс** должен иметь значение внутри диапазона индексов. **Элемент** в позиции **Индекс** и все элементы с большими значениями индекса сдвигаются вправо на одну позицию.
- **Вставить(<Индекс>, <Текст>)** – вставляет текст **Текст** в позицию **Индекс** последовательности. **Индекс** должен иметь значение внутри диапазона индексов. **Элемент** в позиции **Индекс** и все элементы с большими значениями индекса сдвигаются вправо на одну позицию.
- **Удалить(<Индекс>)** – удаляет элемент последовательности в позиции **Индекс**. **Индекс** должен иметь значение внутри диапазона допустимых.

16.4.4. Список XDTO

С помощью объекта **СписокXDTO** моделируется список значений для свойств с множественными значениями (**ВерхняяГраница > 1**). Список представляет собой упорядоченный набор объектов, которые могут являться как значениями XDTO, так и объектами XDTO. Среди них могут иметь место неопределенные значения, если свойство **ВозможноПустое** имеет значение **Истина**. Но понятие «установленности» для элемента списка не определено.

Объект **СписокXDTO** содержит следующие методы:

- **Количество()** – возвращает размер списка.
- **Получить(<Индекс>)** – получает значение, находящееся по индексу **Индекс**. **Индекс** должен находиться в диапазоне допустимых. В противном случае выдается исключение.
- **Установить(<Индекс>, <Элемент>)** – устанавливает значение **Элемент** в позицию **Индекс**. Устанавливаемое значение замещает ранее присутствовавшее значение. **Индекс** должен находиться в диапазоне допустимых, а **Элемент** должен быть допустимым для свойства. В

противном случае выдается исключение.

- **Добавить (<Элемент>)** – добавляет значение в хвост списка. **Элемент** должен быть допустимым для свойства. В противном случае выдается исключение.
- **Вставить (<Индекс>, <Элемент>)** – внесение значения **Элемент** в позицию **Индекс**. **Индекс** должен находиться в диапазоне допустимых, а **Элемент** должен быть допустимым для свойства. В противном случае выдается исключение. Значение в позиции **Индекс** и значения с большими позициями сдвигаются вправо на одну позицию.
- **Удалить (<Индекс>)** – удаление значения в позиции **Индекс**. **Индекс** должен находиться в диапазоне допустимых. Значения с большими позициями сдвигаются на освободившееся место.

16.4.5. XPath

Для навигации по дереву объектов могут использоваться выражения на XPath. Строго говоря, это не совсем XPath, а скорее несколько модифицированное подмножество XPath.

Основной конструкцией данного языка является путь к значению, который состоит из отдельных шагов. Шаги в пути отделяются друг от друга символами "/" (слеш). В качестве шага пути выступает имя свойства или предопределенные конструкции "." (точка) и ".." (две точки).

Выражение приведенного ниже вида обозначает свойство с именем **ИмяСвойства** текущего объекта, а именно объекта, у которого вызвали метод **Получить()** или **Установить()**.

ИмяСвойства

[Копировать в буфер обмена](#)

Выражение приведенного ниже вида означает, что у текущего объекта получено значение свойства **ИмяСвойства1**, а у объекта, ссылка на который является значением свойства **ИмяСвойства1**, получено свойство **ИмяСвойства2**.

ИмяСвойства1/ИмяСвойства2

[Копировать в буфер обмена](#)

Соответственно, шаг, обозначенный как точка, означает текущий объект, а две точки – объект-владелец текущего.

Если путь поиска начинается с символа слеш, то это означает поиск от корня дерева объектов. Если какое-либо свойство в пути не найдено, то это вызывает исключение. Если в пути встречается свойство с множественным значением, то результатом является весь список значений данного свойства.

Например, если в пути, приведенном ниже, свойство **Список** имеет множественное значение, то результат данного выражения – это список (**СписокХДТО**) значений данного свойства.

Свойство/Список

[Копировать в буфер обмена](#)

Для того чтобы получить отдельное значение из этого списка, нужно через точку от имени свойства указать 0-базированный индекс значения в списке, как это показано ниже.

Свойство/Список.0

[Копировать в буфер обмена](#)

Индекс должен быть задан как целое число, находящееся в пределах диапазона допустимых индексов. В противном случае будет вызвано исключение.

Можно получить отдельное значение из списка с использованием 1-базированного индекса. Для этого применяется конструкция следующего вида:

Свойство/Список[1]

[Копировать в буфер обмена](#)

Индекс может принимать значения от 1 до числа элементов в списке.

Имеется также возможность поиска в списке (только для объектов). Выражение для поиска выглядит следующим образом:

Свойство/Список[ИмяСвойства='СтрокаПоиска']

[Копировать в буфер обмена](#)

Здесь **Список** – это свойство с множественным значением. Значением списка являются объекты, у которых имеется свойство с именем **ИмяСвойства**. Результатом будет первый объект в списке, значение свойства **Свойство** которого равно строковому значению **'СтрокаПоиска'**. Если ни одного объекта не найдено, то результатом является неопределенное значение. Свойство **ИмяСвойства** также может присутствовать не у всех объектов в списке. А может и не присутствовать ни у кого. Значение, с которым сравнивается значение свойства, может быть задано в виде числа, логического значения (**True** или **False**) или строкового литерала.

Свойство/Список[ИмяСвойства!='СтрокаПоиска']

[Копировать в буфер обмена](#)

Приведенное выше выражение аналогично предыдущему примеру, за исключением того, что результатом будет первый объект в списке, значение свойства **ИмяСвойства** которого не равно строковому значению **'СтрокаПоиска'**.

Ниже приведено определение описываемого подмножества XPath.

<Путь>

[/] <Список шагов>

[Копировать в буфер обмена](#)

<Список шагов>

`<Шаг> [/<Список шагов>] |``<Шаг>`[Копировать в буфер обмена](#)

`<Имя свойства> [<Уточнение>] | .. | . |``<Имя свойства>`[Копировать в буфер обмена](#)

`[<Буква> | _]<Остаток имени>``<Остаток имени>`[Копировать в буфер обмена](#)

`{<Буква> | <Цифра> | _} <Остаток имени> |``<Уточнение>`[Копировать в буфер обмена](#)

`.<0-базированный индекс> |
[<Имя свойства>=<Значение>] |
[<Имя свойства>!=<Значение>] |
[<1-базированный индекс>]``<0-базированный индекс>`[Копировать в буфер обмена](#)

`<Целое без знака>``<1-базированный индекс>`[Копировать в буфер обмена](#)

`<Целое без знака>``<Целое без знака>`[Копировать в буфер обмена](#)

`<Цифра> <Цифры>``<Цифра>`[Копировать в буфер обмена](#)

`0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9``<Цифры>`[Копировать в буфер обмена](#)

`<Цифра> <Цифры> |``<Значение>`[Копировать в буфер обмена](#)

`<Число> | <Строка> | <Булево>``<Число>`[Копировать в буфер обмена](#)

`[+|-]<Целое без знака>[.<Целое без знака>]``<Строка>`[Копировать в буфер обмена](#)

`"<Символы>" | '<Символы>'``<Булево>`[Копировать в буфер обмена](#)

`true | false`

ПРИМЕЧАНИЕ. В строке с ограничителями «"» среди символов не может встречаться «"». Аналогично в строке с разделителями «'» не может встречаться символ «'».

При сравнении значения свойства со значением, заданным в виде литерала, значение, заданное в виде литерала, приводится к типу свойства по правилам приведения, после чего производится сравнение.

16.5. XML-сериализация на основе XDTO

Значения типов конфигураций системы «1С:Предприятие» могут быть сериализованы непосредственно в(из) файл(а) XML на основе XDTO.

Для этого используется объект [СериализаторXDTO](#), который может быть получен с помощью конструктора на основе существующей фабрики XDTO. Работа с объектом [СериализаторXDTO](#) аналогична работе с глобальными процедурами и функциями работы с XML.

Например, сериализация ссылки на справочник [Номенклатура](#) в файл XML может быть выполнена с помощью программного кода.

Пример:

```
// Получить ссылку на элемент справочника Номенклатура
СсылкаНаЭлементСправочника = Справочники.Номенклатура.НайтиПоКоду("0000001");
// Создать сериализатор XDTO для глобальной фабрики XDTO
НовыйСериализаторXDTO = Новый СериализаторXDTO(ФабрикаXDTO);
// Создать объект записи XML и открыть файл
НоваяЗаписьXML = Новый ЗаписьXML;
НоваяЗаписьXML.ОткрытьФайл("D:/Exchange.xml");
// ...
// Сериализовать ссылку в XML
НовыйСериализаторXDTO.ЗаписатьXML(НоваяЗаписьXML, СсылкаНаЭлементСправочника, НазначениеТипаXML.Явное);
```

Ниже приведен пример сериализации ссылки на справочник *Номенклатура* из файла XML.

Пример:

```
// Создать сериализатор XDTO для глобальной фабрики XDTO
НовыйСериализаторXDTO = Новый СериализаторXDTO(ФабрикаXDTO);
// Прочитать данные объекта XDTO из файла XML
НовоеЧтениеXML = Новый ЧтениеXML;
НовоеЧтениеXML.ОткрытьФайл("D:/Exchange.xml");
...
// Сериализовать ссылку из XML
НоваяСсылкаНаСправочник = НовыйСериализаторXDTO.ПрочитатьXML(НовоеЧтениеXML);
```

Копировать в буфер обмена

16.6. Рекомендации по оформлению схем XML

Преобразование *НаборСхемXML* " *ФабрикаXDTO* " *НаборСхемXML* в общем случае не дает на выходе набор схем XML, эквивалентный исходному. Однако следование набору рекомендаций по оформлению схем XML позволит добиться эквивалентности выходного набора схем исходному, а именно:

- для преобразования *НаборСхемXML* " *ФабрикаXDTO* " *НаборСхемXML* будет соблюдаться эквивалентность выходного набора схем исходному;
- успешная проверка правильности заполнения свойств объекта *ОбъектXDTO* (метод *Проверить()*) гарантирует, что представление объекта в XML будет соответствовать схеме XML;
- обеспечивается максимальная гибкость и отсутствие искажений при использовании полиморфизма.

Схема XML, полученная на основе фабрики XDTO, для типов которой не переопределялись значения по умолчанию параметров, ответственных за XML-представление данных, безусловно, соответствует приведенным рекомендациям.

В общем, приведенный далее список рекомендаций представляет собой набор правил, позволяющий добиться наилучших результатов с различных точек зрения.

Схема XML не должна содержать анонимных типов

Недопустимы конструкции следующего вида:

```
<element name="Person">
  <complexType>
    <sequence>
      <element name="FirstName" type="string" />
      <element name="FamilyName" type="string" />
    </sequence>
  </complexType>
</element>
```

Копировать в буфер обмена

Этот фрагмент нужно оформить следующим образом:

```
<element name="Person" type="tns:PersonType">
  <complexType name="PersonType">
    <sequence>
      <element name="FirstName" type="string" />
      <element name="FamilyName" type="string" />
    </sequence>
  </complexType>
</element>
```

Копировать в буфер обмена

Для контента сложных типов следует использовать только модель *sequence*

Для моделирования контента у сложных типов (*complexType*) следует использовать только единственный блок *sequence* без переопределения значений по умолчанию атрибутов *minOccurs* и *maxOccurs*.

Копировать в буфер обмена

```
<complexType name="PersonType">
  <sequence>
    <element name="FirstName" type="string" />
    <element name="FamilyName" type="string" />
  </sequence>
</complexType>
```

Модель *all* не нарушает тождественность исходной и результирующей схемы при преобразовании *НаборСхемXML* " *ФабрикаXDTO* " *НаборСхемXML*, однако имеет ряд ограничений. В частности, значение атрибута *maxOccurs* для элементов в модели *all* не может быть больше 1.

Модель *choice* нарушает эквивалентность исходной и результирующей схем и не позволяет с помощью метода *Проверить()* проверить соответствие заполнения данных объекта схеме XML.

По тем же причинам не следует использовать в рамках одного сложного типа комбинацию из нескольких моделей контента или же определять для `sequence` значения атрибутов `minOccurs` и `maxOccurs`, отличные от значений по умолчанию.

Желательно свойства объектов представлять в XML в виде элементов

Представление свойств как атрибутов XML не влияет на эквивалентность исходной и результирующей схем. Однако имеется ряд ограничений:

- В атрибуте не может быть представлено значение объектного типа – только типов-значений.
- В атрибуте не могут быть представлены свойства с множественными значениями.
- В случае полиморфных типов у свойства, представленного как атрибут XML, может происходить искажение типа значения. А именно: при присваивании значения свойству тип значения приводится точно к типу свойства, так как только для значения свойства в элементе XML можно указать атрибут `xsi:type` , который позволяет точно указать тип значения свойства.

Как следствие не нужно использовать `simpleContent` для `complexType`, так как эта модель для хранения значений подразумевает использование свойств атрибутов и текста включающего элемента XML.

Не рекомендуется использовать одинаковые имена для объявления атрибута и элемента. Правила формирования имени свойства XDTO см. [здесь](#).

Не следует использовать модель `mixed content`

Применение в схеме XML конструкции приведенного ниже вида означает, что в элементе XML, соответствующем описываемому типу, текст может быть перемешан с элементами XML.

```
<complexType name="FormLetter" mixed="true">
</complexType>
```

[Копировать в буфер обмена](#)

Для поддержки такого информационного содержимого у соответствующего объекта `ТипОбъектаXDTO` значения свойств `Последовательный` и `Смешанный` установлены в значение `Истина`, а у каждого экземпляра соответствующего объекта `ОбъектXDTO` появляется последовательность XDTO (объект `ПоследовательностьXDTO`). Управление информационным содержимым таких объектов намного сложнее, чем у объектов, состояние которых представлено только набором значений свойств.

К счастью, в подавляющем большинстве случаев в применении `mixed content` не возникает необходимости.

Свойство `ФормаЭлементовПоУмолчанию` у схемы XML должно иметь значение `Квалифицированная`.

Данная рекомендация является элементом хорошего стиля, и XDTO придерживается этого стиля.

16.7. Правила проверки фабрики XDTO

Общая схема кодирования идентификаторов сообщений об ошибках проверки фабрики XDTO выглядит следующим образом:

```
xdto-<область>-<раздел>[-<правило>]: <описание ошибки>
```

[Копировать в буфер обмена](#)

- `<область>` – область проверки (фабрика XDTO, пакет XDTO, тип значения XDTO, фасет XDTO, тип объекта XDTO);
- `<раздел>` – номер раздела, проверка правила которого завершилась неудачей;
- `<правило>` – правило раздела;
- `<описание ошибки>` – описание ошибки.

16.7.1. Правила проверки непосредственно фабрики XDTO

При проверке непосредственно фабрики XDTO ошибки кодируются префиксом `model`. При этом общий префикс будет иметь следующий вид:

```
xdto-model-<раздел>[-<правило>]: <описание ошибки>
```

[Копировать в буфер обмена](#)

Пакеты, входящие в модель, должны иметь уникальные URI пространства имен – дублирование пакетов в рамках модели запрещено.

Директива импорта должна определять непустое URI пространства имен импортируемого пакета.

Директивы импорта пакетов должны определять существующие пакеты типов.

Пакеты типов, определенные в модели, должны удовлетворять правилам проверки пакетов.

16.7.2. Правила проверки пакета XDTO

При проверке правильности пакета XDTO ошибки кодируются префиксом `package`. При этом общий префикс будет иметь следующий вид:

```
xdto-package-<раздел>[-<правило>]: <описание ошибки>
```

[Копировать в буфер обмена](#)

Пакет XDTO должен иметь установленное свойство `URIПространстваИмен`.

Типы, определяемые в пакете XDTO, могут иметь ссылки только на типы, указанные в списке импортируемых (свойство `Зависимости`).

Директивы импорта должны удовлетворять следующим правилам:

- директивы импорта должны определять пакеты XDTO, в которых не могут содержаться директивы импорта, указывающие на данный пакет

XDTO, – зацикливание импортируемых директив не допускается;

- директива импорта должна определять не пустое свойство `URIПространстваИмен` импортируемого пакета XDTO;
- директивы импорта должны определять существующие пакеты XDTO.

Свойства пакета должны удовлетворять следующим правилам:

- имена свойств пакета должны быть установлены и не являться пустыми;
- имена свойств пакета должны быть уникальны в пределах пакета;
- тип свойства пакета должен быть установлен или определен;
- типы свойств пакета должны быть определены в пакете или его зависимостях;
- задание одновременно имени типа глобального свойства и анонимного определения типа глобального свойства недопустимо;
- свойство пакета не должно ссылаться на определение другого свойства пакета;
- свойство пакета не должно определять границы количества значений свойства;
- свойство пакета может иметь форму представления только `Атрибут` или `Элемент`.

16.7.3. Правила проверки типа значения XDTO

При проверке правильности типа значения XDTO ошибки кодируются префиксом `valueType`. При этом общий префикс будет иметь следующий вид:

```
xdto-valueType-<раздел>[-<правило>]: <описание ошибки>
```

[Копировать в буфер обмена](#)

- Общие правила проверки типа значения XDTO:
- если тип определен в рамках пакета типов, то должны выполняться следующие условия:
 - тип значения XDTO должен иметь установленное свойство `Имя`, содержащее непустое имя;
 - имя типа значения XDTO должно быть уникальным в пределах пакета XDTO (среди всех типов пакета XDTO);
 - если тип определен в рамках другого определения типа значения или в рамках определения свойства объектного типа, то должно выполняться следующее условие:
 - свойство `Имя` определения типа не должно быть установлено;
 - тип значения XDTO не может содержать ссылок на самого себя ни в базовом типе, ни в типе элемента списка, ни в одном из типов объединения на всю глубину иерархии.
- Правила проверки базового типа – свойство `БазовыйТип`:
 - если свойство `БазовыйТип` не установлено:
 - если свойство `Вариант` не установлено, а свойство `ТипыЧленовОбъединения` не установлено, или свойство `Вариант` установлено и имеет значение `Атомарный`;
 - если свойство `ОпределениеТипа` содержит единственное значение, то это определение типа является определением анонимного базового типа;
 - в противном случае базовым типом считается тип `anySimpleType` пространства имен XML-схемы (<http://www.w3.org/2001/XMLSchema>);
 - если свойство `БазовыйТип` определения типа установлено, то должны выполняться следующие условия:
 - базовый тип должен удовлетворять второму правилу проверки пакета XDTO;
 - базовый тип должен являться типом значения XDTO;
 - базовый тип не может являться данным типом значения XDTO.
- Правила проверки типа элемента списка – свойство `ТипЭлементаСписка`:
 - если свойство `ТипЭлементаСписка` определения типа установлено, то должны выполняться следующие условия:
 - тип элемента списка должен удовлетворять второму правилу проверки пакета XDTO;
 - тип элемента списка должен являться типом значения XDTO;
 - тип элемента списка не может являться данным типом значения XDTO;
 - тип значения XDTO, являющийся элементом списка, должен быть либо атомарным, либо объединением типов значений XDTO, состоящим только из атомарных типов значений XDTO;
 - если свойство `Вариант` не установлено, а свойство `БазовыйТип` установлено или свойство `Вариант` установлено и имеет значение `Список`:

- если свойство `ОпределениеТипа` содержит единственное значение, то это определение типа является определением анонимного типа элемента списка;
- в противном случае значение свойства `ТипЭлементаСписка` определяется из соответствующего свойства базового типа значения XDTO.
- Правила проверки типа объединения – свойство `ТипыЧленовОбъединения`:
 - если свойство `ТипыЧленовОбъединения` определения типа XDTO установлено, то должны выполняться следующие условия:
 - тип объединения должен удовлетворять второму правилу проверки пакета XDTO;
 - тип объединения должен являться типом значения XDTO;
 - тип объединения не должен являться данным типом значения XDTO;
 - тип значения XDTO, являющийся объединением, должен быть либо атомарным, либо списком;
 - если свойство `Вариант` не установлено, а свойство `БазовыйТип` установлено или свойство `Вариант` установлено и имеет значение `Объединение`:
 - свойство `ОпределениеТипа` содержит значения, являющиеся анонимными определениями типов объединения;
 - в противном случае значение свойства `ТипыЧленовОбъединения` определяется значением свойства `ТипыЧленовОбъединения` базового типа значения XDTO.
- Наследование типов значений XDTO считается правильным, если выполняются следующие условия:
 - для атомарных типов значений XDTO (после выполнения правил проверки типа элемента списка и правил проверки типа объединения свойства `ТипЭлементаСписка` и `ТипыЧленовОбъединения` не установлены) должны выполняться следующие условия:
 - тип значения XDTO, являющийся базовым, должен быть атомарным, т. е. свойства `ТипЭлементаСписка` и `ТипыЧленовОбъединения` базового типа XDTO не должны быть установлены;
 - предком типа значения XDTO должен быть один из примитивных типов пространства имен схемы XML (<http://www.w3.org/2001/XMLSchema>);
 - состав фасетов, установленных в описании типа значения XDTO, должен соответствовать списку допустимых фасетов для примитивного типа, являющегося предком данного типа XDTO;
 - значение каждого фасета, установленного в типе значения XDTO, должно удовлетворять правилам ограничения эффективного значения аналогичного фасета базового типа XDTO;
 - для типов элемента списка (после выполнения правил проверки типа элемента списка и правил проверки типа объединения установлено свойство `ТипЭлементаСписка`):
 - базовый тип XDTO не может являться типом объединения, т. е. свойство `ТипыЧленовОбъединения` базового типа XDTO не должно быть установлено;
 - если свойство `ТипЭлементаСписка` базового типа XDTO установлено, то значение этого свойства должно определять базовый тип для типа элемента списка, определенного в свойстве `ТипЭлементаСписка` данного типа значения XDTO;
 - если базовым типом является тип `anySimpleType` пространства имен XML схемы (<http://www.w3.org/2001/XMLSchema>), то среди списка фасетов допускается только фасет `ПробельныеСимволы`;
 - в противном случае могут быть определены только фасеты: `Образец`, `Перечисление`, `Длина`, `МинДлина`, `МаксДлина` и `ПробельныеСимволы`;
 - значение каждого фасета, установленного в типе, должно удовлетворять правилам ограничения эффективного значения аналогичного фасета базового типа;
 - для типов объединения (после выполнения правил проверки типа элемента списка и правил проверки типа объединения установлено свойство `ТипыЧленовОбъединения`):
 - базовым типом может являться тип `anySimpleType` пространства имен XML схемы (<http://www.w3.org/2001/XMLSchema>) или тип объединения, т. е. у базового типа должно быть установлено свойство `ТипыЧленовОбъединения`;
 - если базовым типом является тип объединения, должны выполняться следующие условия:
 - количество типов объединения базового типа не должно быть больше количества типов объединения данного типа;
 - типы объединения должны быть потомками соответствующих им типов объединения базового типа в порядке следования в списке типов объединения;
 - если базовым типом является тип `anySimpleType` пространства имен XML схемы (<http://www.w3.org/2001/XMLSchema>), то определение фасетов не допускается;
 - в противном случае могут быть определены только фасеты `Образец` и `Перечисление`;
 - значение каждого фасета, установленного в типе, должно удовлетворять правилам ограничения эффективного значения аналогичного фасета базового типа.
- Если свойство `Вариант` (модель содержания) определения типа установлено, то оно не должно противоречить определению типа:

- если свойство имеет значение **Атомарный**, то тип имеет атомарную модель содержания и должен удовлетворять правилам [xdto-valueType-5.1](#);
- если свойство имеет значение **Список**, то тип имеет модель содержания **Список** и должен удовлетворять правилам [xdto-valueType-5.2](#);
- если свойство имеет значение **Объединение**, то тип имеет модель содержания **Объединение** и должен удовлетворять правилам [xdto-valueType-5.3](#).

16.7.4. Правила проверки типа объекта XDTO

При проверке правильности типа объекта XDTO ошибки кодируются префиксом `objectType`. При этом общий префикс будет иметь следующий вид:

[Копировать в буфер обмена](#)

`xdto-objectType-<раздел>[-<правило>]: <описание ошибки>`

- Общие правила проверки типа объекта XDTO:
 - если тип определен в рамках пакета типов, то должны соблюдаться следующие условия:
 - тип объекта XDTO должен иметь установленное свойство **Имя**, содержащее непустое имя;
 - имя типа объекта XDTO должно быть уникальным в пределах пакета (среди всех типов пакета);
 - если тип определен в рамках свойства объектного типа, то должны соблюдаться следующие условия:
 - свойство **Имя** определения типа не должно быть установлено.
- Правила проверки базового типа – свойство **БазовыйТип**:
 - если свойство **БазовыйТип** не установлено, то базовым типом считается тип **anyType** пространства имен XML схемы (<http://www.w3.org/2001/XMLSchema>);
 - если свойство **БазовыйТип** типа объекта XDTO установлено, то должны выполняться следующие условия:
 - базовый тип должен удовлетворять второму правилу проверки пакета XDTO;
 - базовый тип должен являться типом объекта XDTO;
 - базовый тип не может являться данным типом объекта XDTO.
- Каждое свойство типа объекта XDTO должно удовлетворять следующим правилам:
 - имя свойства должно быть определено;
 - имя свойства не может быть пустым;
 - имя свойства должно быть уникальным для типа объекта XDTO;
 - если свойство **Тип** установлено, то должны выполняться следующие условия:
 - имя типа должно определять существующий тип объекта XDTO или тип значения XDTO;
 - тип свойства должен удовлетворять второму правилу проверки пакета XDTO;
 - свойство не может содержать определение анонимного типа;
 - если свойство **Тип** не установлено, то должны выполняться следующие условия:
 - если у свойства имеется определение анонимного типа, то типом свойства является тип, соответствующий данному определению;
 - в противном случае типом свойства считается тип **anyType** пространства имен XML схемы (<http://www.w3.org/2001/XMLSchema>);
 - если у свойства определено значение по умолчанию, то должны быть выполнены следующие условия:
 - тип свойства **ЗначениеПоУмолчанию** должен быть типом значения XDTO;
 - лексическое представление значения по умолчанию должно соответствовать пространству значений типа свойства XDTO;
 - удовлетворять следующим требованиям:
 - если свойство **ЛокальноеИмя** свойства XDTO не установлено, то в качестве локального имени XML представления свойства XDTO используется свойство **Имя** свойства XDTO;
 - если свойство **URIПространстваИмен** свойства XDTO не установлено, то URI пространства имен XML представления свойства XDTO определяется следующим образом:
 - если формой XML представления свойства XDTO (свойство **Форма**) является **Элемент**, то используется URI пространства имен типа, которому принадлежит данное свойство;
 - в противном случае URI пространства имен XML-представления свойства XDTO считается отсутствующим;
 - свойство XDTO должно быть уникально по XML-представлению в пределах типа объекта XDTO;

- если свойство XDTO имеет форму XML-представления **Текст**, то должны выполняться следующие правила:
 - имя и URI пространства имен должны быть неустановленными или быть пустыми;
 - среди свойств типа объекта XDTO допускается наличие свойств XDTO с формой XML представления **Атрибут**;
 - если свойство XDTO имеет форму XML представления **Элемент**, то наличие среди свойств типа объекта XDTO свойства XDTO с формой XML-представления **Текст** запрещено;
- значение нижней границы количества появления значений свойства **НижняяГраница** может принимать значения неотрицательных целых чисел. Значение нижней границы **НижняяГраница** должно быть меньше или равно значению верхней границы количества появления значений свойств **ВерхняяГраница** при условии, что данное значение не равно -1;
- значение верхней границы количества значений свойства **ВерхняяГраница** может принимать значения неотрицательных целых чисел или -1. Если данное значение равно -1, то это означает неограниченное количество значений свойства;
- если значение свойства **Фиксированное** установлено, то по умолчанию должно быть установлено **ЗначениеПоУмолчанию**, соответствующее пространству значений типа свойства XDTO;
- если установлена ссылка на определение глобального свойства, то должны выполняться следующие правила:
 - определение свойства не может переопределять значения свойств глобального определения;
 - глобальное свойство, на определение которого ссылается данное определение свойства, должно быть определено в рамках данного пакета или пакетов-зависимостей.
- Если среди свойств типа объекта XDTO имеется свойство XDTO, совпадающее по имени или XML-представлению со свойством базового типа, то вид наследования определяется как наследование ограничением. При таком виде наследования должны выполняться следующие условия:
 - для каждого свойства XDTO должны выполняться следующие условия:
 - в базовом типе должно быть определено свойство XDTO с тем же именем – переопределяемое свойство;
 - если базовый тип определяет порядок следования свойств (свойство **Упорядоченный**), то позиция переопределяемого свойства должна быть идентична позиции свойства в типе-наследнике;
 - форма XML представления переопределяемого свойства и свойства данного типа должна совпадать;
 - локальное имя XML представления переопределяемого свойства и свойства данного типа должно совпадать;
 - URI пространства имен XML представления переопределяемого свойства и свойства данного типа должна совпадать;
 - если переопределяемое свойство определяет фиксированное значение, то:
 - наличие фиксированного значения не может быть отменено в типе-наследнике;
 - фиксированное значение в базовом типе и типе-наследнике должно совпадать;
 - нижняя граница количества значений свойства должна быть меньше или равна нижней границе количества значений переопределяемого свойства;
 - верхняя граница количества значений свойства должна быть больше или равна верхней границе количества значений переопределяемого свойства;
 - тип свойства должен являться потомком типу переопределяемого свойства;
 - если базовый тип не обладает смешанным содержанием (свойство **Смешанный**), то смешанная модель не может быть установлена в типе-наследнике;
 - если порядок следования свойств базового типа фиксирован (свойство **Упорядоченный**), то порядок не может быть изменен в типе-наследнике;
 - если базовый тип не определяет наличие последовательности (свойство **Последовательный**), то наличие последовательности не может быть установлено в типе-наследнике;
 - если базовый тип не определяет открытую модель содержания (свойство **Открытый**), то открытая модель содержания не может быть установлена в типе-наследнике.
- В противном случае вид наследования определяется как наследование расширением. Для данного вида наследования должны выполняться следующие правила:
 - если модель содержания базового типа является смешанной (свойство **Смешанный**), то она не может быть изменена в типе-наследнике;
 - если порядок следования свойств базового типа не фиксирован (свойство **Упорядоченный**), то порядок не может быть изменен в типе-наследнике;
 - если базовый тип определяет наличие последовательности (свойство **Последовательный**), то последовательность не может быть запрещена в типе-наследнике;
 - если базовый тип определяет открытую модель содержания (свойство **Открытый**), то модель содержания не может быть изменена в типе-наследнике.
- При любом виде наследования должны выполняться следующие условия:

• если модель содержания является смешанной (свойство **Смешанный**), то наличие последовательности (свойство **Последовательный**) не может быть запрещено;

• если модель содержания является открытой (свойство **Открытый**), то наличие последовательности (свойство **Последовательный**) не может быть запрещено.

16.7.5. Правила ограничения фасетов

При проверке правильности фасета XDTO ошибки кодируются префиксом **facet**. При этом общий префикс будет иметь следующий вид:

`xdto-facet-<раздел>[-<правило>]: <описание ошибки>`

[Копировать в буфер обмена](#)

Правила для фасета **Длина**:

- значение фасета должно совпадать с эффективным значением фасета базового типа;
- если установлено значение фасета **МинДлина**, то оно должно удовлетворять следующим условиям:
 - значение фасета **МинДлина** должно быть меньше или равно значению фасета **Длина**;
 - в базовом типе не установлено значение фасета **Длина**, или значение фасета **МинДлина** не отличается от эффективного значения фасета **МинДлина** базового типа;
- если установлено значение фасета **МаксДлина**, то оно должно удовлетворять следующим условиям:
 - значение фасета **МаксДлина** должно быть меньше или равно значению фасета **Длина**;
 - в базовом типе не установлено значение фасета **Длина**, или значение фасета **МаксДлина** не отличается от эффективного значения фасета **МаксДлина** базового типа.

Правила для фасета **МинДлина**:

- значение фасета должно быть больше или равно эффективному значению фасета в базовом классе;
- значение фасета должно быть меньше или равно эффективному значению фасета **МаксДлина**.

Правила для фасета **МаксДлина**:

- значение фасета должно быть меньше или равно эффективному значению фасета в базовом классе;
- значение фасета должно быть больше или равно эффективному значению фасета **МинДлина**.

Правила для фасета **ПробельныеСимволы**:

- если эффективное значение фасета базового типа равно **collapse**, то значение фасета не может принимать другое значение;
- если эффективное значение фасета базового типа равно **preserve**, то значение фасета не может принимать значение **replace**.

Правила для фасета **МинВключающее**:

- значение фасета должно быть меньше эффективного значения фасета **МаксИсключающее** данного типа;
- значение фасета должно быть больше или равно эффективному значению фасета базового типа;
- если установлено значение фасета **МаксВключающее**, то оно должно быть меньше или равно эффективному значению фасета **МаксВключающее** базового типа.
- если установлено значение фасета **МинИсключающее**, то оно должно быть больше эффективного значения фасета **МинИсключающее** базового типа;
- если установлено значение фасета **МаксИсключающее**, то оно должно быть меньше эффективного значения фасета **МаксИсключающее** базового типа.

Правила для фасета **МинИсключающее**:

- установка значения фасета **МинИсключающее** и **МинВключающее** фасета в определении типа не допускается;
- если установлено значение фасета **МаксВключающее**, то значение фасета **МинИсключающее** должно быть меньше значения фасета **МаксВключающее** данного типа;
- значение фасета должно быть больше или равно эффективному значению фасета базового типа;
- если установлено значение фасета **МаксВключающее**, то оно должно быть меньше или равно эффективному значению фасета **МаксВключающее** базового типа;
- если установлено значение фасета **МинВключающее**, то оно должно быть больше или равно эффективному значению фасета **МинВключающее** базового типа;
- если установлено значение фасета **МаксИсключающее**, то оно должно быть меньше эффективного значения фасета **МинВключающее** базового типа.

Правила для фасета **МаксВключающее**:

- значение фасета должно быть больше или равно значению фасета **МинВключающее** данного типа;
- если установлено значение фасета **МаксИсключающее**, то оно должно быть меньше эффективного значения базового типа;
- если установлено значение фасета **МинВключающее**, то оно должно быть больше или равно эффективному значению базового типа;
- если установлено значение фасета **МинИсключающее**, то оно должно быть больше эффективного значения базового типа.

Правила для фасета **МаксИсключающее**:

- установка значения фасета **МаксИсключающее** и **МаксВключающее** фасета в определении типа не допускается;
- если установлено значение фасета **МинИсключающее**, то значение фасета **МаксИсключающее** должно быть больше значения фасета **МинИсключающее** данного типа;
- значение фасета должно быть меньше или равно эффективному значению фасета базового типа;
- если установлено значение фасета **МаксВключающее**, то оно должно быть меньше или равно эффективному значению базового типа;
- если установлено значение фасета **МинИсключающее**, то оно должно быть больше эффективного значения базового типа;
- если установлено значение фасета **МинВключающее**, то оно должно быть больше эффективного значения базового типа.

Правило для фасета **ЦифрВсего**:

- значение фасета должно быть меньше или равно эффективному значению фасета базового типа.

Правила для фасета **ЦифрДробнойЧасти**:

- значение фасета должно быть меньше или равно эффективному значению фасета **ЦифрВсего**;
- значение фасета должно быть меньше или равно эффективному значению фасета базового типа.

Глава 17. Работа с JSON

17.1. Общая информация

Формат JSON является универсальным способом представления при обмене данными (RFC 7159, <https://tools.ietf.org/html/rfc7159>). Данные в формате JSON представляют собой (в закодированном виде):

- **Объект** – неупорядоченное множество пар `ключ:значение`, заключенный в фигурные скобки (`{}`). Пары `ключ:значение` разделяются запятыми (`,`).
- **Массив** – множество значений. Массив заключается в квадратные скобки (`[]`). Значения разделяются запятыми (`,`).
- **Значение** – может быть строкой, числом, объектом, массивом или литералом `true`, `false`, `null`.
 - **Строка** – набор символов, заключенный в двойные кавычки (`" "`).
 - **Число** – сериализуется с разделителем точка (`.`). Точность числа не ограничена.

Таким образом, с помощью вышеперечисленных элементов допускается описание объектов любой сложности для представления в формате JSON. Например, некоторый код на встроенном языке, формирует некоторый набор данных (в структуре):

```
Данные = Новый Структура;
Данные.Вставить("Фамилия", "Иванов");
Данные.Вставить("Имя", "Иван");
Данные.Вставить("Отчество", "Иванович");
Данные.Вставить("Резерв", "Иванович");
Данные.Вставить("ДеньРождения", ДатаРождения);
Телефоны = Новый Массив;
Телефоны.Добавить("+7-987-123-45-67");
Телефоны.Добавить("+7-986-987-65-43");
Данные.Вставить("Телефоны", Телефоны);
```

[Копировать в буфер обмена](#)

Сериализуя этот объект в JSON, может быть получен следующий документ:

```
{
  "Фамилия": "Иванов",
  "Имя": "Иван",
  "Отчество": "Иванович",
  "ДатаРождения": "2009-02-15T00:00:00Z",
  "Телефоны": [
    "+7-987-123-45-67",
    "+7-986-987-65-43"
  ]
}
```

[Копировать в буфер обмена](#)

JSON не стандартизует формат представления даты. В силу этого представление даты в JSON-документе определяется предпочтениями прикладного разработчика, который формирует документ, и требованиями системы, которая будет обрабатывать JSON-документ. Система «1С:Предприятие» поддерживает несколько форматов представления даты (задается с помощью системного перечисления `ФорматДатыJSON`):

1. Формат ISO (значение `ФорматДатыJSON.ISO`). В этом случае дата сериализуется следующим образом: `"2009-02-15T00:00:00+0400"`.
2. Формат JavaScript (значение `ФорматДатыJSON.JavaScript`). В этом случае дата сериализуется следующим образом: `"new Date(1234656000000)"`.
3. Формат Microsoft (значение `ФорматДатыJSON.Microsoft`). В этом случае дата сериализуется следующим образом: `"\Date(1234656000000)V"` или `"/Date(1234656000000)/"` (в зависимости от режима экранирования символов).

Дата может записываться несколькими вариантами (для примера используется дата 10 мая 2014 13:14:15 в зоне UTC+4):

- как локальная дата: `2014-05-10T13:14:15`.
- как локальная дата с указанием смещения: `2014-05-10T13:14:15+04:00`.
- как дата в UTC: `2014-05-10T09:14:15Z`.

Управлять этим можно с помощью системного перечисления `ВариантЗаписиДатыJSON`. Дату в варианте UTC можно записать в любом формате (ISO, JavaScript и Microsoft), остальные варианты представления даты возможны только в том случае, если сериализация выполняется в формате ISO.

При записи JSON-документа предоставляются возможности по управлению формируемыми данными: настраивать перенос строк, формат сериализации даты и т. д. Эти настройки можно выполнять с помощью объектов `НастройкиСериализацииJSON` и `ПараметрыЗаписиJSON`.

Работа с данными в формате JSON может выполняться в нескольких техниках:

- **Объектная техника** – позволяет формировать простые и небольшие JSON-документы для обмена с внешними приложениями или веб-сервисами. Структура JSON-документа автоматически формируется системой «1С:Предприятие» во время записи документа. Более подробно описание данной техники см. [здесь](#).
- **Потоковая техника** – позволяет работать с данными большого объема без загрузки их в память приложения. Навигация по JSON-документу полностью ложится на прикладного разработчика (как при записи, так и при чтении документа). Более подробно описание

966
данной техники см. [здесь](#).

- Совмещенная техника – позволяет сочетать гибкость потоковой техники и простоту объектной техники. Более подробно описание данной техники см. [здесь](#).

17.2. Объектная техника работы

17.2.1. Общая информация

Система «1С:Предприятие» поддерживает сериализацию следующих данных в формат JSON:

- **Строка** – сериализуется в строку;
- **Число** – сериализуется в число;
- **Булево** – сериализуется в литералы `true` и `false`;
- **Неопределено** – сериализуется в `null`
- **Массив, ФиксированныйМассив** – сериализуется в массив JSON в том случае, если любой элемент массива может быть сериализован в JSON.
- **Структура, ФиксированнаяСтруктура** – сериализуется в объект JSON:
 - Ключ – ключ элемента структуры.
 - Значение – значение элемента структуры в том случае, если значение может быть сериализовано в JSON.
- **Соответствие, ФиксированноеСоответствие** – сериализуется в объект JSON:
 - Ключ – ключ элемента соответствия. Ключ может быть только значением типа **Строка**, в противном случае будет генерироваться исключение.
 - Значение – значение элемента соответствия в том случае, если значение может быть сериализовано в JSON.
- **Дата** – формат сериализации определяется настройками.
- Если выполняется попытка сериализации типа, отсутствующего в данном списке – будет вызвано исключение.

При работе с объектной техникой, имеется возможность читать (и писать) данные в соответствие или структуру. Основное отличие между этими объектами состоит в том, что ключ элемента структуры подчиняется правилам формирования переменной на встроенном языке, а ключ элемента соответствия может быть любым. С учетом того, чтоб JSON не накладывает ограничений на значение ключа, не все JSON-документы можно прочитать в структуру. Еще одним различием между структурой и соответствием является то, что к элементам структуры можно обращаться «через точку», а к элементам соответствия такой доступ не предоставляется. В связи с этим, может оказаться удобным получать данные в виде структуры, если ключи из JSON-документа соответствуют требованиям к ключам структур системы «1С:Предприятие».

Объектная техника предполагает достаточно простую работу с данными, однако платой за это является большой расход памяти, т. к. весь JSON-документ обрабатывается целиком в оперативной памяти.

17.2.2. Запись

Для того чтобы выполнить запись объекта в формате JSON, необходимо использовать (в простейшем случае) следующие объекты:

1. Собственно записываемый объект, например типа **Структура**.
2. Объект, обеспечивающий низкоуровневую запись данных в формате JSON – **ЗаписьJSON**.
3. Объект настроек сериализации **НастройкиСериализацииJSON**.

Метод глобального контекста `ЗаписатьJSON()` оперирует вышеперечисленными объектами. Рассмотрим пример, в котором потребуется записать структуру, которая состоит из трех элементов разного типа (но типы являются примитивными):

[Копировать в буфер обмена](#)

```
Запись = Новый ЗаписьJSON;  
Запись.ОткрытьФайл("c:\temp\simpleWrite.json");  
Данные = Новый Структура;  
Данные.Вставить("ДлинаЗаписи", 20);  
Данные.Вставить("КлючЗаписи", "abcdefgh");  
Данные.Вставить("ДатаИзменения", ТекущаяДата());  
ЗаписатьJSON(Запись, Данные, Новый НастройкиСериализацииJSON);  
Запись.Закрыть();
```

В результате работы данный пример сформирует следующий JSON-документ:

[Копировать в буфер обмена](#)

```
{  
  "ДлинаЗаписи": 20,  
  "КлючЗаписи": "abcdefgh",  
  "ДатаИзменения": "2014-09-24T17:32:11"  
}
```

Если в сериализуемых данных участвуют значения типа `Дата`, то может потребоваться управлять форматом сериализации. Для этого необходимо установить параметры объекта `НастройкиСериализацииJSON`. В следующем примере дата будет сериализоваться в формате `JavaScript`:

[Копировать в буфер обмена](#)

```
НастройкиСериализации = Новый НастройкиСериализацииJSON;
НастройкиСериализации.ФорматСериализацииДаты = ФорматДатыJSON.JavaScript;
Запись = Новый ЗаписьJSON;
Запись.ОткрытьФайл("c:\temp\simpleWrite.json");
Данные = Новый Структура;
Данные.Вставить("ДлинаЗаписи", 20);
Данные.Вставить("КлючЗаписи", "abcdefgh");
Данные.Вставить("ДатаИзменения", ТекущаяДата());
ЗаписатьJSON(Запись, Данные, НастройкиСериализации);
Запись.Закрыть();
```

В результате работы данный пример сформирует следующий JSON-документ:

[Копировать в буфер обмена](#)

```
{
  "ДлинаЗаписи": 20,
  "КлючЗаписи": "abcdefgh",
  "ДатаИзменения": "new Date(1411565832000)"
}
```

Если несколько расширить набор записываемых данных, например, добавить к ним значение типа `УникальныйИдентификатор`, то запись не будет выполнена. В результате исполнения кода:

[Копировать в буфер обмена](#)

```
Запись = Новый ЗаписьJSON;
Запись.ОткрытьФайл("c:\temp\simpleWrite.json");
Данные = Новый Структура;
Данные.Вставить("ДлинаЗаписи", 20);
Данные.Вставить("КлючЗаписи", "abcdefgh");
Данные.Вставить("ДатаИзменения", ТекущаяДата());
Данные.Вставить("UID", Новый УникальныйИдентификатор);
ЗаписатьJSON(Запись, Данные, Новый НастройкиСериализацииJSON);
Запись.Закрыть();
```

Будет вызвано исключение:

[Копировать в буфер обмена](#)

```
Ошибка при вызове метода контекста (ЗаписатьJSON)
  ЗаписатьJSON(Запись, Данные, НастройкиСериализации);
по причине:
Передано значение недопустимого типа
```

Причиной такого поведения является то, что тип `УникальныйИдентификатор` не входит в состав JSON-сериализуемых типов данных системы «1С:Предприятие». Однако система предоставляет возможность «обойти» это ограничение: необходимо передать в функцию `ЗаписатьJSON()` имя функции обратного вызова, которая будет заниматься JSON-сериализацией неподдерживаемых объектов. Эта функция будет называться **функцией преобразования**. При этом формат такой сериализации будет разрабатывать непосредственно сам прикладной разработчик. Надо понимать, что такая сериализация не будет универсальной, т. к. принимающая сторона, не обладающая знаниями о формате сериализации, не сможет прочесть переданные данные. Другими словами, формат сериализации необходимо разрабатывать совместно всеми сторонами обмена такого рода данными.

С учетом вышесказанного, более сложный вариант обмена теперь происходит следующим образом:

- Вызывается функция сериализации объекта в формат JSON (`ЗаписатьJSON()`).
- Система «1С:Предприятие» для каждого элемента структуры, тип значения которого не сериализуется в формат JSON, будет вызываться функция преобразования.
- Функция преобразования анализирует переданный объект и принимает решение – отказаться от его записи или вернуть платформе значение, которое может быть сериализовано в JSON.

Доработанный код записи будет выглядеть следующим образом:

[Копировать в буфер обмена](#)

```
&НаСервере
Процедура ЗаписьНаСервере()
  Запись = Новый ЗаписьJSON;
  Запись.ОткрытьФайл("c:\temp\compositeWrite.json");
  Данные = Новый Структура;
  Данные.Вставить("ДлинаЗаписи", 20);
  Данные.Вставить("КлючЗаписи", "abcdefgh");
  Данные.Вставить("ДатаИзменения", ТекущаяДата());
  Данные.Вставить("UID", Новый УникальныйИдентификатор);
  ЗаписатьJSON(Запись, Данные, Новый НастройкиСериализацииJSON, "ФункцияПреобразованияЗаписи", ЭтотОбъект);
  Запись.Закрыть();
КонецПроцедуры
&НаСервере
Функция ФункцияПреобразованияЗаписи(Свойство, Значение, ДополнительныеПараметры, Отказ) Экспорт
  Если ТипЗнч(Значение) = Тип("УникальныйИдентификатор") Тогда
    Возврат Строка(Значение);
  КонецЕсли;
  Отказ = Истина;
КонецФункции
```

Следует обратить внимание, что функция преобразования должна быть объявлена с указанием ключевого слова **Экспорт**. Также следует помнить, что функция преобразования (в модуле управляемой формы) может быть описана только в «контекстной» части модуля, т. е. с использованием директивы компиляции **&НаКлиенте** или **&НаСервере**.

В результате работы приведенного примера будет сформирован следующий JSON-документ:

[Копировать в буфер обмена](#)

```
{
  "ДлинаЗаписи": 20,
  "КлючЗаписи": "abcdefgh",
  "ДатаИзменения": "2014-09-24T18:09:13",
  "UID": "5a80e5dc-252b-416f-b487-d9ddcebc523e"
}
```

При создании функции преобразования следует помнить о следующих особенностях:

- Функция преобразования вызывается для значений всех типов, которые не поддерживают сериализацию в JSON (см. [здесь](#)).
- Если функция преобразования вернет значение, которое не может быть сериализовано в JSON – будет вызвано исключение.
- Если переданное значение является структурой, то вызов функции преобразования для элементов этой структуры будет вызваться до того, как в поток будет записано имя свойства этого элемента. В результате функция преобразования может отказаться от записи значения и структура формируемого JSON-документа не будет нарушена.
- Имя свойства будет передано в функцию преобразования только для элементов структур и соответствий.

17.2.3. Чтение

Чтение данных в объектной технике выглядит аналогично записи. Рассмотрим пример чтения файлов, которые формировались во время рассмотрения объектной записи (предыдущий раздел).

[Копировать в буфер обмена](#)

```
Чтение = Новый ЧтениеJSON;
Чтение.ОткрытьФайл("c:\temp\simpleWrite.json");
Данные = ПрочитатьJSON(Чтение, Ложь);
Чтение.Закрыть();
```

Документ имеет следующее содержание (с точностью до значения даты):

[Копировать в буфер обмена](#)

```
{
  "ДлинаЗаписи": 20,
  "КлючЗаписи": "abcdefgh",
  "ДатаИзменения": "new Date(1411565832000)"
}
```

При чтении JSON-документа в переменную **Данные** будет сформирована структура вида:

[Копировать в буфер обмена](#)

```
Ключ = ДлинаЗаписи, значение = 20
Ключ = КлючЗаписи, значение = abcdefgh
Ключ = ДатаИзменения, значение = <значение даты и времени>
```

Такой вариант чтения хорошо подходит в том случае, если читаемые данные могут быть преобразованы в структуру или соответствие и все читаемые данные могут быть однозначно десериализованы без потери информации о типе. Если читаемые данные обладают сложной структурой или требуют выполнения дополнительных преобразований при чтении, то можно пойти двумя путями:

1. Получить соответствие (или структуру), в которое будет полностью загружен JSON-документ, и потом завершить преобразование с помощью обхода получившегося объекта.
2. Заниматься необходимым преобразованием непосредственно во время загрузки данных. Для этого необходимо передать в функцию **ПрочитатьJSON()** имя функции обратного вызова, которая будет заниматься десериализацией JSON-данных в нужные объекты системы «1С:Предприятие». Эта функция будет называться **функцией восстановления**.

В обоих случаях прикладной разработчик должен знать, какие данные и в каком виде находятся в JSON-документе. Далее будут подробно рассмотрены оба варианта загрузки данных. В качестве макетной задачи будет рассматриваться получение погоды в г. Москва с помощью некоторого интернет-сервиса. В качестве ответа интернет-сервис возвращает JSON-документ следующего содержания:

[Копировать в буфер обмена](#)

```
{
  "id":524901,"name":"Moscow","dt":1411464940000,
  "coord": { "lon":37.62,"lat":55.75 },
  "sys": { "country":"RU","sunrise":1411442400000,"sunset":1411485660000 },
  "weather": [
    { "main":"Clouds","description":"пасмурно" }
  ],
  "main": { "temp":282.93,"pressure":1014,"humidity":93 },
  "wind": { "speed":4,"deg":350 },
  "clouds": { "all":90 }
}
```

Кратко рассмотрим описание формата:

- **id** – идентификатор города;
- **name** – имя города;

- `dt` – дата и время получения погоды, в формате Unix, GMT;
- `coord` – местоположение города:
 - `lon` – долгота;
 - `lan` – широта.
- `sys` – дополнительная информация:
 - `country` – страна расположения города;
 - `sunrise` – время восхода Солнца в формате Unix, GMT;
 - `sunset` – время заката Солнца, в формате Unix, GMT.
- `weather` – дополнительная информация о погоде:
 - `main` – общая характеристика погоды;
 - `description` – описание погоды.
- `main` – собственно описание погоды:
 - `temp` – температура, в градусах Кельвина. Для получения градусов Цельсия необходимо вычесть 273.15;
 - `pressure` – давление в гектопаскалях. Для перевода в миллиметры ртутного столба, надо значение давления умножить на 0,75.
 - `humidity` – влажность в %.
- `wind` – параметры ветра:
 - `speed` – скорость в милях в час. Для перевода в километры в час необходимо умножить на 1,61.
 - `deg` – направление ветра, в градусах.
- `clouds` – информация об осадках:
 - `all` – вероятность возникновения осадков, в %.

В результате загрузки этих данных должна получиться структура, где все времена представлены стандартным типом `Дата`, температура – в градусах Цельсия, скорость – в километрах в час, а давление – в миллиметрах ртутного столба.

Рассмотрим загрузку данной информации обоими способами. Данные записаны в файле `c:\temp\weather.json`.

17.2.3.1. Чтение с постобработкой

Собственно процесс чтения выглядит просто:

[Копировать в буфер обмена](#)

```
Чтение = Новый ЧтениеJSON;
Чтение.ОткрытьФайл("c:\temp\weather.json");
Данные = ПрочитатьJSON(Чтение, Ложь);
Чтение.Закрыть();
```

В результате в переменной `Данные` будет следующая информация:

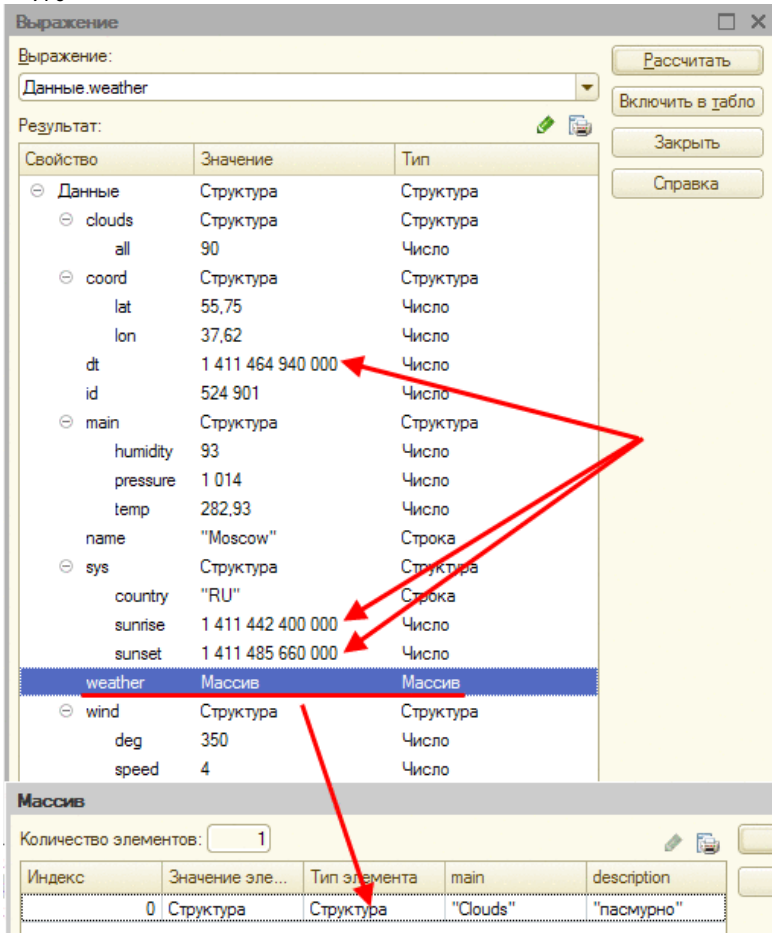


Рис. 333. Результат загрузки

Без учета необходимости конвертации все выглядит предсказуемо. Однако дата и время автоматически не преобразовались. Можно попробовать указать системе на то, что поле `dt` (например) является полем, где находится дата и время:

[Копировать в буфер обмена](#)

```
Чтение = Новый ЧтениеJSON;
Чтение.ОткрытьФайл("c:\temp\weather.json");
Данные = ПрочитатьJSON(Чтение, Ложь, "dt", ФорматДатыJSON.JavaScript);
Чтение.Закрыть();
```

В результате получим сообщение об ошибке следующего вида:

[Копировать в буфер обмена](#)

```
Ошибка при вызове метода контекста (ПрочитатьJSON)
Данные = ПрочитатьJSON(Чтение, Ложь, "dt", ФорматДатыJSON.JavaScript);
по причине:
Представление даты имеет неверный формат
```

Ошибка произошла потому, что система не понимает формат даты и времени, выраженной простым числом. Программный интерфейс работы с JSON предлагает функцию, помогающую выполнить конвертацию полей типа *Дата* – `ПрочитатьДатуJSON()`. Для использования этой функции необходимо привести десериализованную строку к формату, принятому, например, в JavaScript. В рассматриваемом примере получится следующий программный текст:

[Копировать в буфер обмена](#)

```
Данные.dt = ПрочитатьДатуJSON("new Date("+Формат(Данные.dt, "ЧГ="+")+")", ФорматДатыJSON.JavaScript);
```

В результате значение свойства `Данные.dt` станет равно значению `23.09.2014 13:35:40` (типа *Дата*). Остальная конвертация выполняется аналогичным образом.

17.2.3.2. Чтение с функцией восстановления

Вариант постобработкой выглядит не очень хорошо, если чтение JSON-документа предполагается выполнять в разных местах прикладного решения. В этом случае может возникнуть ситуация, когда код преобразования будет расположен в нескольких местах прикладного решения.

Хорошей заменой постобработки, в этом случае, является использование функции восстановления. Для применения функции восстановления необходимо описать саму функцию и несколько изменить само чтение документа:

[Копировать в буфер обмена](#)

```
&НаСервере
Функция ФункцияВосстановленияЧтения(Свойство, Значение, ДополнительныеПараметры) Экспорт
Если Свойство = "dt" ИЛИ Свойство = "sunrise" ИЛИ Свойство = "sunset" Тогда
    Возврат ПрочитатьДатуJSON("new Date("+Формат(Значение, "ЧГ="+")+")", ФорматДатыJSON.JavaScript);
КонецЕсли;
Если Свойство = "pressure" Тогда
    Возврат Значение*0.75;
КонецЕсли;
```

```

Если Свойство = "temp" Тогда
    Возврат Значение-273.15;
КонецЕсли;
Если Свойство = "speed" Тогда
    Возврат Значение*1.61;
КонецЕсли;
КонецФункции
&НаСервере
Процедура ЧтениеНаСервере()
    Чтение = Новый ЧтениеJSON;
    Чтение.ОткрытьФайл("c:\temp\weather.json");
    Данные = ПрочитатьJSON(Чтение, Ложь, , , "ФункцияВосстановленияЧтения", ЭтотОбъект);
    Чтение.Закрыть();
КонецПроцедуры

```

В вышеуказанном примере присутствует особенность – функция восстановления будет вызвана для всех свойств, которые будут обнаружены в JSON-документе. Это не всегда удобно и, кроме того, существенно снижает производительность чтения JSON-документа (за счет вызова функции восстановления). Например, при чтении метеорологических данных нам необходимо выполнять особое преобразование только для свойств, в которых содержится дата и время, а остальные свойства мы конвертировать не собираемся. Чтобы не анализировать в функции восстановления имя реквизита (как в вышеприведенном примере), можно поступить другим способом: явным образом передать в функцию чтения JSON-документа список реквизитов, которые содержат дату и время, а функцию восстановления написать исходя из того, что эта функция будет вызвана только для необходимых свойств. В результате получится следующий код:

[Копировать в буфер обмена](#)

```

&НаСервере
Процедура ЧтениеНаСервере()
    Чтение = Новый ЧтениеJSON;
    Чтение.ОткрытьФайл("c:\temp\weather.json");
    Реквизиты = Новый Массив;
    Реквизиты.Добавить("dt");
    Реквизиты.Добавить("sunrise");
    Реквизиты.Добавить("sunset");
    Данные = ПрочитатьJSON(Чтение, Ложь, , , "ФункцияВосстановленияЧтения", ЭтотОбъект, , Реквизиты);
    Чтение.Закрыть();
КонецПроцедуры
&НаСервере
Функция ФункцияВосстановленияЧтения(Свойство, Значение, ДополнительныеПараметры) Экспорт
    Возврат ПрочитатьДатуJSON("new Date("+Формат(Значение, "ЧГ="+")", , ФорматДатыJSON.JavaScript);
КонецФункции

```

Следует обратить внимание, что функция восстановления должна быть объявлена с указанием ключевого слова **Экспорт**. Также следует помнить, что функция восстановления (в модуле управляемой формы) может быть описана только в «контекстной» части модуля, т. е. с использованием директивы компиляции **&НаКлиенте** или **&НаСервере**. При разработке функции восстановления необходимо принимать во внимание тот факт, что свойства документа считываются не в том порядке, как они представлены в файле.

Рассмотрим последовательность, в которой свойства JSON-документа попадают в функцию восстановления. Для этого разместим в таблице каждое свойство файла и то, в каком порядке будет прочитано свойство:

Данные JSON-документа	Последовательность вызова функции восстановления
{ "id":524901,	1
"name": "Moscow",	2
"dt": 1411464940000,	3
"coord":	6
{ "lon": 37.62,	4
"lat": 55.75 },	5
"sys":	9
{ "country": "RU",	6
"sunrise": 1411442400000,	7
"sunset": 1411485660000 },	8
"weather":	12
[11
{ "main": "Clouds",	9
"description": "пасмурно" }	10
],	
"main":	16
{ "temp": 282.93,	13
"pressure": 1014,	14
"humidity": 93 },	15
"wind":	19
{ "speed": 4,	17

"deg": 350 },	18
"clouds":	20
{ "all": 90 }	21
	22

В общем случае, можно сформулировать следующее правило обхода: первым будет прочитано свойство, которому не подчинено ни одно другое свойство. Например, свойству `id` не подчинено никакое свойство, и оно считывается первым. Однако свойству `coord` подчинено свойства `lon` и `lat`, поэтому вначале будут считаны эти свойства, а лишь затем – свойство `coord`, которое в качестве значения получит структуру (или соответствие) из подчиненных свойств документа.

17.3. Потоквая техника работы

17.3.1. Общая информация

Потоковая техника работы с документом ориентирована на то, что документ целиком не загружается в память и обрабатывается последовательно, от элемента к элементу. Например, если надо прочитать только какой-то объект из JSON-документа, то потоковая техника может дать существенный выигрыш, особенно в том случае, если требуемый элемент находится в начале обрабатываемого документа.

17.3.2. Запись

Для того чтобы выполнить потоковую запись JSON-документа, необходимы записываемые данные и объект `ЗаписьJSON`. При этом следует понимать, что формирование корректной структуры JSON-документа полностью лежит на прикладном разработчике, который формирует документ. Для упрощения такого контроля объект `ЗаписьJSON` имеет свойство `ПроверятьСтруктуру`.

Рассмотрим простой пример записи документа:

[Копировать в буфер обмена](#)

```

ПараметрыJSON = Новый ПараметрыЗаписиJSON(ПереносСтрокJSON.Авто, " ", Истина);
Запись = Новый ЗаписьJSON;
Запись.ПроверятьСтруктуру = Истина;
Запись.ОткрытьФайл("c:\temp\streamWrite.json", , , ПараметрыJSON);
Запись.ЗаписатьНачалоОбъекта();
Запись.ЗаписатьИмяСвойства("ДлинаЗаписи");
Запись.ЗаписатьЗначение(20);
Запись.ЗаписатьИмяСвойства("Товар");
Запись.ЗаписатьНачалоОбъекта();
Запись.ЗаписатьИмяСвойства("Код");
Запись.ЗаписатьЗначение("0020");
Запись.ЗаписатьИмяСвойства("Наименование");
Запись.ЗаписатьЗначение("Товар");
Запись.ЗаписатьКонецОбъекта();
Запись.ЗаписатьИмяСвойства("МассивЧисел");
Запись.ЗаписатьНачалоМассива();
Запись.ЗаписатьЗначение(3.141592654, Истина);
Запись.ЗаписатьЗначение(2.718281828, Ложь);
Запись.ЗаписатьКонецМассива();
Запись.ЗаписатьКонецОбъекта();
Запись.Закрыть();

```

В результате исполнения этого программно кода будет сформирован следующий документ:

[Копировать в буфер обмена](#)

```

{
  "ДлинаЗаписи": 20,
  "Товар": {
    "Код": "0020",
    "Наименование": "Товар"
  },
  "МассивЧисел": [
    3.141592654E0,
    2.718281828
  ]
}

```

Такой формат документа удобен для визуального просмотра, но занимает больше места. Можно изменить значение первого параметра конструктора `ПараметрыЗаписиJSON` на значение `ПереносСтрокJSON.Нет` и результирующий документ примет такой вид (разница составит примерно 20%):

[Копировать в буфер обмена](#)

```

{"ДлинаЗаписи":20,"Товар":{"Код":"0020","Наименование":"Товар"},"МассивЧисел":[3.141592654E0,2.718281828]}

```

Если необходимо получить JSON-документ без формирования файла, то можно использовать метод `УстановитьСтроку()` объекта `ЗаписьJSON`. После вызова этого метода, для получения строки со сформированным JSON-документом, достаточно просто завершить запись документа методом `Закрыть()` объекта `ЗаписьJSON`:

[Копировать в буфер обмена](#)

```

ПараметрыJSON = Новый ПараметрыЗаписиJSON(ПереносСтрокJSON.Нет, " ", Истина);
Запись = Новый ЗаписьJSON;
Запись.УстановитьСтроку();
...
Документ = Запись.Закрыть();

```

Теперь данный документ (из переменной `Документ`) можно передавать, например, в тело HTTP-запроса.

17.3.3. Чтение

Потоковое чтение JSON-документа выполняется аналогично его записи: прикладной разработчик в цикле читает следующий элемент, определяет, что считано и обрабатывает считываемые данные.

Примитивный случай потокового чтения документа может выглядеть следующим образом:

[Копировать в буфер обмена](#)

```
Чтение = Новый ЧтениеJSON;
Чтение.ОткрытьФайл("c:\temp\streamWrite.json");
Пока Чтение.Прочитать() Цикл
    Сообщить("Тип текущего элемента " + Чтение.ТипТекущегоЗначения);
    Если Чтение.ТипТекущегоЗначения = ТипЗначенияJSON.ИмяСвойства Тогда
        Сообщить("Имя = " + Чтение.ТекущееЗначение);
    КонецЕсли;
    Если Чтение.ТипТекущегоЗначения = ТипЗначенияJSON.Булево Или
        Чтение.ТипТекущегоЗначения = ТипЗначенияJSON.Строка Или
        Чтение.ТипТекущегоЗначения = ТипЗначенияJSON.Число Или
        Чтение.ТипТекущегоЗначения = ТипЗначенияJSON.Комментарий Тогда
        Сообщить("Значение = " + Чтение.ТекущееЗначение);
    КонецЕсли;
КонецЦикла;
Чтение.Закрыть();
```

Тогда при чтении документа, сформированного при рассмотрении потоковой записи JSON-документа (см. [здесь](#)), будет сформирован следующий результат:

[Копировать в буфер обмена](#)

```
Тип текущего элемента Начало объекта
Тип текущего элемента Имя свойства
Имя = ДлинаЗаписи
Тип текущего элемента Число
Значение = 20
Тип текущего элемента Имя свойства
Имя = Товар
Тип текущего элемента Начало объекта
Тип текущего элемента Имя свойства
Имя = Код
Тип текущего элемента Строка
Значение = 0020
Тип текущего элемента Имя свойства
Имя = Наименование
Тип текущего элемента Строка
Значение = Товар
Тип текущего элемента Конец объекта
Тип текущего элемента Имя свойства
Имя = МассивЧисел
Тип текущего элемента Начало массива
Тип текущего элемента Число
Значение = 3,141592654
Тип текущего элемента Число
Значение = 2,718281828
Тип текущего элемента Конец массива
Тип текущего элемента Конец объекта
```

17.4. Совмещение техник

Для упрощения работы с JSON, можно совмещать различные техники при формировании одного документа. Например, необходимо сформировать документ, который содержит в себе некоторый набор структур и массив. В этом случае можно все оформление документа выполнять с помощью потоковой техники, а уже готовые структуры и массив записывать с помощью объектной техники. Важно только обеспечить корректную структуру документа перед началом объектной записи.

Приведем пример совмещения техник на следующем примере:

- В ответ на запрос внешней системы, этой системе необходимо возвращать список заказов.
- Список должен содержать дату формирования и набор заказов (соответствующий некоторому критерию).
- Каждый заказ описывается следующими параметрами:
 1. Номер заказа;
 2. Дата формирования заказа;
 3. Контрагент по заказу;
 4. Уникальный идентификатор заказа, по которому впоследствии можно получить всю информацию по заказу.

Фрагмент кода на встроенном языке, который формирует JSON-документ, будет иметь следующий вид:

[Копировать в буфер обмена](#)

```
НастройкиСериализации = Новый НастройкиСериализацииJSON;
НастройкиСериализации.ВариантЗаписиДаты = ВариантЗаписиДатыJSON.УниверсальнаяДата;
НастройкиСериализации.ФорматСериализацииДаты = ФорматДатыJSON.ISO;
НастройкиСериализации.СериализоватьМассивыКакОбъекты = Истина;
```

```

ПараметрыJSON = Новый ПараметрыЗаписиJSON(ПереносСтрокJSON.Авто, " ", Истина);
Запись = Новый ЗаписьJSON;
Запись.ПроверитьСтруктуру = Истина;
Запись.ОткрытьФайл("c:\temp\combinedWrite.json", , , ПараметрыJSON);
Запись.ЗаписатьНачалоОбъекта();
Запись.ЗаписатьИмяСвойства("ДатаФормирования");
Запись.ЗаписатьЗначение(ЗаписатьДатуJSON(КогдаСформировано, ФорматДатыJSON.ISO, ВариантЗаписиДатыJSON.УниверсальнаяДата));
Запись.ЗаписатьИмяСвойства("Заказы");
Заказы = Новый Массив;
Для каждого Заказ Из СписокЗаказов Цикл
    ОписаниеЗаказа = Новый Структура("Ссылка, Номер, Дата, Контрагент");
    ОписаниеЗаказа.Ссылка = Строка(Заказ.Ссылка);
    ОписаниеЗаказа.Номер = Заказ.Номер;
    ОписаниеЗаказа.Дата = Заказ.Дата;
    ОписаниеЗаказа.Контрагент = Строка(Заказ.Контрагент);
    Заказы.Добавить(ОписаниеЗаказа);
КонецЦикла;
ЗаписатьJSON(Запись, Заказы, НастройкиСериализации);
Запись.ЗаписатьКонецОбъекта();
Запись.Закреть();

```

При исполнении данного кода предполагается, что:

- Переменная `КогдаСформировано` содержит значение типа `Дата`. Содержит дату и время формирования JSON-документа.
- Переменная `СписокЗаказов` является массивом ссылок на документы заказов.

Исполнение данного код приведет к формированию следующего JSON-документа:

[Копировать в буфер обмена](#)

```

{
  "ДатаФормирования": "2014-10-06T12:57:35Z",
  "Заказы": {
    "0": {
      "Ссылка": "f4d1495a-02b5-4d56-92c6-840c11dfb592",
      "Номер": 234,
      "Дата": "2014-09-30T20:00:00Z",
      "Контрагент": "Иванов И.И."
    },
    "1": {
      "Ссылка": "ee821799-2d57-475e-a330-f414e53b8bda",
      "Номер": 436,
      "Дата": "2014-09-24T20:00:00Z",
      "Контрагент": "Петров А.П."
    },
    "2": {
      "Ссылка": "e058a5a8-3c0d-453b-8b1c-963a35fe2b7a",
      "Номер": 118,
      "Дата": "2014-08-31T20:00:00Z",
      "Контрагент": "Иванов И.И."
    }
  }
}

```

Изменяя значения параметров объектов `НастройкиСериализации` и `ПараметрыJSON`, а также манипулируя параметрами метода `ЗаписатьДатуJSON()`, можно изменять результирующий JSON-документ для максимального соответствия «ожиданиям» принимающей системы.

Глава 18. Механизмы интернет-сервисов

18.1. Web-сервисы

18.1.1. Общая информация

Механизм Web-сервисов в системе «1С:Предприятие» является средством поддержки сервисно-ориентированной архитектуры (**Service-Oriented Architecture, SOA**).

Сервисно-ориентированная архитектура представляет собой прикладную архитектуру, в которой все функции определены как независимые сервисы с вызываемыми интерфейсами. Обращение к этим сервисам в определенной последовательности позволяет реализовать тот или иной бизнес-процесс.

Сервисно-ориентированная архитектура предлагает новый подход к созданию распределенных информационных систем, в которых программные ресурсы рассматриваются как сервисы, предоставляемые по сети. Такой подход позволяет обеспечить быструю консолидацию распределенных компонентов (сервисов) в единое решение для поддержки определенных бизнес-процессов.

Механизм Web-сервисов позволяет использовать систему «1С:Предприятие» как набор сервисов в сложных распределенных и гетерогенных системах, а также позволяет интегрировать ее с другими промышленными системами с использованием сервисно-ориентированной архитектуры.

Конфигурация системы «1С:Предприятие» может экспортировать свою функциональность через Web-сервисы. Определения Web-сервисов задаются в дереве конфигурации и становятся доступны произвольным информационным системам благодаря публикации их на веб-сервере.

Кроме этого, система «1С:Предприятие» может обращаться к Web-сервисам сторонних производителей как через статические ссылки, определенные в дереве конфигурации, так и с помощью динамических ссылок, создаваемых средствами встроенного языка.

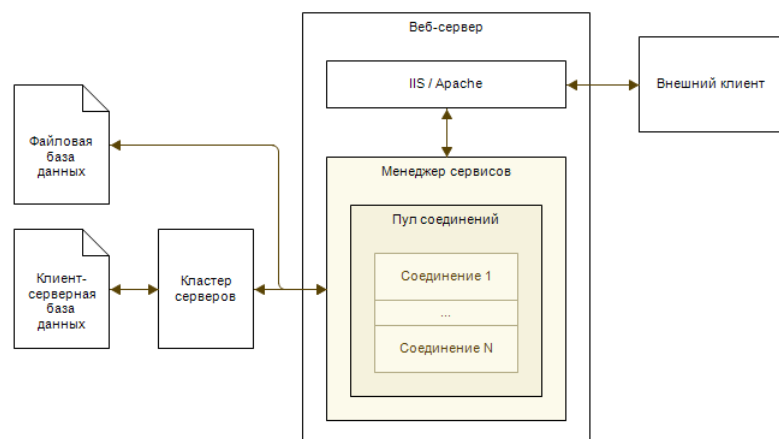


Рис. 334. Web-сервисы

В основе сервисной архитектуры системы «1С:Предприятие 8» находится менеджер сервисов. Менеджер сервисов выполняет следующие функции:

- управление пулом соединений с информационными базами;
- поддержка WSDL описания сервиса;
- реализация протокола SOAP, сериализация сообщений, вызов соответствующего сервиса.

Менеджер сервисов выполняется в процессе сервисного хоста, который выполняет функцию приема/передачи сообщений из/в менеджер сервисов. В качестве сервисного хоста может использоваться веб-сервер IIS или Apache.

Менеджер сервисов содержит в себе пул соединений, через которые идет взаимодействие с базами данных системы «1С:Предприятие».

Механизм Web-сервисов, реализованный в системе «1С:Предприятие», поддерживает следующие стандарты:

- SOAP 1.1,
- SOAP 1.2,
- WSDL 1.1,
- WS-I Basic Profile 1.1,
- HTTP 1.1,
- SSL 3.0/TLS 1.0.

Механизм Web-сервисов, реализованный в системе «1С:Предприятие», не поддерживает:

- Механизм WS-Policy.

18.1.2. Предоставление функциональности через Web-сервисы

Для того чтобы функциональность системы «1С:Предприятие» могла быть доступна внешним потребителям Web-сервисов, нужно выполнить следующие действия:

- создать в конфигурации необходимое количество Web-сервисов,
- опубликовать Web-сервисы с помощью специального инструмента конфигулятора.

Публикация Web-сервисов подробно описана в книге «1С:Предприятие 8.3. Руководство администратора».

Создание Web-сервиса заключается:

- в добавлении в дерево метаданных объекта конфигурации Web-сервис,
- описании операций, которые может выполнять данный Web-сервис,
- описании параметров операций.

Объект конфигурации Web-сервис содержит модуль, в котором создаются процедуры на встроенном языке, выполняемые при вызове тех или иных операций Web-сервиса. Типы параметров операций Web-сервиса описываются с помощью типов XDTO и могут представлять собой либо значения XDTO, либо объекты XDTO.

Вызов Web-сервиса происходит следующим образом:

• из пула соединений выбирается подходящее соединение с информационной базой; при отсутствии необходимого соединения соединение создается;

• создается новый сеанс и для созданного сеанса вызывается событие `УстановкаПараметровСеанса` (в модуле сеанса);

• выполняется вызов затребованного метода Web-сервиса, при этом происходит вызов обработчика `УстановкаПараметровСеанса()` (в модуле сеанса) каждый раз, когда происходит обращение к неинициализированному параметру сеанса.

СОВЕТ. Не рекомендуется выполнять ресурсоемкие операции в обработке события `УстановкаПараметровСеанса`.

Событие `УстановкаПараметровСеанса` модуля сеанса вызывается на сервере в привилегированном режиме. Модуль вызванного сервиса исполняется на сервере в обычном режиме.

Модуль сеанса (см. [здесь](#)) служит для инициализации параметров сеанса и выполнения некоторого набора команд при вызове любого Web-сервиса системы «1С:Предприятие».

18.1.3. Пример реализации Web-сервиса

Например, требуется создать Web-сервис системы «1С:Предприятие», который должен по переданному номеру расходной накладной возвращать состав ее табличной части. Аналогичный пример с помощью HTTP-сервисов см. [здесь](#).

Для описания возвращаемого значения создадим пакет XDTO `ДанныеРасходнойНакладной` с пространством имен <http://www.MyCompany.ru/shipment>, содержащий три типа объектов XDTO:

- **Номенклатура** – для передачи данных элемента справочника `Номенклатура`. Этот тип объекта XDTO будет содержать следующие свойства:
 - **Наименование** – тип `string` из пространства имен <http://www.w3.org/2001/XMLSchema>;
 - **ПолноеНаименование** – тип `string` из пространства имен <http://www.w3.org/2001/XMLSchema>;
 - **ШтрихКод** – тип `string` из пространства имен <http://www.w3.org/2001/XMLSchema>;
 - **ЗакупочнаяЦена** – тип `int` из пространства имен <http://www.w3.org/2001/XMLSchema>.
- **СтрокаРасходнойНакладной** – для передачи данных одной строки расходной накладной. Этот тип объекта XDTO будет содержать следующие свойства:
 - **Номенклатура** – тип `Номенклатура` из пространства имен <http://www.MyCompany.ru/shipment>; представляет собой ссылку на объект XDTO, который мы определили выше;
 - **Количество** – тип `int` из пространства имен <http://www.w3.org/2001/XMLSchema>;
 - **Цена** – тип `int` из пространства имен <http://www.w3.org/2001/XMLSchema>;
 - **Сумма** – тип `int` из пространства имен <http://www.w3.org/2001/XMLSchema>.
- **РасходнаяНакладная** – для передачи данных всех строк расходной накладной. Этот тип объекта XDTO будет содержать единственное свойство:
 - **Состав** – тип `СтрокаРасходнойНакладной` из пространства имен <http://www.MyCompany.ru/shipment>. Представляет собой ссылку на объект XDTO, который мы определили выше. Для того чтобы это свойство могло содержать неограниченное множество значений, необходимо установить его свойство `Верхняя граница` в значение `-1`.

После того как необходимые типы XDTO созданы, следует добавить в конфигурацию новый Web-сервис `ДанныеРасходнойНакладной` со следующими значениями свойств:

- **URI Пространства имен** – <http://www.MyCompany.ru/shipment>;
- **Пакеты XDTO** – `ДанныеРасходнойНакладной`;
- **Имя файла публикации** – `shipment.lcws`.

У созданного Web-сервиса следует определить операцию `Получить` со следующими значениями свойств:

- **Тип возвращаемого значения** – `РасходнаяНакладная` из пространства имен <http://www.MyCompany.ru/shipment>;
- **Возможно пустое** – `установлен`;
- **Имя процедуры** – `Получить`.

У операции `Получить` следует определить параметр `НомерДокумента` со следующими значениями свойств:

- **Тип значения** – тип `string` из пространства имен <http://www.w3.org/2001/XMLSchema>;
- **Направление передачи** – `входной`.

После этого следует открыть модуль созданного Web-сервиса и поместить в него функцию `Получить()`, которая будет выполняться при вызове данного Web-сервиса.

[Копировать в буфер обмена](#)

```

Функция Получить(НомерДокумента) Экспорт
// Получить объект расходной накладной по переданному номеру
ДокументСсылка = Документы.РасходнаяНакладная.НайтиПоНомеру(НомерДокумента, ТекущаяДата());
Если ДокументСсылка.Пустая() Тогда
    Возврат Неопределено;
КонецЕсли;
Документ = ДокументСсылка.ПолучитьОбъект();
// Получить типы объектов XDTO
НоменклатураТип = ФабрикаXDTO.Тип("http://www.MyCompany.ru/shipment", "Номенклатура");
РасходнаяНакладнаяТип = ФабрикаXDTO.Тип("http://www.MyCompany.ru/shipment", "РасходнаяНакладная");
СтрокаРасходнойНакладнойТип = ФабрикаXDTO.Тип("http://www.MyCompany.ru/shipment", "СтрокаРасходнойНакладной");
// Создать объект XDTO расходной накладной
РасходнаяНакладная = ФабрикаXDTO.Создать(РасходнаяНакладнаяТип);
Для Каждого СтрокаДокумента Из Документ.Состав Цикл
    // Создать объекты XDTO строки расходной накладной
    // и номенклатуры
    СтрокаРасходнойНакладной = ФабрикаXDTO.Создать(СтрокаРасходнойНакладнойТип);
    Номенклатура = ФабрикаXDTO.Создать(НоменклатураТип);
    // Заполнить свойства номенклатуры
    Номенклатура.Наименование = СтрокаДокумента.Номенклатура.Наименование;
    Номенклатура.ПолноеНаименование = СтрокаДокумента.Номенклатура.ПолноеНаименование;
    Номенклатура.ШтрихКод = СтрокаДокумента.Номенклатура.ШтрихКод;
    Номенклатура.ЗакупочнаяЦена = СтрокаДокумента.Номенклатура.ЗакупочнаяЦена;
    // Заполнить свойства строки расходной накладной
    СтрокаРасходнойНакладной.Номенклатура = Номенклатура;
    СтрокаРасходнойНакладной.Количество = СтрокаДокумента.Количество;
    СтрокаРасходнойНакладной.Цена = СтрокаДокумента.Цена;
    СтрокаРасходнойНакладной.Сумма = СтрокаДокумента.Сумма;
    // Добавить строку расходной накладной
    РасходнаяНакладная.Состав.Добавить(СтрокаРасходнойНакладной);

```



```

КонецЦикла;
// Вернуть расходную накладную
Возврат РасходнаяНакладная;
КонецФункции

```

В заключение следует опубликовать созданный Web-сервис на веб-сервере, например <http://www.MyCompany.ru>, в каталоге `shipment`.

18.1.4. Работа с веб-сервисами сторонних поставщиков

Система «1С:Предприятие» может использовать веб-сервисы, предоставляемые другими поставщиками, двумя способами:

- с помощью статических ссылок, создаваемых в дереве конфигурации;
- с помощью динамических ссылок, создаваемых средствами встроенного языка.

Преимущество использования статических ссылок заключается в большей скорости работы, т. к. описание веб-сервиса поставщика получается один раз, при создании ссылки. В дальнейшем при обращении к данному веб-сервису используется существующее описание веб-сервиса.

При использовании динамических ссылок описание веб-сервиса поставщика будет получаться системой «1С:Предприятие» каждый раз при вызове веб-сервиса, что, естественно, будет замедлять работу с данным веб-сервисом. Однако преимуществом такого подхода является возможность получения актуального описания веб-сервиса поставщика. При использовании же статических ссылок для получения актуального описания веб-сервиса следует выполнить повторный импорт WSDL-описания средствами конфигурирования и сохранение измененной конфигурации.

18.1.4.1. Пример использования статической WS-ссылки

В качестве примера использования веб-сервисов стороннего поставщика рассмотрим обращение к Web-сервису, который ранее был создан в примере (см. [здесь](#)).

Прежде всего, следует добавить в дерево конфигурации новый объект конфигурации WS-ссылка с именем `ДанныеРасходнойНакладной`, ссылающийся на опубликованный сервис. Для этого следует выполнить импорт WSDL-описания опубликованного сервиса и в качестве URL указать http://www.MyCompany.ru/shipment/ws/shipment_1cws?wsdl. Описание импорта WSDL-описания см. [здесь](#).

После этого, например, в модуле приходной накладной, можно создать процедуру, приведенную ниже. Она заполняет табличную часть документа данными расходной накладной поставщика, полученными с помощью веб-сервиса поставщика.

```

Процедура ПолучитьДанныеРасходнойНакладной(НомерНакладнойПоставщика)
// Создать WS-прокси на основании ссылки
// и выполнить операцию Получить()
Прокси = WSCСсылки.ДанныеРасходнойНакладной. СоздатьWSПрокси("http://www.MyCompany.ru/shipment", "ДанныеРасходнойНакладной", "ДанныеРасходнойНакладнойSoap");
ДанныеНакладной = Прокси.Получить();
Если ДанныеНакладной = Неопределено Тогда
    Возврат;
КонецЕсли;
// Заполнить приходную накладную полученными данными
Для Каждого СтрокаНакл Из ДанныеНакладной.Состав Цикл
    НоваяСтрока = ДокументОбъект.Состав.Добавить();
    НоваяСтрока.Количество = СтрокаНакл.Количество;
    НоваяСтрока.Цена = СтрокаНакл.Цена;
    НоваяСтрока.Сумма = СтрокаНакл.Сумма;
// Найти элемент номенклатуры по переданным данным
// (например, по штрихкоду)
НоваяСтрока.Номенклатура = Справочники.Номенклатура.НайтиПоРеквизиту("ШтрихКод", СтрокаНакл.Номенклатура.ШтрихКод);
КонецЦикла;
КонецПроцедуры

```

[Копировать в буфер обмена](#)

18.1.4.2. Пример использования динамической WS-ссылки

Использование динамической ссылки отличается от использования статической ссылки только способом создания WS-прокси и отсутствием необходимости создавать WS-ссылку в дереве конфигурации.

Если провести сравнение с примером, представленным в предыдущем разделе, то, в отличие от создания прокси на основе статической ссылки, при использовании динамической ссылки WS-прокси создается с помощью конструктора следующим образом:

```

// Создать WSпрокси на основании WS-определения
// и выполнить операцию Получить()
Определение = Новый WSOпределения("http://www.MyCompany.ru/shipment/ws/shipment_1cws?wsdl");
Прокси = Новый WSПрокси(Определение, "http://www.MyCompany.ru/shipment", "ДанныеРасходнойНакладной", "ДанныеРасходнойНакладнойSoap");
ДанныеНакладной = Прокси.Получить();

```

[Копировать в буфер обмена](#)

В то время как создание WS-прокси на основе статической ссылки выглядит следующим образом:

```

// Создать WSпрокси на основании ссылки
// и выполнить операцию Получить()
Прокси = WSCСсылки.ДанныеРасходнойНакладной. СоздатьWSПрокси("http://www.MyCompany.ru/shipment", "ДанныеРасходнойНакладной", "ДанныеРасходнойНакладнойSoap");
ДанныеНакладной = Прокси.Получить();

```

[Копировать в буфер обмена](#)

18.1.5. Редактирование свойств Web-сервиса

На закладке `Основные` вводится имя, синоним и комментарий объекта.

На закладке `Операции` создаются подчиненные объекты `Операции`, которые, в свою очередь, могут иметь подчиненные объекты `Параметры`, необходимые для работы с объектами данного типа. Описание подчиненных объектов выполняется в палитре свойств.

18.1.5.1. Свойства «Операции»

Помимо общих свойств объектов конфигурации операция Web-сервиса содержит следующие свойства:

- **Тип возвращаемого значения** – тип значения, которое возвращает операция Web-сервиса. Может являться типом значения XDTO или типом объекта XDTO.
- **Возможно пустое значение** – показывает, может ли возвращаемое значение принимать неопределенное значение.
- **В транзакции** – показывает, будет ли выполняться код модуля Web-сервиса в транзакции или нет. Если свойство установлено, то при вызове Web-сервиса автоматически будет начата транзакция, а при завершении работы Web-сервиса транзакция будет либо зафиксирована, либо произойдет откат транзакции (в зависимости от результатов выполнения). Если свойство не установлено, при начале исполнения модуля Web-сервиса транзакция не будет начата.
- **Имя метода** – имя экспортируемой процедуры модуля Web-сервиса, которая будет выполнена при вызове данного свойства.

18.1.5.2. Свойства «Параметр»

Помимо общих свойств объектов конфигурации параметр операции Web-сервиса содержит следующие свойства:

- **Тип значения** – тип значения параметра операции Web-сервиса. Может являться типом значения XDTO или типом объекта XDTO.
- **Возможно пустое значение** – показывает, может ли значение параметра операции принимать неопределенное значение.

• **Направление передачи** – определяет направление передачи данных с помощью данного параметра. Возможные значения:

- **Входной** – означает, что параметр используется для передачи данных Web-сервису;
- **Выходной** – означает, что параметр используется для получения данных от Web-сервиса;
- **Входной – Выходной** – означает, что параметр может использоваться как для передачи данных, так и для их получения от Web-сервиса.

На закладке **Подсистемы** указывается, к каким подсистемам относятся объекты данного типа.

На закладке **Прочее** определяются следующие свойства:

- **URI пространства имен** – содержит URI пространства имен Web-сервиса. Каждый Web-сервис может быть однозначно идентифицирован по своему имени и URI пространству имен, которому он принадлежит.
- **Пакеты XDTO** – перечень пакетов XDTO, типы которых могут использоваться в качестве типов возвращаемого значения операций и типов параметров операций Web-сервиса.
- **Имя файла публикации** – имя файла описания Web-сервиса, который расположен на веб-сервере.

По кнопке **Модуль** открывается редактор модуля Web-сервиса.

18.2. HTTP-сервисы

18.2.1. Стандартный интерфейс OData

18.2.1.1. Общая информация

Стандартный интерфейс OData в «1С:Предприятии» предназначен для получения доступа к данным системы из внешнего приложения без модификации кода прикладного решения (например, если прикладное решение стоит на поддержке). Для получения такого доступа необходимо особым образом опубликовать приложение на веб-сервере и указать, какие объекты конфигурации будут использоваться таким образом (см. [здесь](#)).

Для доступа к данным используется протокол OData (<http://www.odata.org/>, на английском языке) версии 3 (<http://www.odata.org/documentation/odata-version-3-0/odata-version-3-0-core-protocol>, на английском языке). Поддерживаются следующие форматы представления данных: **atom-xml** и **json**. Для доступа к данным, при публикации, автоматически генерируется стандартный интерфейс OData, который позволяет читать данные «1С:Предприятия», изменять их, создавать новые объекты данных и удалять существующие. Прикладное решение на базе «1С:Предприятия» может выступать как клиентом, так и сервером при работе со стандартным интерфейсом OData. Для работы сервером практически никаких дополнительных действий осуществлять не надо (эта возможность предоставляется автоматически). Для того, чтобы стать клиентом стандартного интерфейса OData, необходимо в прикладном решении реализовать программный слой, который будет использовать данные сервера с использованием стандартных интерфейсов «1С:Предприятия», например объекта **HTTPСоединение**.

В одной системе могут одновременно существовать и SOAP-сервисы (которые реализуются в «1С:Предприятие» в виде Web-сервисов, см. [здесь](#)) и стандартный интерфейс OData и HTTP-сервис (см. [здесь](#)).

Стандартный интерфейс OData можно использовать для следующих задач:

- Интеграция прикладного решения с различными сайтами (например, на базе Microsoft SharePoint);
- Реализация сторонними средствами дополнительной функциональности прикладного решения без изменения его конфигурации;
- Загрузка данных в прикладное решение и выгрузка данных из него;
- Интеграция прикладного решения с корпоративными системами, возможно даже без дополнительного программирования.

Типичные операции, выполняемые с помощью стандартного интерфейса OData:

- Получение списка объектов системы с установленным отбором;
- Получение данных конкретного объекта системы;
- Создание нового и запись изменений существующего объекта системы;
- Проведение одного документа, старт бизнес-процесса.

18.2.1.2. Общие принципы работы

При работе с протоколом OData используется специальная терминология. При применении протокола используются следующие термины:

- **Сущность** – нечто, обладающее идентичностью. Сущность также обладает набором свойств. Некоторые свойства описывают ее идентичность, комбинация которых определяет ключ сущности. По этому ключу можно получить конкретную сущность.
- **Набор сущностей** – коллекция сущностей определенного типа.
- **Составной тип** – набор свойств, не обладающий идентичностью.
- **Функция** – набор некоторых операций, выполняемых на стороне сервера, возвращающий данные (не обязательно сущность или набор сущностей) и не приводящий к наблюдаемым побочным эффектам (изменениям данных). Функция обязательно связана с сущностью или набором сущностей.
- **Действие** – функция, которая может изменять данные.

Термины протокола некоторым образом соответствуют терминам, принятым в системе «1С:Предприятие»:

- **Сущность** – в качестве сущности может выступать один из 4 групп объектов системы «1С:Предприятие»: объектные типы, наборы записей регистров, записи регистров и строки табличных частей объектных типов. Записи регистров и строки табличных частей объектных типов (как отдельные сущности) доступны только на чтение.
- Реквизиты объектов «1С:Предприятия» представляются как **свойства** сущностей. В некоторых случаях (например, реквизит объекта конфигурации составного типа) реквизит может быть представлен несколькими свойствами, одно из которых будет **навигационным**. Такое свойство содержит в качестве значения ссылку (URL) на сущность, описывающую объект «1С:Предприятия». Свойство, описывающее тип такого реквизита, называется **диспетчеризационным**. Название такого свойства завершается суффиксом **_Type**. Более подробное описание диспетчеризационного свойства приведено далее.

Обращение к стандартному интерфейсу OData выполняется с помощью HTTP-запроса, выполнено по определенному URL. URL формируется специальным образом и состоит из следующих частей:

1. Адрес информационной базы;
2. Признак обращения к стандартному интерфейсу OData;
3. Имя ресурса, к которому выполняется обращение;
4. Параметры запроса обращения к ресурсу.

Само обращение выполняется с помощью HTTP-запроса определенного вида. При более подробном описании работы со стандартным интерфейсом OData вид запроса будет указываться отдельно.

Рассмотрим части URL более подробно.

Адрес информационной базы

Это обычный URL, по которому выполняется доступ, например, к информационной базе с помощью веб-клиента. Например, <http://host/base> или <http://host.server.zone/data-base>. Также необходимо помнить, что при использовании информационной базы, в которой настроено разделение, допустимо использовать только один способ указания значений разделителей, а именно: значения разделителей можно указывать только в URL информационной базы. Указание разделителей с помощью параметра **Z** не поддерживается. Более подробная информация про настройки указания значений разделителей в URL информационной базы можно получить в описании файла [default.vrd](#) в книге «1С:Предприятие 8.3. Руководство администратора».

Признак обращения к стандартному интерфейсу OData

В качестве такого признака выступает последовательность `/odata/standard.odata`.

Имя ресурса, к которому выполняется обращение

Особым образом сформированный идентификатор ресурса (возможно, с параметром) или предопределенные ресурсы. Например, `$metadata`, `Catalog_Контрагент(guid'value)`.

Параметры обращения к ресурсу

В качестве параметров обращения выступают параметры в виде, принятом для HTTP-запросов: `?ключ=значение&ключ2=значение2`.

При обращении к ресурсу могут использоваться специальные ключевые слова, имеющие специальное назначение:

- **\$format** – указывает, в каком формате необходимо получить данные. Если ключевое слово не указано, данные получаются в формате `atom+xml`.
 - **\$format=atom** – возвращает данные в формате `atom+xml`.
 - **\$format=json** – возвращает данные в формате `json`. Для указания того, что данные должны возвращаться в формате `json`, можно указать MIME-тип `application/json` в заголовке `Accept` http-запроса на получение данных.
- **\$metadata** – указывает, что требуется получить описание стандартного интерфейса OData. Подробнее см. [здесь](#).
- **\$filter** – описывает отбор, применяемый при получении данных. Подробнее см. [здесь](#).
- **\$select** – описывает перечень свойств сущности, которые получаются при обращении к стандартному интерфейсу OData (см. [здесь](#)).

После того, как сформирован URL необходимого ресурса, следует выполнить HTTP-запрос нужного вида. В зависимости от того, какая операция выполняется, используется соответствующий HTTP-метод:

- Получение данных – метод `GET`;
- Создание объекта – метод `POST`;
- Обновление данных:
 - метод `PATCH` – в этом случае можно указывать только те свойства, которые необходимо обновить;
 - метод `PUT` – в этом случае необходимо указывать все свойства сущности;
- Удаление данных – метод `DELETE`.

В результате выполнения запроса клиентское приложение получает ответ сервера, который кроме кода состояния может содержать различные данные, предоставленные сервером, в виде XML-документа.

В примерах URL, которые используются в данном разделе, выражение `guid'value` означает выражение OData, которое описывает конкретное значение GUID (тип `Edm.Guid`).

18.2.1.3. Представление данных

Данные, возвращаемые стандартным интерфейсом OData, могут быть представлены в виде XML-документа или JSON-документа. Это зависит от того, в каком формате запрашиваются данные.

Для различных типов данных используется следующее соответствие между типом «1С:Предприятия» и типом OData:

Тип «1С:Предприятия»	Тип свойства OData
Строка	<code>Edm.String</code>
Дата	<code>Edm.DateTime</code>
Число (целое)	<code>Edm.Int16</code> , <code>Edm.Int32</code> , <code>Edm.Int64</code>
Число (дробное)	<code>Edm.Double</code>
Булево	<code>Edm.Boolean</code>
Ссылка	<code>Edm.Guid</code>
Перечисление	<code>Edm.String</code>
ХранилищеЗначения	Состоит из трех свойств*: <ol style="list-style-type: none"> 1. Навигационное свойство; 2. <code><Имя свойства>_Base64Data</code>; 3. <code><Имя свойства>_Type</code>.
Ссылочный тип	Два свойства: <ul style="list-style-type: none"> • Собственно значение ссылки на объект типа <code>Edm.Guid</code>; • Навигационное свойство.
Составные типы	Два свойства типа <code>Edm.String</code> : <ul style="list-style-type: none"> • Значение фактического типа, выраженное в виде строки; • Имя фактического типа.

В таблице символ * означает, что:

- С помощью навигационного свойства можно получить данные, хранящиеся в реквизите:
 - Для типа `Картинка` – будет получена собственно картинка с соответствующим значением HTTP-заголовка `content-type`;

- Для типа [ДвоичныеДанные](#) – будет получен байтовый поток;
- Для остальных типов – XDTO-сериализованное значение хранимых данных.
- Свойство `<Имя свойства>_Base64Data` хранит данные, которые могут быть получены с помощью навигационного свойства, только закодированные в Base64. Редактировать данные реквизита типа [ХранилищеЗначения](#) можно только с помощью этого свойства.
- Свойство `<Имя свойства>_Type` описывает тип данных, хранимых в реквизите. Может принимать одно и трех значений:
 - `application/octet-stream` – двоичные данные;
 - `application/xml+xdto` – XDTO-сериализованный объект;
 - значение HTTP-заголовка `content-type`, соответствующего картинке, хранящейся в реквизите, например, `image/jpeg` для картинки формата JPEG.

Остальные типы не поддерживаются, и при попытке их чтения будет сгенерирована ошибка с кодом 501.

Имена свойств могут оканчиваться на различные суффиксы. Такие свойства имеют специальный смысл. Далее будут более подробно рассмотрены возможные суффиксы:

- [Key](#);
- [Type](#);
- [Base64Data](#).

Key

Свойство с таким суффиксом содержит значение ключа соответствующего ссылочного реквизита объекта (без такого суффикса). Имя с таким суффиксом используется для установки отбора в качестве имени реквизита, по которому выполняется отбор. Например, для установки отбора по ссылочному полю [Контрагент](#), условие будет выглядеть следующим образом: `Контрагент_Key=guid'value`. При этом свойство [Контрагент](#) будет содержать представление контрагента с указанным значением ссылки.

Type

Данный суффикс используется для описания реквизита составного типа. Так, если в данных есть поле составного типа [Контрагент](#), то в документе, который возвращает стандартный интерфейс OData, этому полю будет соответствовать два свойства:

- [Контрагент_Type](#) – будет содержать описание типа значения реквизита в виде строки (тип `Edm.String`, диспетчеризационное свойство);
- [Контрагент](#) – будет содержать значение реквизита (соответствующего типа).

Перечень допустимых типов, которые могут быть использованы в поле с таким суффиксом, определяется схемой сервиса, который можно получить при запросе полного описания стандартного интерфейса OData (см. [здесь](#)). Таким образом, при необходимости установить тип [Документ.РасходТовара](#), в элемент с суффиксом `_Type` должно быть записано значение `StandardODATA.Document_РасходТовара`.

Если значение реквизита составного типа в информационной базе «1С:Предприятия» имеет значение [Неопределено](#), то диспетчеризационное свойство будет иметь значение `StandardODATA.Undefined`, а само значение свойства должно игнорироваться.

Пример представления реквизита составного типа:

```
<d:РеквизитСоставной/>
<d:РеквизитСоставной_Type>StandardODATA.Undefined</d:РеквизитСоставной_Type>
```

[Копировать в буфер обмена](#)

Base64Data

Данный суффикс используется при указании имени свойства, содержащего данные, расположенные в реквизите типа [ХранилищеЗначения](#), в виде строки `Base64`. Так, если в объекте конфигурации есть реквизит [Файл](#), который имеет тип [ХранилищеЗначения](#), то в документе, который возвращает стандартный интерфейс OData, этому полю будет соответствовать два свойства:

- [Файл_Type](#) – содержит наименование типа данных, хранимых реквизитом;
- [Файл_Base64Data](#) – содержит строку `Base64`, содержащую сами данные.

18.2.1.4. Правила формирования имени ресурса

При обращении к какому-либо ресурсу, его идентификатор формируется по следующему принципу: `ПрефиксИмени_ИмяОбъектаКонфигурации_СуффиксИмени`. С помощью стандартного интерфейса OData можно получить доступ к следующим объектам ([ПрефиксИмени](#)):

Объект конфигурации	Префикс имени для указания в URL
Справочник	Catalog
Документ	Document
Журнал документов	DocumentJournal
Константа	Constant
План обмена	ExchangePlan
План счетов	ChartOfAccounts
План видов расчета	ChartOfCalculationTypes
План видов характеристик	ChartOfCharacteristicTypes
Регистр сведений	InformationRegister
Регистр накопления	AccumulationRegister
Регистр расчета	CalculationRegister
Регистр бухгалтерии	AccountingRegister
Бизнес-процесс	BusinessProcess
Задача	Task

`ИмяОбъектаКонфигурации` – свойство `Имя` объекта конфигурации, как оно задано при разработке прикладного решения в конфигураторе.

`СуффиксИмени` – предназначено для уточнения имени ресурса и является необязательной частью имени. В качестве суффикса имени могут выступать следующие выражения:

- Имя табличной части объекта;
- Имя виртуальной таблицы регистра;
- [RowType](#);

- RecordType;

Далее будут более подробно рассмотрены вышеописанные уточнения имени ресурса:

Имя табличной части объекта

Если объект обладает табличной частью, то для получения доступа ко всем записям этой табличной части необходимо добавить имя табличной части после имени самого объекта, например, для получения всех строк табличных частей **Товары** всех документов **РасходТовара** будет необходимо выполнить GET-запрос по следующему адресу: [http:// host/base/odata/standard.odata/Document_РасходТовара_Товары](http://host/base/odata/standard.odata/Document_РасходТовара_Товары).

Имя виртуальной таблицы регистра

В роли виртуальной таблицы регистра выступает функция, связанная с ресурсом, возвращающей набор сущностей регистра. Имя функции совпадает с английским вариантом имени используемой виртуальной таблицы языка запросов. Параметры функции соответствуют параметрам виртуальной таблицы. Так, для получения среза последних для регистра сведений **КурсыВалют**, следует выполнить GET-запрос по следующему адресу: [http://localhost/demo/odata/standard.odata/InformationRegister_КурсыВалют/SliceLast\(\)](http://localhost/demo/odata/standard.odata/InformationRegister_КурсыВалют/SliceLast()).

RowType

Сущность с таким суффиксом описывает тип строки табличной части какого-либо объекта.

RecordType

Сущность с таким суффиксом описывает отдельную запись регистра.

18.2.1.5. Правила формирования условия отбора

\$filter

При получении данных можно фильтровать данные. Для этого предназначен специальный язык, который позволяет описывать условия, каким должны соответствовать данные, которые возвращает стандартный интерфейс OData. Описание отбора начинается с ключевого слова **\$filter**, после которого следует собственно условие. Поддерживаются следующие операции:

- Логические операции:

Описание	Имя	Пример
Равно	eq	/Catalog_Города?\$filter=Наименование eq 'Главный'
Не равно	ne	/Catalog_Города?\$filter=Наименование ne 'Пермь'
Больше	gt	/Catalog_Товары?\$filter=Цена gt 10
Больше или равно	ge	/Catalog_Товары?\$filter=Цена ge 10
Меньше	lt	/Catalog_Товары?\$filter=Цена lt 10
Меньше или равно	le	/Catalog_Товары?\$filter=Цена le 10
Логическое ИЛИ	or	/Catalog_Товары?\$filter=Цена lt 10 or Цена gt 100
Логическое И	and	/Catalog_Товары?\$filter=Цена gt 10 and Цена lt 100
Отрицание	not	/Catalog_Товары?\$filter=not (Цена eq 10)

- Арифметические операции:

Описание	Имя	Пример
Сложение	add	/Catalog_Товары?\$filter=Цена add 5 gt 10
Вычитание	sub	/Catalog_Товары?\$filter=Цена sub 5 gt 10
Умножение	mul	/Catalog_Товары?\$filter=Цена mul 5 gt 1000
Деление	div	/Catalog_Товары?\$filter=Цена div 4 gt 2

- Группирующие операторы:

Описание	Имя	Пример
Приоритет операции	()	/Catalog_Товары?\$filter=(Цена add 5) gt 10

Пример отбора:

[http://host/odata/standard.odata/Catalog_Товары?\\$filter=Имя eq 'Молоко' and Цена lt 2500](http://host/odata/standard.odata/Catalog_Товары?$filter=Имя eq 'Молоко' and Цена lt 2500)

[Копировать в буфер обмена](#)

При формировании условия отбора следует учитывать приоритет операций. В следующей таблице представлен список операций языка выражений отбора в порядке уменьшения приоритета. Операции с одинаковым приоритетом вычисляются слева направо:

Оператор	Описание
()	Повышение приоритета операции
/	Навигация
-	Арифметическое отрицание
Not	Логическое отрицание
Mul	Умножение
Div	Деление
Add	Сложение
Sub	Вычитание
Gt	Больше
Ge	Больше или равно
Lt	Меньше
Le	Меньше или равно
Eq	Равно
Ne	Не равно
And	Логическое «И»
Or	Логическое «ИЛИ»

ПРИМЕЧАНИЕ. Не поддерживаются операции сравнения с реквизитом составного типа, типа *УникальныйИдентификатор* и типа *ХранилищеЗначения*.

Не поддерживаются следующие операции сравнения:

- Стандартные функции в запросах;
- Лямбда-выражения и условия на свойства-коллекции.

Stop

Имеется возможность ограничить количество записей, возвращаемых при обращении к ресурсу. Для этого используется ключевое слово **\$top**.

Пример:

[http://host/odata/standard.odata/Catalog_Товары?\\$filter=Цена lt 1000&\\$top=10](http://host/odata/standard.odata/Catalog_Товары?$filter=Цена lt 1000&$top=10)

[Копировать в буфер обмена](#)

allowedOnly

Если при выполнении запроса необходимо получить только те объекты данных, которые не попадают под ограничения доступа к данным, то в URL получения данных необходимо добавить параметр **allowedOnly**.

Пример:

http://host/base/odata/standard.odata/Catalog_Товары?allowedOnly=true

[Копировать в буфер обмена](#)

Если параметр не указан или указан со значением **false**, то во время исполнения запроса к данным может возникнуть ошибка с кодом 401 (если результат выполнения запроса содержит данные, доступ к которым запрещен). Ошибка может не произойти в том случае, если были указаны дополнительные условия, которые ограничили выборку только разрешенными данными.

Например, для справочника **Данные** настроено ограничение доступа к данным, которое не позволяет получить элементы справочника, у которых реквизит **ЗначениеСекретности** равно 1, но позволяет получить элементы справочника, у которых реквизит **ЗначениеСекретности** равно 2. В этом случае следующий запрос не приведет к возникновению ошибки, т. к. явно накладывается условие, которое оставляет в выборе только разрешенные данные:

[http://host/base/odata/standard.odata/Catalog_Данные?allowedOnly=false&\\$filter=ЗначениеСекретности eq 2](http://host/base/odata/standard.odata/Catalog_Данные?allowedOnly=false&$filter=ЗначениеСекретности eq 2)

[Копировать в буфер обмена](#)

Параметр **allowedOnly** можно использовать только для GET-запросов к наборам сущностей.

Описание возможностей отбора в протоколе OData можно получить в документации по протоколу (на английском языке):

- http://www.odata.org/documentation/odata-v3-documentation/odata-core/#10231_The_filter_System_Query_Option;
- http://www.odata.org/documentation/odata-v3-documentation/url-conventions/#512_Filter_System_Query_Option.

18.2.1.6. Способы получения описания стандартного интерфейса OData

Для получения упрощенного описания стандартного интерфейса OData (только список сущностей) необходимо выполнить GET-запрос с использованием URL <http://host/base/odata/standard.odata>.

Для получения описания стандартного интерфейса OData необходимо выполнить GET-запрос с использованием URL [http://host/base/odata/standard.odata/\\$metadata](http://host/base/odata/standard.odata/$metadata). В результате будет получен полный список доступных сущностей, их атрибутов и функций в виде XML-документа. Подробное описание документа можно получить по адресу <http://www.odata.org/documentation/odata-v3-documentation/common-schema-definition-language-csdl/> (на английском языке).

В случае получения данных в формате **json**, не поддерживается запрос получения расширенного описания метаданных вида [http://host/base/odata/standard.odata/\\$metadata](http://host/base/odata/standard.odata/$metadata). Однако имеется возможность управлять детализацией информации о метаданных, которая возвращается при получении данных (как сущностей, так и списков сущностей). Для этого необходимо особым образом формировать параметр **\$format** при выполнении запроса:

- **\$format=application/json;odata=minimalmetadata** – в этом случае информация о метаданных передается в минимальном объеме. Это значение по умолчанию.
- **\$format= application/json;odata=nometadata** – в этом случае информация о метаданных не передается вовсе.
- **\$format= application/json;odata=fullmetadata** – в этом случае информация о метаданных передается в полном объеме.

18.2.1.7. Способы получения данных

С помощью стандартного интерфейса OData можно получать как списки сущностей, так и сами сущности. Эти способы получения данных отличаются URL, по которому происходит обращение к данным.

Для получения списка сущностей URL выглядит следующим образом: http://host/base/odata/standard.odata/Catalog_Товары. Это **URL набора сущностей**. Для получения сущности URL будет выглядеть следующим образом: [http://host/base/odata/standard.odata/Catalog_Товары\(guid'value\)](http://host/base/odata/standard.odata/Catalog_Товары(guid'value)). В общем виде такой адрес будет называться **канонический URL экземпляра сущности**. Разница заключается в том, что в первом случае получается вся таблица справочника товаров, а вот втором выбирается один объект, который описывается набором ключевых значений (в примере ключевое значение единственное – ссылка на объект). В конкретном случае значени набор ключевых полей, описывающих сущность, зависит от получаемой сущности. Если ключевых полей более одного, то они должны быть указаны все. Ключевые значения указываются парами **Имя=Значение**, разделенными запятыми. Если ключевое значение одно – имя ключа можно не указывать. Например, для получения значения курса валюты на конкретную дату необходимо выполнить GET-запрос со следующим URL: [http://host/base/odata/standard.odata/InformationRegister_КурсыВалют\(Period=datetime'2008-02-05T00:00:00', Валюта_Key=guid'9d5c4222-8c4c-11db-a9b0-00055d49b45e'\)](http://host/base/odata/standard.odata/InformationRegister_КурсыВалют(Period=datetime'2008-02-05T00:00:00', Валюта_Key=guid'9d5c4222-8c4c-11db-a9b0-00055d49b45e')).

При получении данных существует возможность указать, какие свойства сущности (или набора сущностей) необходимо получить с помощью стандартного интерфейса OData. Для этого существует ключевое слово **\$select**, которое имеет несколько вариантов использования:

- Признак получения всех свойств.

В этом случае следует указать выражение вида **\$select=*** или совсем не указывать выражение **\$select**.

- Перечень конкретных свойств, которые необходимо получить.

В этом случае необходимо указать в выражении оператора **\$select** перечень (через запятую) всех необходимых полей. При этом следует указывать суффиксы свойств, если это необходимо. Например, для получения значения ссылки на справочник и свойств **Код** и **Артикул**, для справочника **Товары**, необходимо использовать следующий URL: [http://host/base/odata/standard.odata/Catalog_Товары?\\$select=Ref_Key, Код, Артикул](http://host/base/odata/standard.odata/Catalog_Товары?$select=Ref_Key, Код, Артикул).

Для получения этих же полей для конкретного экземпляра сущности (с известным идентификатором) будет использоваться следующий URL: [http://host/base/odata/standard.odata/Catalog_Товары\(guid'value\)?\\$select=Ref_Key, Код, Артикул](http://host/base/odata/standard.odata/Catalog_Товары(guid'value)?$select=Ref_Key, Код, Артикул).

- Признак получения всех свойств, кроме табличных частей.

Для этого в выражении оператора **\$select** необходимо указать значение ******. При использовании следующего URL будут получены все документы **РасходТовара** со всеми реквизитами, кроме табличных частей: [http://host/base/odata/standard.odata/Document_РасходТовара?\\$select=**](http://host/base/odata/standard.odata/Document_РасходТовара?$select=**).

Имеется возможность получать данные, доступные «через точку». В этом случае к адресу, описывающему конкретную сущность, прибавляются имена свойств, идущие «через точку», но разделенные символом «/». При этом для стандартных реквизитов следует использовать только английские имена реквизитов. Например, для получения свойства **Наименование** реквизита **Валюта** документа **РасходТовара**, необходимо использовать GET-запрос со следующим URL:

[http://host/base/odata/standard.odata/Document_РасходТовара\(guid'value\)/Валюта/Description](http://host/base/odata/standard.odata/Document_РасходТовара(guid'value)/Валюта/Description).

18.2.1.8. Выполнение функций и действий

С некоторыми сущностями и наборами сущностей могут быть связаны функции. Например, через функции выполняется работа с виртуальными таблицами регистров. В этом случае URL ресурса формируется следующим образом: [. Подробнее рассмотрим, из чего состоит адрес. Первая часть адреса \(<http://host/base/odata/standard.odata>\) представляет собой стандартный префикс адреса при обращении к стандартному интерфейсу OData. \[<ресурс>\]\(#\) – имя ресурса, правила формирования которого см. \[здесь\]\(#\). Имя функции соответствует англоязычному имени виртуальной таблицы языка запросов \(см. \[здесь\]\(#\)\). \[<параметры>\]\(#\) у функции задаются парами **Ключ=Значение** и разделяются запятыми. Если в качестве параметра функции используется отбор, то выражение, описывающее отбор, должно удовлетворять общим правилам описания отборов \(см. \[здесь\]\(#\)\). Так, получение среза первых для регистра курсов валют \(с параметрами\), будет выполняться по следующему URL: \[http:// host/base/odata/standard.odata/InformationRegister_КурсыВалют/SliceFirst\\(Period=datetime'2008-01-01T00:00:00',Condition='Валюта_Key eq guid'value"\\)\]\(http://host/base/odata/standard.odata/InformationRegister_КурсыВалют/SliceFirst\(Period=datetime'2008-01-01T00:00:00',Condition='Валюта_Key eq guid'value'\)\).](http://host/base/odata/standard.odata/<ресурс>/<функция>(<параметры>)

18.2.1.9. Ошибочные ситуации

В случае ошибочной ситуации возвращается ответ с HTTP-статусом **4xx** или **5xx**. Статус **4xx** информирует об ошибках на стороне клиентского приложения, статус **5xx** информирует об ошибке на стороне сервера.

В случае статуса **4xx** сервер пытается уточнить причину ошибки и может передать клиентскому приложению дополнительный внутренний код ошибки и информационное сообщение (в виде xml-документа) в теле ответа.

Пример:

```
<m:error xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
  <m:code>9</m:code>
  <m:message>Экземпляр сущности "НакладнаяОтгрузки" не найден по переданному ключу.</m:message>
</m:error>
```

[Копировать в буфер обмена](#)

Далее перечислены внутренние коды ошибок с описанием причины появления:

Код	Описание
0	Возможность не поддерживается
1	Не удалось разобрать строку
2	Неверный формат запроса
3	Запрошенный тип представления не поддерживается
4	Неверное значение свойства
5	Отсутствует обязательное значение свойства
6	Неверный URL
7	Не хватает элемента ключа сущности
8	Тип сущности не найден
9	Экземпляр сущности не найден
10	Запрошенное свойство не найдено
11	Метод не найден
12	Отсутствует обязательный аргумент метода
13	Создание строк табличных частей напрямую не поддерживается
14	Ошибка разбора опций запроса
15	Сущность с таким ключом уже существует
16	Не удалось присвоить свойство
17	Объект не поддерживает режим загрузки данных
18	Ошибка инициализации интерфейса OData: в объекте есть свойства с одинаковыми именами
19	Использованный HTTP-метод запрещен в данном контексте
20	Ошибка прав доступа. Может возникать: <ul style="list-style-type: none"> • Когда у пользователя нет прав на запрошенное действие над данным объектом; • Когда в выборку попадает объект, недоступный в связи с ограничением доступа к данным и параметр allowedOnly не используется.

18.2.1.10. Установка значений разделителей

Значения разделителей можно устанавливать только следующими способами:

- С помощью фрагментов URL при обращении к стандартному интерфейсу OData;
- С помощью файла описания публикации [default.vrd](#).

Указанием разделителей с помощью параметра командной строки **Z** не поддерживается.

18.2.1.11. Публикация стандартного интерфейса OData

Публикация стандартного интерфейса OData выполняется с помощью диалога публикации на веб-сервере ([Администрирование – Публикация на веб-сервере](#)) и описано в книге «1С:Предприятие 8.3. Руководство администратора».

Для того чтобы объекты конфигурации стали доступны через стандартный интерфейс OData, необходимо разрешить это с помощью метода глобального контекста [УстановитьСоставСтандартногоИнтерфейсаOData\(\)](#). Если прикладное решение функционирует в режиме совместимости с версией 8.3.4 (и ниже), данный метод не используется. В этом случае с помощью стандартного интерфейса OData доступны все поддерживаемые объекты конфигурации.

С помощью данного метода можно ограничить перечень доступных объектов конфигурации только теми (из общего перечня поддерживаемых, см. [здесь](#)), доступ к которым действительно необходим для внешних приложений.

Механизм установки состава объектов, доступных с помощью стандартного интерфейса OData, можно выполнить в виде внешней обработки. Для этого не требуется модифицировать прикладное решение.

18.2.1.12. Выполнение типовых операций

В данном разделе будут приведены примеры выполнения некоторых типовых операций по работе с данными с помощью стандартного интерфейса OData.

ВНИМАНИЕ! Данные примеры не являются законченными. Они приведены только для иллюстрации применения различных конструкций.

18.2.1.12.1. Работа с одним объектом

Чтение данных

Для получения информации о сущности следует использовать канонический URL сущности. Чтение выполняется с помощью GET-запроса.

Пример чтения ссылочного объекта:

```
http://host/base/odata/standard.odata/Catalog_Товары(guid'value')
```

[Копировать в буфер обмена](#)

Пример чтения набора записей подчиненного регистра сведений:

```
http://host/base/odata/standard.odata/InformationRegister_ПриходныеЦены(Recorder_Key=guid'value')
```

[Копировать в буфер обмена](#)

В данном примере **guid'value'** идентифицирует документ, выполнивший движение по регистру сведений.

Пример чтения записи регистра сведений:

```
http://host/base/odata/standard.odata/InformationRegister_ПриходныеЦены_RecordType(Товар_Key=guid'value', ТипЦены="Приходная")
```

[Копировать в буфер обмена](#)

Пример чтения строки конкретного документа:

```
http://host/base/odata/standard.odata/Document_РасходТовара_Товары(Ref_Key=guid'value', LineNumber=1)
```

[Копировать в буфер обмена](#)

Создание объекта

Для создания объекта следует воспользоваться POST-запросом с использованием URL набора сущностей, передав в теле запроса XML-документ (в формате **atom**), который содержит значения полей создаваемого объекта.

Далее приводится пример создания элемента справочника **Товары** и ответ стандартного интерфейса OData после успешного выполнения операции.

Пример POST-запроса:

```
POST http://host/base/odata/standard.odata/Catalog_Товары HTTP/1.1
```

[Копировать в буфер обмена](#)

```
User-Agent: Fiddler
Host: host
Content-Length: 981
Authorization: Basic QWRtaW5pc3RyYXRvcjox
<entry>
  <category term="StandardODATA.Catalog_Товары" scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
  <title type="text"/>
  <updated>2014-02-14T12:05:55</updated>
  <author/>
  <summary/>
  <content type="application/xml">
    <m:properties xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
      <d:DeletionMark>false</d:DeletionMark>
      <d:Parent_Key>bbb079ae-8c51-11db-a9b0-00055d49b45e</d:Parent_Key>
      <d:IsFolder>false</d:IsFolder>
      <d:Code>000000800</d:Code>
      <d:Description>Шлепанцы</d:Description>
      <d:Артикул>SL56X</d:Артикул>
      <d:Поставщик_Key>086715b0-f348-11db-a9c5-00055d49b45e</d:Поставщик_Key>
      <d:Вид>Товар</d:Вид>
      <d:Штрихкод/>
      <d:Описание>&lt;html&gt;Шлепанцы пляжные&lt;/html&gt;&lt;/d:Описание>
    </m:properties>
  </content>
</entry>
```

Пример ответа стандартного интерфейса OData:

[Копировать в буфер обмена](#)

```
HTTP/1.1 201 Created
Content-Length: 2705
Content-Type: application/atom+xml;type=entry;charset=utf-8
Location: http://host/base/odata/standard.odata/Catalog_Товары(guid'41aa6331-954f-11e3-814b-005056c00008')
Server: Microsoft-IIS/7.5
DataServiceVersion: 3.0
X-Powered-By: ASP.NET
Date: Fri, 14 Feb 2014 08:18:36 GMT
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://www.w3.org/2005/Atom"
  xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
  <id>http://host/base/odata/standard.odata/Catalog_Товары(guid'41aa6331-954f-11e3-814b-005056c00008')</id>
  <category term="StandardODATA.Catalog_Товары" scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
  <title type="text"/>
  <updated>2014-02-14T12:18:36</updated>
  <author/>
  <summary/>
  <link rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/ФайлКартинки"
    href="Catalog_Товары(guid'41aa6331-954f-11e3-814b-005056c00008')/ФайлКартинки"
    type="application/atom+xml;type=entry;charset=utf-8"
    title="ФайлКартинки"/>
  <link rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/Parent"
    href="Catalog_Товары(guid'41aa6331-954f-11e3-814b-005056c00008')/Parent"
    type="application/atom+xml;type=entry;charset=utf-8"
    title="Parent"/>
  <link rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/Поставщик"
    href="Catalog_Товары(guid'41aa6331-954f-11e3-814b-005056c00008')/Поставщик"
    type="application/atom+xml;type=entry;charset=utf-8"
    title="Поставщик"/>
  <link rel="edit"
    href="Catalog_Товары(guid'41aa6331-954f-11e3-814b-005056c00008')"
    title="edit-link"/>
  <content type="application/xml">
    <m:properties xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
      <d:Ref_Key>41aa6331-954f-11e3-814b-005056c00008</d:Ref_Key>
      <d:DataVersion m:null="true"/>
      <d:Description>Шлепанцы</d:Description>
      <d:Code>000000800</d:Code>
      <d:Parent_Key>bbb079ae-8c51-11db-a9b0-00055d49b45e</d:Parent_Key>
      <d:IsFolder>false</d:IsFolder>
      <d:DeletionMark>false</d:DeletionMark>
      <d:Артикул>SL56X</d:Артикул>
      <d:Поставщик_Key>086715b0-f348-11db-a9c5-00055d49b45e</d:Поставщик_Key>
      <d:ФайлКартинки_Key>00000000-0000-0000-0000-000000000000</d:ФайлКартинки_Key>
      <d:Вид>Товар</d:Вид>
      <d:Штрихкод/>
      <d:Описание>&lt;html&gt;Шлепанцы пляжные&lt;/html&gt;&lt;/d:Описание>
      <d:ФайлКартинки_Key>00000000-0000-0000-0000-000000000000</d:ФайлКартинки_Key>
      <d:Parent_Key>bbb079ae-8c51-11db-a9b0-00055d49b45e</d:Parent_Key>
      <d:Поставщик_Key>086715b0-f348-11db-a9c5-00055d49b45e</d:Поставщик_Key>
    </m:properties>
  </content>
</entry>
```



```

    </m:properties>
  </content>
</entry>

```

Обновление объекта

Для обновления объекта необходимо выполнить PUT-/PATCH-запрос с использованием канонического URL сущности (аналогично запросу GET для получения сущности), передав в теле запроса XML-документ (в формате atom), который содержит значения свойств сущности.

В случае PATCH-запроса пропущенные свойства сущности будут проигнорированы, т. е. будут изменены только те свойства, которые переданы в запросе на изменение. Для PUT-запроса необходимо указывать значения всех свойств обновляемой сущности.

Удаление объекта

Для удаления объекта следует воспользоваться DELETE-запросом с использованием канонического URL сущности.

ВНИМАНИЕ! Пометка на удаление не выполняется, объект удаляется непосредственно.

Оптимистическая блокировка данных

Для оптимистической блокировки данных (т.е. для проверки, что данные не изменились с момента считывания) нужно использовать заголовок `If-Match` HTTP-запроса, связанного с модификацией (PATCH) или удалением (DELETE) данных. В качестве значения заголовка должно выступать значение свойства `DataVersion`, которое получено при предварительном чтении сущности.

Механизм работает следующим образом:

- При запросе экземпляра сущности среди прочих возвращается свойство `DataVersion`. При запросе набора сущностей с каждой из сущностей набора возвращается свойство `DataVersion`.
- Клиентское приложение должно сохранить у себя это значение версии объекта и затем использовать его в запросах PATCH и DELETE, передавая его в заголовке `If-Match`.
- Если значение свойства `DataVersion` в момент исполнения запроса совпадает со значением заголовка `If-Match`, то происходит запрошенное действие. В противном случае действие не выполняется, а клиенту возвращается HTTP статус 412.

Запись объекта в режиме загрузки данных

Если при записи объекта необходимо эмулировать запись, выполняемую во время работы механизма обмена данными (свойство `ОбменДанными.Загрузка = Истина`) следует использовать HTTP-заголовок `1C_OData_DataLoadMode` со значением `true` при выполнении соответствующей операции записи

Проведение и отмена проведения документа

Для проведения документа необходимо выполнить POST-запрос с использованием канонического URL сущности (аналогично запросу GET для получения сущности), указав особым образом сформированный суффикс URL. Суффикс в этом случае будет состоять из команды `Post` и параметра, указывающего режим проведения документа: [http://host/base/odata/standard.odata/Document_ПасходТоваров\(guid'value'\)/Post?PostingModeOperational=false](http://host/base/odata/standard.odata/Document_ПасходТоваров(guid'value')/Post?PostingModeOperational=false).

Для отмены проведения документа суффикс состоит из команды `Unpost` без параметров.

18.2.1.12.2. Работа с коллекцией объектов

Для получения набора сущностей какого-либо вида, следует выполнить GET-запрос с использованием URL следующего вида: http://host/base/odata/standard.odata/ExchangePlan_ОбменДанными.

Если необходимо установить отбор на получаемый список, его можно выполнить с помощью параметра `$filter` (см. [здесь](#)), который указывается в URL: [http://host/base/standard.odata/Document_Накладная?\\$filter=Приоритет eq 1](http://host/base/standard.odata/Document_Накладная?$filter=Приоритет eq 1).

Обращение к виртуальной таблице регистра выглядит следующим образом (на примере виртуальной таблицы `ОстаткиИОбороты` регистра накопления `ТоварныеЗапасы`): [http://host/base/odata/standard.odata/AccumulationRegister_ТоварныеЗапасы/BalanceAndTurnovers\(StartPeriod=datetime'2014-01-01', EndPeriod=datetime'2014-02-01', Condition='Товар_Key eq guid'value'\)](http://host/base/odata/standard.odata/AccumulationRegister_ТоварныеЗапасы/BalanceAndTurnovers(StartPeriod=datetime'2014-01-01', EndPeriod=datetime'2014-02-01', Condition='Товар_Key eq guid'value')).

Также для ограничения набора сущностей можно использовать параметры `$top` и `$select` (см. [здесь](#)).

18.2.1.12.3. Работа с планами обмена

Формирование сообщения обмена

Для формирования сообщения обмена необходимо выполнить POST-запрос с использованием URL следующего вида: <http://host/base/odata/standard.odata/SelectChanges?<params>>. Где в качестве параметров необходимо указать следующее:

- Параметр `DataExchangePoint` – должен содержать канонический URL сущности требуемого элемента плана обмена;
- Параметр `MessageNo` – должен содержать номер сообщения обмена данными, который будет сформирован в результате данного вызова.

В результате, полный URL для формирования сообщения обмена, будет выглядеть следующим образом: [http://host/base/odata/standard.odata/SelectChanges?DataExchangePoint='http://host/base/odata/standard.odata/ExchangePlan_ОбменДанными\(guid'value\)'\)&MessageNo=34](http://host/base/odata/standard.odata/SelectChanges?DataExchangePoint='http://host/base/odata/standard.odata/ExchangePlan_ОбменДанными(guid'value)')&MessageNo=34).

В результате будет получен список изменений, которые необходимо передать в другой узел, в виде потока `atom-feed`. Каждый элемент будет представлен в формате `atom-entry`, а удаленные элементы будут представлены в формате `atom-deleted-entry` (RFC 6721, <http://tools.ietf.org/html/rfc6721>, на английском языке).

Уведомление о получении изменений

Для уведомления сервера о том, что сообщение обмена успешно получено, необходимо выполнить POST-запрос с использованием URL следующего вида: <http://host/base/odata/standard.odata/NotifyChangesReceived?<params>>. Где в качестве параметров необходимо указать следующее:

- Параметр `DataExchangePoint` – должен содержать канонический URL сущности требуемого элемента плана обмена;
- Параметр `MessageNo` – должен содержать номер сообщения обмена данными, подтверждение получения которого необходимо зафиксировать.

В результате, полный URL для подтверждения получения сообщения обмена, будет выглядеть следующим образом: [http://host/base/odata/standard.odata/NotifyChangesReceived?DataExchangePoint='http://host/base/odata/standard.odata/ExchangePlan_ОбменДанными\(guid'value\)'\)&MessageNo=34](http://host/base/odata/standard.odata/NotifyChangesReceived?DataExchangePoint='http://host/base/odata/standard.odata/ExchangePlan_ОбменДанными(guid'value)')&MessageNo=34).

18.2.2. HTTP-сервисы

18.2.2.1. Общая информация

Помимо стандартного интерфейса OData (см. [здесь](#)), система «1С:Предприятие» позволяет прикладному разработчику реализовать собственные HTTP-сервисы. Создание HTTP-сервисов позволяет расширить (по сравнению со стандартным интерфейсом OData) или создать (если стандартного интерфейса OData не используется) некий прикладной интерфейс, который будет доступен с помощью HTTP-запросов. Общее описание протокола HTTP 1.1 приведено по ссылке: <http://tools.ietf.org/html/rfc2616> (на английском языке).

HTTP-сервисы (REST-сервисы) являются аналогами SOAP-сервисов, которые в «1С:Предприятии» представлены Web-сервисами (см. [здесь](#)). SOAP-сервис – это набор параметризованных действий, причем перечень действий не ограничен. Набор действий определяется во время разработки сервиса. HTTP-сервис – это ограниченный набор

действий (ограничен списком HTTP-методов) и разветвленный набор ресурсов. Состав ресурсов определяем в процессе разработки сервиса. Набор ресурсов и определяет функциональные возможности сервиса. При разработке SOAP-сервисов также определяются структуры данных, передаваемые между клиентом и сервером, а при разработке HTTP-сервиса – структуры передаваемых данных жестко не определяются.

Обращение к HTTP-сервису выполняется по некоторому URL. В качестве примера можно привести HTTP-сервис, который при обращении по одному URL возвращает список каких-либо документов (например, расходных накладных), а при обращении по другому URL будет возвращена конкретная накладная. Действие, которое следует выполнить серверу, определяется тем, какой HTTP-метод адресован серверу. Например, если на сервер отправляется GET-запрос, то выполняется получение каких-либо данных, DELETE-запрос приводит к удалению каких-либо данных и т. д.

При реализации HTTP-сервиса прикладной разработчик определяет следующие элементы сервиса:

1. Определяет базовую часть URL, по которому будет выполняться обращение к сервису.
2. Определяет состав предоставляемой функциональности и структуру ресурсов, на которые эта функциональность будет отображаться.
3. Определяет действия (HTTP-методы), которые можно будет выполнить при обращении к тому или иному ресурсу.
4. Для каждого выбранного действия реализуется специальный метод, написанный на встроенном языке системы «1С:Предприятие», который и реализует необходимую функциональность. Метод расположен в специальном модуле, связанном с HTTP-сервисом.

После реализации HTTP-сервиса, необходимо опубликовать его на веб-сервере с помощью стандартного механизма публикации.

При обращении к HTTP-сервису формируется URL, который выглядит следующим образом: <http://host/base/hs/корневойURL/относительныйURL>. Более подробно рассмотрим составные части адреса:

- <http://host/base> – обычный URL, по которому выполняется доступ, например, к информационной базе с помощью веб-клиента. При наличии разделителей, не поддерживается указание значений разделителей с помощью параметра **Z** командной строки запуска клиентского приложения.
- **hs** – признак того, что выполняется обращение к HTTP-сервису (в отличие от **ws**, который определяет доступ к Web-сервису).
- **корневойURL** – имя ресурса, которое определяет группу ресурсов, связанных общим смыслом. Задается в свойстве объекта **HTTP-сервис**.
- **относительныйURL** – определяет ресурс, к которому будет выполняться обращение. Относительный URL, указанный в запросе, будет использован для определения конкретного ресурса, к которому выполнялось обращение. Правило сопоставления задается в объекте **Шаблон URL**.

18.2.2.2. Разработка HTTP-сервиса

Создание HTTP-сервисов выполняется в ветке **Общие – HTTP-сервисы** дерева объектов конфигурации. Вначале создается собственно сервис, затем в каждом сервисе необходимо создать определенный шаблон фрагмента URL (группу ресурсов), и затем для каждой группы ресурсов необходимо определить HTTP-метод, который будет реализован для данной группы. Также надо понимать, что один сервис может включать в себя более одного шаблона URL (отношение «один ко многим»). Фактически, шаблон URL определяет группу ресурсов, обрабатываемых по единым правилам.

18.2.2.2.1. Редактирование HTTP-сервиса

При создании HTTP-сервиса следует обратить внимание на свойство **Корневой URL**. Определяет группу ресурсов, объединенных общим смыслом. Например, если необходимо создать несколько ресурсов, которые работают с заказами, то корневой URL в этом случае может выглядеть как **order**. Тогда начало URL при обращении к такому сервису будет выглядеть следующим образом: <http://host/base/hs/order>.

18.2.2.2.2. Редактирование шаблона URL

При создании шаблона URL описываются возможные адреса ресурсов, которые можно будет использовать для обращения к HTTP-сервису. Собственно шаблон задается в свойстве **Шаблон**. При создании шаблона можно использовать следующие символы:

- Любые символы, допустимые в идентификаторах языка «1С:Предприятие».
- Символ **«/»**;
- Символы **«{ }»** с непустым текстом между ними;
- Символ *****.

Например, шаблон может выглядеть следующим образом:

```
/query
/documents/{id}/props/{PropertyName}/*
```

[Копировать в буфер обмена](#)

При описании шаблона будут использоваться следующие термины:

- **Сегмент** – часть URL между двумя символами **«/»**. Например, **props**, из примера выше, является сегментом.
- **Параметризованный сегмент** – сегмент, заключенный в фигурные скобки (**«{сегмент}»**). Фактически, параметризованный сегмент описывает переменную с указанным именем, к которой можно получить доступ из встроенного языка.

Сегмент (не параметризованный) из шаблона должен быть перенесен в URL дословно. На месте параметризованного сегмента может располагаться любое значение, допустимое в URL. Символ **«*»** может находиться только в конце шаблона. На месте символа **«*»** может находиться любое количество сегментов, включая отсутствие сегмента. В одном шаблоне не может встречаться параметризованных сегментов с одинаковым именем. Имя параметризованного сегмента может состоять только из букв, цифр и символа **«_»**.

Шаблон вида **/documents/*number** не является допустимым. Если задан шаблон вида **/documents/{kind}/{number}**, то являются допустимыми следующие URL сервиса:

```
/documents/trade/13
/documents/12/test
```

[Копировать в буфер обмена](#)

При получении запроса система пытается определить, какой шаблон следует использовать. Для этого выполняется попытка сопоставить текст входящего запроса с заданными шаблонами. Сопоставление выполняется по следующим правилам:

1. Все строки сравниваются регистрозависимо;
2. Точное совпадение всегда более приоритетно, чем соответствие шаблону с параметризованными сегментами или символом **«*»**;
3. Символ **«*»** сопоставляется с любым количеством сегментов (включая отсутствие сегмента);
4. Параметризованный сегмент имеет приоритет над символом **«*»**;
5. Если переданному запросу не удалось однозначно сопоставить ни один шаблон – будет сгенерирована ошибка.

Данные правила поясняют следующие примеры:

1. При сопоставлении URL **/Накладная** с шаблонами **/Накладная/** и **/{EntityName}/** для использования будет выбран первый шаблон;
2. Шаблон **/Document/*/{Num}** не является допустимым;
3. URL **/Накладная/500/Строки/2/Товар** может быть сопоставлен шаблону **/Накладная/***;

4. При сопоставлении URL `/Накладная/500` с шаблонами `/Document/{Num}` и `/Document/*`, для использования будет выбран первый шаблон;

5. При сопоставлении URL `/Накладная/500/` шаблонам `{Document}/{Num}` и `/Document/{Number}` произойдет ошибка, т. к. не удалось однозначно выбрать шаблон.

18.2.2.2.3. Редактирование метода

Для созданного шаблона необходимо определить HTTP-методы, которые могут быть использованы при работе с ресурсом, совпадающим с шаблоном. Для этого необходимо создать нужное количество объектов `Метод`, подчиненных объекту `Шаблон URL`. В каждом методе необходимо указать свойство `HTTP-метод` и создать обработчик. Имя обработчика автоматически формируется из имени шаблона URL и собственно имени метода. Так, если в шаблоне URL указано имя `ПримерДляДокументации`, и для метода указано имя `GET`, то для метода встроеного языка, с помощью которого будет обрабатываться HTTP-запрос, использовано имя обработчика `ПримерДляДокументацииGET()`.

COBET. Рекомендуется называть объекты `HTTP-метод` именем обслуживаемого метода на английском языке.

Имеется возможность создать обработчики для следующих методов:

- GET;
- POST;
- PUT;
- DELETE;
- PATCH;
- MERGE;
- CONNECT;
- OPTIONS;
- TRACE;
- PROPFIND;
- PROPPATCH;
- MKCOL;
- COPY;
- MOVE;
- LOCK;
- UNLOCK.

Если создается обработчик для метода `Любой`, то обработчик этого метода будет вызван для всех HTTP-методов (из приведенного выше списка), которые явно не описаны в том шаблоне, который «обрабатывает» входящий HTTP-запрос.

После получения необходимого шаблона, будет предпринята попытка вызвать метод, обслуживающий нужный HTTP-запрос (GET, POST и т. д.). Если обработчик для необходимого запроса не был обнаружен, вызывающей стороне будет возвращена ошибка 405. Если в процессе исполнения метода обнаружена ошибка встроеного языка – вызывающей стороне будет возвращена ошибка 500, а в тело ответа будет помещен текст исключения.

При формировании ответа параметр `Content-Length` может быть не задан. В этом случае он будет определен автоматически.

18.2.2.2.4. Входные и выходные данные обработчика метода

Обработчиком того или иного HTTP-метода является функция, которая получает на вход один параметр типа `HTTPСервисЗапрос` и должна вернуть объект типа `HTTPСервисОтвет`.

Входной параметр полностью описывает запрос, поступивший в систему. С помощью объекта можно определить:

- Какой HTTP-метод используется в запросе (свойство `HTTPМетод`);
- На какой URL поступил запрос (свойство `БазовыйURL`) – часть полного URL запроса, не включающая относительный URL (см. [здесь](#)) и параметры запроса;
- Относительный URL (свойство `ОтносительныйURL`) – часть полного URL запроса, которая использовалась для определения группы ресурсов и правил обработки входящего HTTP-запроса;
- Какие заголовки (свойство `Заголовки`) содержит HTTP-запроса;
- Какие параметризованные сегменты (свойство `ПараметрыURL`) выделены из входящего HTTP-запроса и их значения.
- Какие параметры (свойство `ПараметрыЗапроса`) перечислены в URL HTTP-запроса после символа "?" и их значения.

Также существует возможность получать тело запроса в виде двоичных данных или строки (в зависимости от передаваемой информации).

Таким образом, прикладному разработчику не требуется самостоятельно «разбираться» с входящим запросом. С помощью примера, приведенного далее в этой главе (см. [здесь](#)), можно посмотреть, каким образом входящий HTTP-запрос распределяется по свойствам объекта `HTTPСервисЗапрос`.

После того, как прикладное решение выполнило все действия, которые определены для данного ресурса и HTTP-метода, необходимо сформировать ответ сервиса. Для этого необходимо создать объект типа `HTTPСервисОтвет`. В зависимости от результата обработки, следует указать свойство `КодСостояния`, которое описывает стандартный код возврата HTTP (<http://tools.ietf.org/html/rfc2616#section-10>, на английском языке). Затем, при необходимости, следует сформировать тело ответа HTTP-сервиса. Тело может быть установлено как двоичные данные, как строка или с помощью указания имени файла, откуда будет загружено тело ответа. Если тип возвращаемого функцией (которая реализует реакцию на HTTP-метод) значения отличается от `HTTPСервисОтвет`, то клиенту будет возвращена 500 ошибка.

18.2.2.3. Примеры реализации HTTP-сервисов

ПРИМЕЧАНИЕ. Примеры, приведенные ниже, не являются законченным. Они предназначены для демонстрации некоторых возможностей HTTP-сервисов.

18.2.2.3.1. Отображение входящего запроса

В качестве примера реализуем HTTP-сервис, который будет возвращать в качестве ответа все параметры входящего запроса.

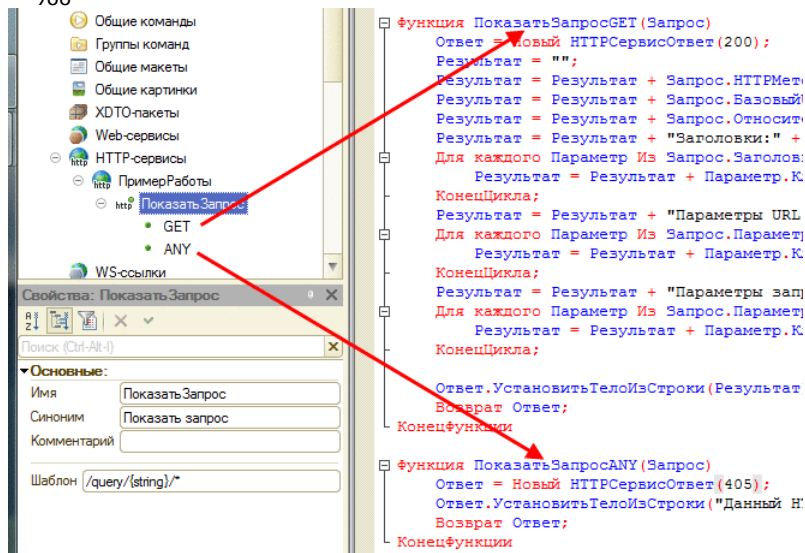


Рис. 335. Тестовый пример

Для этого нужно создать HTTP-сервис **ПримерРаботы**, указав для него корневой URL **example**. Затем следует для сервиса создать шаблон URL **ПоказатьЗапрос** с шаблоном `/query/{string}/*`. Теперь необходимо в шаблон **ПоказатьЗапрос** добавить следующие методы:

- Метод **GET** для HTTP-запроса **GET**;
- Метод **ANY** для HTTP-запроса **Любой**.

Обработчики должны выглядеть следующим образом:

```

Функция ПоказатьЗапросGET(Запрос)
    Ответ = Новый HTTPСервисОтвет(200);
    Результат = "";
    Результат = Результат + Запрос.HTTPМетод + Символы.ПС;
    Результат = Результат + Запрос.ВазовыйURL + Символы.ПС;
    Результат = Результат + Запрос.ОтносительныйURL + Символы.ПС;
    Результат = Результат + "Заголовки:" + Символы.ПС;
    Для каждого Параметр Из Запрос.Заголовки Цикл
        Результат = Результат + Параметр.Ключ + ":" + Параметр.Значение + Символы.ПС;
    КонечЦикла;
    Результат = Результат + "Параметры URL:" + Символы.ПС;
    Для каждого Параметр Из Запрос.ПараметрыURL Цикл
        Результат = Результат + Параметр.Ключ + ":" + Параметр.Значение + Символы.ПС;
    КонечЦикла;
    Результат = Результат + "Параметры запроса:" + Символы.ПС;
    Для каждого Параметр Из Запрос.ПараметрыЗапроса Цикл
        Результат = Результат + Параметр.Ключ + ":" + Параметр.Значение + Символы.ПС;
    КонечЦикла;
    Ответ.УстановитьТелоИзСтроки(Результат);
    Возврат Ответ;
КонечФункции
Функция ПоказатьЗапросANY(Запрос)
    Ответ = Новый HTTPСервисОтвет(405);
    Ответ.УстановитьТелоИзСтроки("Данный HTTP-сервис не поддерживает метод " + Запрос.HTTPМетод);
    Возврат Ответ;
КонечФункции

```

[Копировать в буфер обмена](#)

После публикации HTTP-сервисов (описано в книге «1С:Предприятие 8.3. Руководство администратора») сервис должен возвращать полное «описание» поступившего на вход запроса.

При использовании следующего запроса: `http://localhost/httpservice/hs/example/query/value/another?p1=1&p2=2` будет получен следующий ответ (значение некоторых заголовков зависит от используемого веб-браузера):

```

GET
http://localhost/httpservice/hs/example
/query/value/another
Заголовки:
Accept-Encoding:gzip,deflate,sdch
Connection:keep-alive
Accept-Language:ru-RU,ru;q=0.8,en-US;q=0.6,en;q=0.4
DNT:1
Accept:text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Content-Length:0
User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.107 Safari/537.36
Host:localhost
Параметры URL:
*/another
string:value
Параметры запроса:
p1:1
p2:2

```

[Копировать в буфер обмена](#)

Рассмотрим, каким образом был разобран запрос. В примере было указано, что корневой URL сервиса – **example**, шаблон URL – `/query/{string}/*` и был сформирован обработчик HTTP-запроса GET.

В результате видно, что система определила следующие параметры URL:

- Параметр `*` со значением `/another`;
- Параметр `string` со значением `value`.

Это полностью соответствует указанному шаблону URL. Если шаблон URL установить в значение `/query/{string}`, то демонстрационный запрос закончится сообщением об ошибке 404, т. к. такой шаблон не будет допускать наличие в URL информации, после параметризованного сегмента.

Также в обработчике метода будут переданы все параметры запроса, которые были указаны в URL.

Любой другой вид запроса (отличный от GET) будет приводить к ошибке 405 с указанием, какой именно запрос не поддерживается сервисом.

18.2.2.3.2. Работа с документами

Данный пример является некоторым аналогом Web-сервиса, который должен по переданному номеру расходной накладной возвращать состав ее табличной части (см. [здесь](#)).

Данный пример будет предоставлять возможность получить список «документов» и получить «документ» по номеру. Также будет реализована реакция на неверный номер «документа». Для упрощения примера реальных объектов типа `Документ` в примере создано не будет.

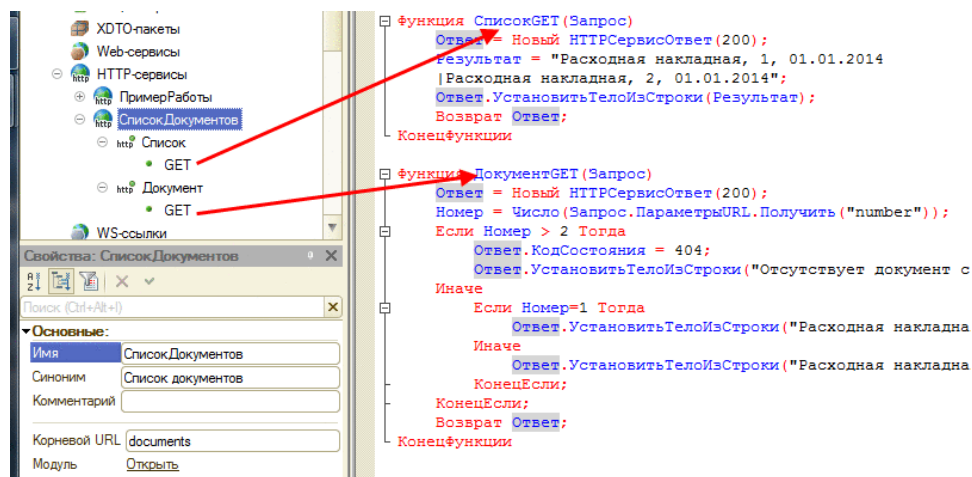


Рис. 336. Тестовый пример

Для этого нужно создать HTTP-сервис `СписокДокументов`, указав для него корневой URL `documents`. Затем следует для сервиса создать шаблон URL `Список` с шаблоном `/*` и шаблон URL `Документ` с шаблоном `/ {number}`. Для каждого шаблона URL следует добавить реализацию метода GET.

Обработчики должны выглядеть следующим образом:

```

Функция СписокGET (Запрос)
    Ответ = Новый HTTPСервисОтвет(200);
    Результат = "Расходная накладная, 1, 01.01.2014
    |Расходная накладная, 2, 01.01.2014";
    Ответ.УстановитьТелоИзСтроки(Результат);
    Возврат Ответ;
КонецФункции
Функция ДокументGET (Запрос)
    Ответ = Новый HTTPСервисОтвет(200);
    Номер = Число(Запрос.ПараметрыURL.Получить("number"));
    Если Номер > 2 Тогда
        Ответ.КодСостояния = 404;
        Ответ.УстановитьТелоИзСтроки("Отсутствует документ с номером: " + Номер);
    Иначе
        Если Номер=1 Тогда
            Ответ.УстановитьТелоИзСтроки("Расходная накладная, 1, 01.01.2014");
        Иначе
            Ответ.УстановитьТелоИзСтроки("Расходная накладная, 2, 01.01.2014");
        КонецЕсли;
    КонецЕсли;
    Возврат Ответ;
КонецФункции

```

После публикации HTTP-сервисов (описано в книге «1С:Предприятие 8.3. Руководство администратора») сервис должен возвращать полное «описание» поступившего на вход запроса.

Рассмотрим, каким образом работает созданный сервис:

- Запрос `http://localhost/httpservice/hs/documents` вернет список «документов»:

```

Расходная накладная, 1, 01.01.2014
Расходная накладная, 2, 01.01.2014

```

- Запрос вида `http://localhost/httpservice/hs/documents/get/list` вернет список «документов», т. к. этот запрос соответствует шаблону `Список` и не соответствует шаблону `Документ`.
- Запрос вида `http://localhost/httpservice/hs/documents/list` вернет ошибку времени исполнения, т. к. приведенный запрос соответствует шаблону `Документ`, но в функции `ДокументGET()` нет проверки на то, что переданный номер является числом.
- Запрос вида `http://localhost/httpservice/hs/documents/2` вернет «документ» с номером 2:

```

Расходная накладная, 2, 01.01.2014

```

- Запрос вида `http://localhost/httpservice/hs/documents/3` вернет ошибку:

```

Отсутствует документ с номером: 3

```

Обработка корректности номера документа явно реализована в обработчике метода.

18.2.2.4. Установка значений разделителей

Значения разделителей можно устанавливать только следующими способами:

- С помощью фрагментов URL при обращении к REST-сервису;
- С помощью файла описания публикации `default.vrd`.

Указанием разделителей с помощью параметра командной строки `Z` не поддерживается.

18.2.2.5. Публикация HTTP-сервиса

Публикация HTTP-сервисов выполняется с помощью диалога публикации на веб-сервере ([Администрирование – Публикация на веб-сервере](#)) и описано в книге «1С:Предприятие 8.3. Руководство администратора».

Глава 19. Механизм заданий

Механизм заданий предназначен для выполнения какой-либо прикладной функциональности по расписанию или асинхронно.

Механизм заданий решает следующие задачи:

- возможность определения регламентных процедур на этапе конфигурирования системы;
- выполнение заданных действий по расписанию;
- выполнение вызова заданной процедуры или функции асинхронно, т. е. без ожидания ее завершения;
- отслеживание хода выполнения определенного задания и получение его статуса завершения (значения, указывающего успешность или неуспешность его выполнения);
- получение списка текущих заданий;
- возможность ожидания завершения одного или нескольких заданий;
- управление заданиями (возможность отмены, блокировка выполнения и др.).

Механизм заданий состоит из следующих компонентов:

- метаданных регламентных заданий,
- регламентных заданий,
- фоновых заданий,
- планировщика заданий.

Регламентные задания предназначены для выполнения прикладных задач по расписанию. Регламентные задания хранятся в информационной базе и создаются на основе метаданных, определяемых в конфигурации. Метаданные регламентного задания содержат такую информацию, как наименование, метод, использование и т. д.

Регламентное задание имеет **расписание**, которое определяет, в какие моменты времени нужно выполнять связанный с регламентным заданием метод. Расписание, как правило, задается в информационной базе, но может быть задано и на этапе конфигурирования (например, для предопределенных регламентных заданий).

Фоновые задания предназначены для выполнения прикладных задач асинхронно, реализуются средствами встроеного языка.

Планировщик заданий используется для планирования выполнения регламентных заданий. Для каждого регламентного задания планировщик периодически проверяет, соответствует ли текущая дата и время расписанию регламентного задания. Если соответствует, планировщик назначает такое задание на выполнение. Для этого по данному регламентному заданию планировщик создает фоновое задание, которое и выполняет реальную обработку.

19.1. Фоновые задания

Фоновые задания удобно использовать для выполнения сложных вычислений, когда результат вычисления может быть получен через продолжительное время. Механизм заданий имеет средства для выполнения таких вычислений асинхронно.

С фоновым заданием связан метод, который вызывается при выполнении фонового задания. В качестве метода фонового задания может выступать любая процедура или функция неглобального общего модуля, которую можно вызвать на сервере. Параметрами фонового задания могут быть любые значения, которые разрешено передавать на сервер. Параметры фонового задания должны в точности соответствовать

параметрам той процедуры или функции, которую оно вызывает. Если методом фонового задания является функция, то ее возвращаемое значение игнорируется.

Фоновое задание может иметь **ключ** – любое прикладное значение. Ключ вводит ограничение на запуск фоновых заданий – в единицу времени может выполняться только одно фоновое задание с определенным значением ключа и заданным именем метода фонового задания (имя метода состоит из имени модуля и имени процедуры или функции). Ключ позволяет группировать фоновые задания, имеющие одинаковые методы, по определенному прикладному признаку с тем, чтобы в рамках одной группы выполнялось не более одного фонового задания.

Создание и управление фоновыми заданиями выполняются программно из любого соединения. Создавать фоновое задание разрешено любому пользователю. При этом оно выполняется от имени того пользователя, который его создал. Получать задания, а также ожидать их завершения разрешено из любого соединения пользователю с административными правами либо пользователю, который создал эти фоновые задания.

Фоновое задание является чисто сеансовым объектом, но не принадлежит какому-либо пользовательскому сеансу. Для каждого задания создается специальный системный сеанс, выполняющийся от имени того пользователя, который выполнил вызов. Фоновые задания не имеют сохраняемого состояния.

Фоновое задание может порождать другие фоновые задания. В клиент-серверном варианте это позволяет распараллеливать сложные вычисления по рабочим процессам кластера, что может значительно ускорить процесс вычисления в целом. Распараллеливание реализуется порождением нескольких дочерних фоновых заданий с ожиданием завершения каждого из них в основном фоновом задании.

Фоновое задание имеет возможность помещать данные во временное хранилище вызывающего сеанса (см. [здесь](#)). Это может быть использовано, например, для передачи в вызывающий сеанс сформированного отчета или подготовленных для обработки данных. Передача данных из вызывающего сеанса в сеанс фонового задания невозможна.

Запуск фонового задания выполняется следующим образом:

- Выполняются служебные действия:
 - Фиксируется факт запуска фонового задания в журнале регистрации (см. [здесь](#));
 - Устанавливаются параметры соединения с внешними источниками данных (см. [здесь](#));
 - Устанавливается режим интерфейса и т. д.
- Выполняется обработчик события `УстановкаПараметровСеанса`;
- Выполняется собственно фоновое задание.

Завершившиеся успешно или аварийно фоновые задания хранятся в течение суток, а потом удаляются. Если количество выполнившихся фоновых заданий превышает 1000, то наиболее старые фоновые задания также удаляются. Для файлового варианта информационной базы это количество относится к одной базе, а для клиент-серверного варианта – к одному кластеру.

19.2. Регламентные задания

Регламентные задания используются, когда необходимо выполнить определенные периодические или однократные действия в соответствии с расписанием.

Регламентные задания хранятся в информационной базе и создаются на основе метаданных регламентного задания, определенных в конфигурации. Метаданные задают такие параметры регламентного задания, как вызываемый метод, наименование, ключ, возможность использования, признак предопределенности и др. При создании регламентного задания дополнительно можно указать расписание (может быть указано в метаданных), значения параметров метода, имя пользователя, от имени которого должно выполняться регламентное задание, и др.

Создание и управление регламентными заданиями выполняются программно из любого соединения и

разрешены только пользователям, имеющим административные права.

ПРИМЕЧАНИЕ. При работе в файловом варианте создавать и редактировать регламентные задания возможно без запуска планировщика заданий.

С регламентным заданием связан метод, который вызывается при выполнении регламентного задания. В качестве метода регламентного задания может выступать любая процедура или функция неглобального общего модуля, которую можно вызвать на сервере. Параметрами регламентного задания могут быть любые значения, которые разрешено передавать на сервер. Параметры регламентного задания должны в точности соответствовать параметрам той процедуры или функции, которую оно вызывает. Если методом регламентного задания является функция, то ее возвращаемое значение игнорируется.

Регламентное задание может иметь **ключ** – любое прикладное значение. Ключ вводит ограничение на запуск регламентных заданий, т. к. в единицу времени среди регламентных заданий, связанных с одним и тем же объектом метаданных, может выполняться только одно регламентное задание с определенным значением ключа. Ключ позволяет группировать регламентные задания, связанные с одним и тем же объектом метаданных, по определенному прикладному признаку с тем, чтобы в рамках одной группы выполнялось не более одного регламентного задания.

При конфигурировании можно определить **предопределенные** регламентные задания. Предопределенные регламентные задания ничем не отличаются от обычных регламентных заданий, за исключением того, что их нельзя явно создавать и удалять. Если в метаданных регламентного задания установлен признак предопределенного регламентного задания, то при обновлении конфигурации в информационной базе автоматически будет создано предопределенное регламентное задание. Если признак предопределенности снят, то при обновлении конфигурации в информационной базе автоматически будет удалено предопределенное регламентное задание. Начальные значения свойств предопределенного регламентного задания (например, расписание) устанавливаются в метаданных. В дальнейшем при работе приложения их можно менять. Предопределенные регламентные задания не имеют параметров.

Расписание регламентного задания определяет, в какие моменты времени регламентное задание должно быть запущено. Расписание позволяет:

- задавать дату и время начала и окончания выполнения задания;
- период выполнения;
- дни недели и месяцы, по которым нужно выполнять регламентное задание и др. (см. описание встроеного языка).

Примеры расписаний регламентных заданий:

Расписание	Значения параметров
Каждый час, только один день	ПериодПовтораДней = 0 ПериодПовтораВТечениеДня = 3600
Каждый день один раз в день	ПериодПовтораДней = 1 ПериодПовтораВТечениеДня = 0
Один день, один раз	ПериодПовтораДней = 0
Через день один раз в день	ПериодПовтораДней = 2
Каждый час с 01.00 до 07.00 каждый день	ПериодПовтораДней = 1 ПериодПовтораВТечениеДня = 3600 ВремяНачала = 01.00 ВремяКонца = 07.00

Каждую субботу и воскресенье в 09.00	ПериодПовтораДней = 1 ДниНедели = 6, 7 ВремяНачала = 09.00
Каждый день одну неделю, неделя пропуска	ПериодПовтораДней = 1 ПериодНедель = 2
В 01.00 один раз	ВремяНачала = 01.00
Последнее число каждого месяца в 9.00	ПериодПовтораДней = 1 ДеньВМесяце = -1 ВремяНачала = 09.00
Пятое число каждого месяца в 9.00	ПериодПовтораДней = 1 ДеньВМесяце = 5 ВремяНачала = 09.00
Вторая среда каждого месяца в 9.00	ПериодПовтораДней = 1 ДеньНеделиВМесяце = 2 ДниНедели = 3 ВремяНачала = 09.00

Можно проверять, выполняется ли задание для заданной даты (метод `ТребуетсяВыполнение()` объекта `РасписаниеРегламентныхЗаданий`). Регламентные задания всегда выполняются от имени определенного пользователя. Если пользователь регламентного задания не указан, то выполнение происходит с правами, которые определяются набором ролей, указанных в свойстве конфигурации `ОсновныеРоли`. В том случае, если в этом свойстве не указано ни одной роли, выполнение происходит без ограничения прав доступа.

Выполнение регламентных заданий происходит с использованием фоновых заданий. Когда планировщик определяет, что регламентное задание должно быть запущено, то автоматически на основе данного регламентного задания создается фоновое задание, которое и выполняет всю дальнейшую обработку. Если данное регламентное задание уже выполняется, то оно не будет запущено повторно, вне зависимости от его расписания.

Регламентные задания имеют возможность перезапуска. Это особенно актуально, когда метод регламентного задания должен быть гарантированно выполнен. Перезапуск регламентного задания осуществляется в том случае, когда оно завершено аварийно, либо когда рабочий процесс (в клиент-серверном варианте) или клиентский процесс (в файловом варианте), на котором выполнялось регламентное задание, завершён аварийно. В регламентном задании можно указать, сколько раз нужно его перезапускать (свойство `Количество повторов при аварийном завершении`), а также интервал между перезапусками (свойство `Интервал повтора при аварийном завершении`). После выполнения заданного количества перезапусков, попытки запуска прекращаются до момента, когда наступит очередное время старта регламентного задания (в соответствии с расписанием). При этом счетчик количества повторов сбрасывается и при аварийном завершении регламентного задания, процесс перезапуска начинается сначала.

При реализации метода перезапускаемого регламентного задания нужно учитывать, что при перезапуске его выполнение будет начато с начала, а не продолжено с момента аварийного завершения.

19.3. Особенности выполнения фоновых заданий в файловом и клиент-серверном вариантах

Механизмы выполнения фоновых заданий в файловом и клиент-серверном вариантах различаются.

19.3.1.1. Файловый вариант

Фоновые и регламентные задания выполняются клиентскими приложениями или расширением веб-сервера. Фоновые задания выполняются на том клиентском приложении, которое инициировало запуск фонового задания. Фоновые задания исполняются последовательно, т. е. на одном клиентском приложении не может одновременно выполняться два фоновых задания. В случае работы через веб-сервер, для каждой информационной базы, доступ к которой осуществляется через этот веб-сервер, организовано последовательное выполнение фоновых и регламентных заданий.

Поведение фоновых и регламентных заданий в файловом варианте имеет следующие особенности:

- Информация о фоновых заданиях, вызванных с помощью методов языка или выполняющих отчеты, доступна только в том клиентском приложении, которое их выполняло. Информация не сохраняется после завершения работы клиентского приложения.
- Информация о фоновых заданиях, инициированных регламентными заданиями, доступна на всех клиентских приложениях и сохраняется между запусками.
- Регламентные задания выполняются только одним клиентским приложением. Имеется возможность запретить запуск регламентных заданий определенным клиентским приложением или указать, что какое-либо клиентское приложение принудительно назначается исполнителем регламентных заданий. Такое управление осуществляется с помощью:
 - Параметра командной строки `/AllowExecuteScheduledJobs` для толстого и тонкого клиентских приложений.
 - Атрибута `allowexecutescheduledjobs` элемента `point` файла публикации `default.vrd` (см. книгу «1С:Предприятие 8.3. Руководство администратора») в случае публикации информационной базы на веб-сервере.
- Регламентные задания выполняются первым, по порядку запуска, клиентским приложением, у которого не указан запрет выполнения регламентных заданий. После завершения сеанса этого клиентского приложения, выполнение переходит к какому-либо из оставшихся запущенными клиентских приложений. Если запускается клиентское приложение с явным указанием на необходимость исполнения регламентных заданий, то регламентные задания начинают выполняться этим клиентским приложением, независимо от наличия других клиентских приложений (включая расширения веб-сервера).
- Регламентные задания выполняются расширением веб-сервера до тех пор, пока расширение веб-сервера обслуживает хотя бы один клиентский сеанс.
- Регламентные задания обрабатываются один раз в 60 секунд.
- Регламентные задания начинают выполняться не ранее, чем через 1 минуту после старта клиентского приложения.
- Выполняется принудительное завершение фоновых заданий, которые не завершились в течении 30 секунд после завершения работы клиентского приложения. В журнале регистрации в этом случае фиксируется событие **Фоновое задание. Принудительное завершение** (`_Job_.Terminate`).

19.3.1.2. Клиент-серверный вариант

В клиент-серверном варианте для выполнения фоновых заданий используется **планировщик заданий**, который физически находится в менеджере кластера. Планировщик получает наименее загруженный рабочий процесс для всех поставленных в очередь на выполнение фоновых заданий и использует его для выполнения соответствующего фонового задания. Рабочий процесс выполняет задание и уведомляет планировщик о результатах выполнения.

В клиент-серверном варианте имеется возможность блокирования выполнения регламентных заданий. Блокирование выполнения регламентных заданий происходит в следующих случаях:

- На информационную базу установлена явная блокировка регламентных заданий. Блокировка может быть установлена через консоль кластера.
- На информационную базу установлена блокировка начала сеансов. Блокировка может быть установлена через консоль кластера.
- Из встроенного языка вызван метод `УстановитьМонопольныйРежим()` с параметром `Истина`.
- В некоторых других случаях (например, при обновлении конфигурации базы данных).

19.4. Создание метаданных регламентного задания

Перед тем как программно создать регламентное задание в информационной базе, необходимо создать для него объект метаданных.

Для создания объекта метаданных регламентного задания в ветке **Общие** дерева конфигурации для ветки **Регламентные задания** нужно выполнить команду **Добавить**.

Регламентное задание обладает рядом свойств, которые описаны ниже.

Имя метода – указывается имя метода регламентного задания.

Ключ – указывается произвольное строковое значение, которое будет использовано в качестве ключа регламентного задания.

Расписание – указывается расписание регламентного задания. Для формирования расписания следует перейти по ссылке **Открыть** и в открывшейся форме расписания установить нужные значения.

На закладке **Общее** указывается дата начала и завершения задания и режим повтора.

Рис. 337. Общее расписание

На закладке **Дневное** указывается дневное расписание задания.

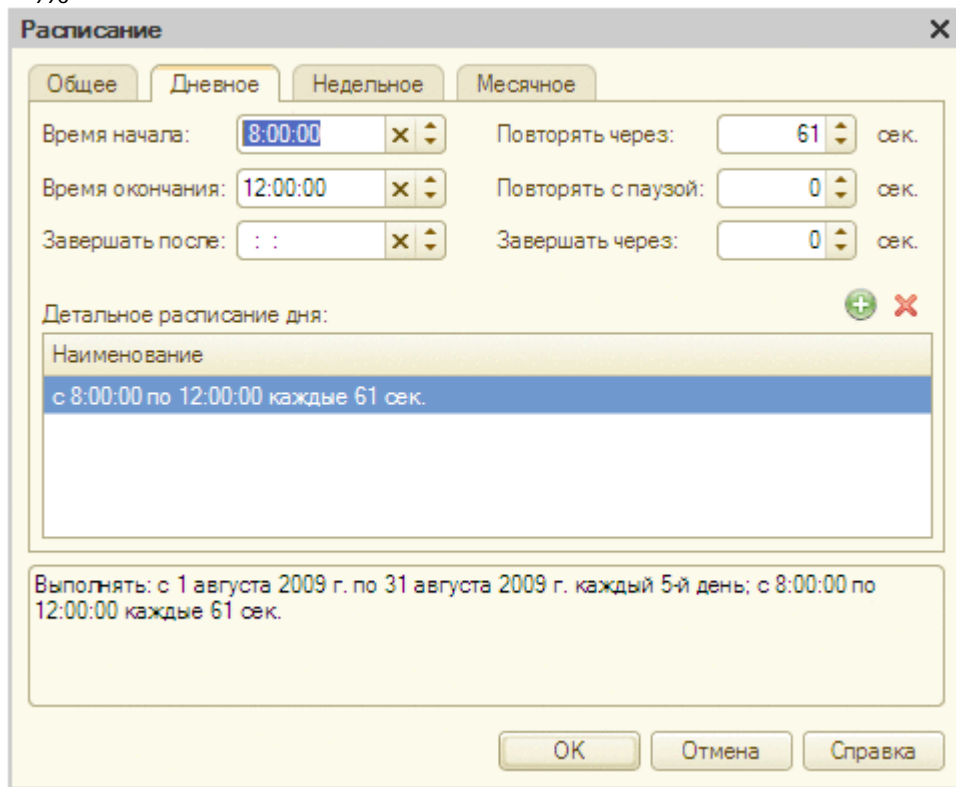


Рис. 338. Дневное расписание

В расписании можно указать:

- время начала и окончания задания;
- время завершения задания, после которого оно будет принудительно завершено;
- период повтора задания;
- продолжительность паузы между повторами;
- продолжительность выполнения.

Допускается указание произвольного сочетания условий.

На закладке [Недельное](#) указывается недельное расписание задания.

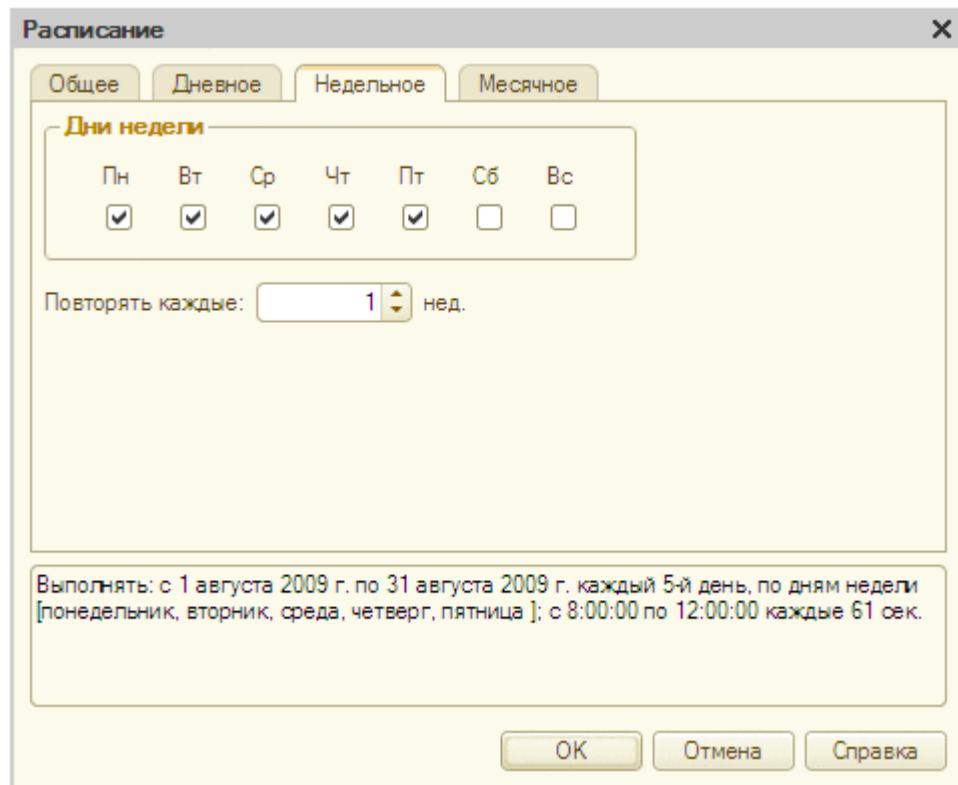


Рис. 339. Недельное расписание

Нужно установить флажки по тем дням недели, по которым задание будет выполняться. Если требуется повторять задание, указать интервал повтора в неделях. Например, задание выполняется через 2 недели значение повтора – 2.

На закладке [Месячное](#) указывается месячное расписание задания.

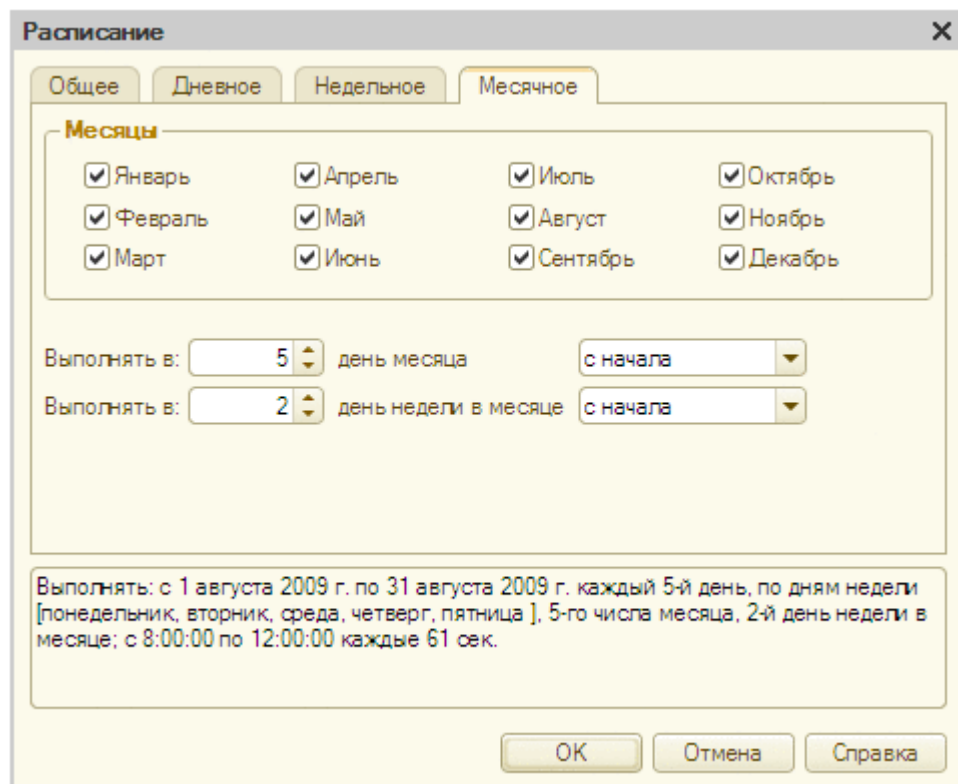


Рис. 340. Месячное расписание

Нужно установить флажки по тем месяцам, в которых задание будет выполняться. При необходимости можно указать конкретный день (месяца или недели) выполнения с начала месяца/недели или конца.

[Использование](#) – если установлено, то задание будет выполняться согласно расписанию.

Предопределенное – если установлено, то задание является предопределенным.

Количество повторов при аварийном завершении – указывает количество повторов при аварийном завершении.

Интервал повтора при аварийном завершении – указывает интервал повтора при аварийном завершении.

Пример создания фонового задания «Обновление индекса полнотекстового поиска»:

[Копировать в буфер обмена](#)

```
ФоновыеЗадания.Выполнить ("ОбновлениеИндексаПолнотекстовогоПоиска");
```

Пример создания регламентного задания «Восстановление последовательностей»:

[Копировать в буфер обмена](#)

```
Расписание = Новый РасписаниеРегламентногоЗадания;  
Расписание.ПериодПовтораДней = 1;  
Расписание.ПериодПовтораВТечениеДня = 0;  
Задание = РегламентныеЗадания.СоздатьРегламентноеЗадание ("ВосстановлениеПоследовательностей");  
Задание.Расписание = Расписание;  
Задание.Записать();
```

Глава 2. Работа с конфигурацией

2.1. Общая информация

В этой главе будут рассмотрены общие приемы работы с объектами конфигурации, которые применимы к объектам любых типов. Особенности создания и редактирования объектов конфигурации различных типов рассматриваются далее в соответствующих главах Руководства.

В данной главе в основном рассматриваются визуальные средства управления объектами конфигурации, имеющиеся в конфигураторе. Описание встроенного языка представлено в справке по встроенному языку.

О сохранении и восстановлении информационной базы данных см. книгу «1С:Предприятие 8.3. Руководство администратора» или «1С:Предприятие 8.3. Клиент-серверный вариант. Руководство администратора».

В информационной базе системы «1С:Предприятие» хранятся две конфигурации:

- **основная конфигурация** (редактируемая, далее просто конфигурация),
- **конфигурация базы данных.**

Конфигурация базы данных определяет структуру таблиц базы данных и всю функциональность, с которой работают пользователи. Основная конфигурация используется только для изменения. Это позволяет вносить и сохранять изменения в конфигурации в процессе работы пользователей. Произвести изменение в конфигурации базы данных в процессе работы пользователей нельзя. О работе с конфигурацией базы данных см. [здесь](#). Все основные команды работы с конфигурацией (как основной, так и конфигурацией базы данных) собраны в меню **Конфигурация**.

Кроме того, в конфигураторе могут быть открыты окна расширений конфигурации (см. [здесь](#)). Внешний вид окна расширения конфигурации практически не отличается от окна редактирования основной конфигурации. Однако, действия, которые пользователь может выполнить с конфигурацией расширения, недоступны из меню **Конфигурация** (кроме открытия списка расширений (см. [здесь](#)) и сохранения конфигурации расширения, см. [здесь](#)). Команды, предназначенные для работы с конфигурацией расширения, собраны в меню **Действия – Конфигурация** окна с расширением конфигурации или в контекстном меню списка расширений (см. [здесь](#)).

Если нет отдельной оговорки, все действия, описанные далее, относятся к основной конфигурации, если они выполняются из меню **Конфигурация** главного меню конфигуратора. Команды, которые доступны при управлении расширениями, относятся к соответствующим расширениям. Отдельно действие таких команд не описывается.

2.2. Открытие конфигурации

Работа с конфигурацией осуществляется в окне **Конфигурация**. Для открытия конфигурации нужно выбрать пункт **Конфигурация – Открыть конфигурацию**. На экран выводится окно **Конфигурация**.

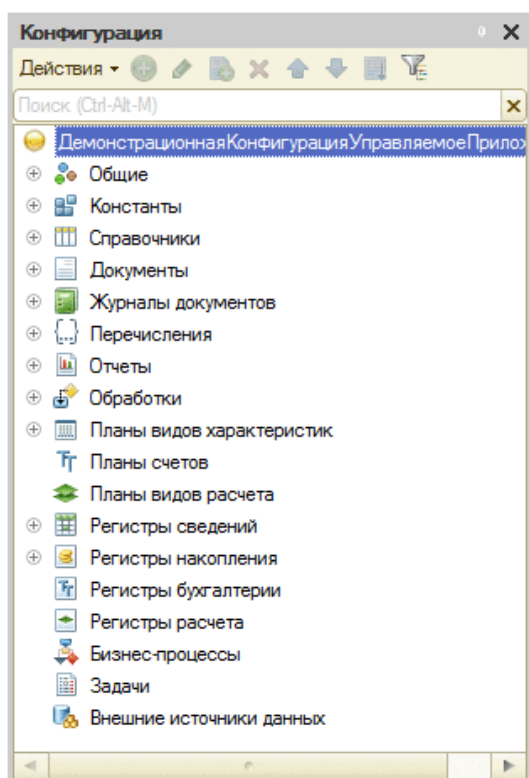


Рис. 2. Конфигурация

Чтобы увеличить рабочую область для работы с различными объектами, окно **Конфигурация** можно временно закрыть.

Закрытие окна [Конфигурация](#) не означает завершения работы с отдельными ее составляющими, открытыми для редактирования: закрывается не конфигурация, а окно конфигурации. Для открытия окна [Конфигурация](#) нужно выбрать пункт [Конфигурация – Окно конфигурации](#).

2.3. Сохранение конфигурации

В процессе редактирования конфигурации могут быть созданы новые, изменены существующие или удалены имеющиеся объекты, подчиненные объекты (формы, реквизиты и т. д.). Любое такое изменение приводит к модифицированности конфигурации. Признак модифицированности конфигурации «*» показывается в заголовке окна [Конфигурация](#).

Для сохранения конфигурации (без завершения работы с конфигурацией) нужно выбрать пункт [Конфигурация – Сохранить конфигурацию](#). Пункт доступен, если конфигурация была изменена, в отличие от редакторов текстового, табличного и HTML-документов, для которых пункт доступен всегда.

Сохранение измененной конфигурации возможно в любой момент времени, даже если запущен режим 1С:Предприятие или производится отладка.

Описанное поведение верно также для окна расширения конфигурации (см. [здесь](#)).

2.4. Закрытие конфигурации

Для закрытия конфигурации нужно выбрать пункт [Конфигурация – Закрыть конфигурацию](#). При этом если конфигурация была модифицирована (были произведены изменения), то конфигуратор выводит сообщение: [Конфигурация 'Синоним конфигурации' была изменена. Сохранить изменения?](#) Для сохранения внесенных изменений выбрать кнопку [Да](#).

Закрытие конфигурации с сохранением произведенных изменений возможно в любой момент времени, даже если запущен режим 1С:Предприятие или производится отладка.

2.5. Сохранение конфигурации в файл

Для сохранения конфигурации в файл на диск нужно выбрать пункт [Конфигурация – Сохранить конфигурацию в файл](#). На экран выводится стандартный диалог выбора файла. Следует выбрать каталог и указать имя файла, в который будет записана конфигурация.

Сохраненный файл конфигурации необходим для операции сравнения и объединения конфигураций (см. [здесь](#)).

ПРИМЕЧАНИЕ. Если конфигурация стоит на поддержке, то в информационной базе всегда хранится конфигурация поставщика.

2.6. Загрузка конфигурации из файла

Для полной замены текущей конфигурации на конфигурацию, сохраненную в файле, нужно выбрать пункт [Конфигурация – Загрузить конфигурацию из файла](#).

2.7. Дерево объектов конфигурации

Конфигурация представляется в виде древовидной структуры, каждая ветвь которой описывает определенную составляющую конфигурации. Объекты конфигурации в дереве конфигурации представлены своими именами. Например, в ветви [Документы](#) располагаются объекты всех документов, используемых в конфигурации, а ветвь [РасходнаяНакладная](#) описывает объект документа [Расходная накладная](#), ветвь [Общие – Роли – все роли](#) (права доступа к информации), используемые для различных видов пользователей, а ветвь [Менеджер](#) – права доступа менеджера отдела продаж и т. д.

Корневые ветви дерева объединяют объекты конфигурации, логически связанные между собой и имеющие общее назначение.

Например, ветвь [Документы](#) объединяет объекты типа [Последовательности](#), [Нумераторы](#) и, собственно, объекты конфигурации типа [Документ](#). Все эти объекты предназначены для организации ввода документов в системе «1С:Предприятие».

Для работы в конфигураторе удобно использовать клавиатуру. Перечень сочетаний клавиш, которые могут использоваться в конфигураторе, можно получить во встроенной справке (раздел называется [Сочетания клавиш \(Конфигуратор\)](#)).

Объекты конфигурации в пределах «своей» группы объектов конфигурации можно расставить в требуемом порядке. В таком же порядке объекты конфигурации будут выдаваться в различных списках.

Для перестановки объекта конфигурации необходимо выделить его в окне [Конфигурация](#) и использовать пункты меню [Действия – Переместить вверх](#) или [Действия – Переместить вниз](#), а также [Действия – Упорядочить список](#).

Для удобства поиска в дереве конфигурации объекта, редактируемого в данный момент (окно редактирования объекта, формы, макеты, модули), следует использовать пункт меню [Правка – Найти в дереве](#). Предварительно нужно сделать активным окно редактирования объекта. После выполнения команды в дереве конфигурации будет выбран объект, данные которого в данный момент редактируются.

Поиск в дереве метаданных можно осуществить несколькими способами:

- Начать набирать на клавиатуре наименования объекта (при этом поиск будет выполняться только в открытых ветках дерева).
- Использовать стандартный механизм поиска (для вызова окна поиска необходимо нажать **Ctrl + F**). В этом случае будет открыто окно **Результаты поиска**, из которого можно перейти к требуемому объекту метаданных.
- С помощью специального поля поиска, расположенного между списком объектов и командной панелью окна конфигурации. Переход в это поле и любого места конфигулятора выполняется с помощью сочетания клавиш **Ctrl+Alt+M**. Поиск начинается после того, как в поле введено более двух непробельных символов. Поиск осуществляется по свойствам объектов конфигурации **Имя**, **Синоним** и **Комментарий**. Поиск выполняется по вхождению строки (см. [рис. 3](#)). В дереве конфигурации остаются только те объекты конфигурации, которые соответствуют набранной подстроке.

При включенном отборе, нажатие клавиши **ESC** очищает поле ввода и отменяет поиск в том случае, если фокус ввода находится в дереве конфигурации или поле поиска. Клавиша **TAB** предназначена для перехода между деревом объектов конфигурации и полем поиска.

Если при установленном отборе пользователь добавляет объект, который не соответствует наложенному отбору – система отменяет наложенный отбор и отображает все дерево конфигурации.

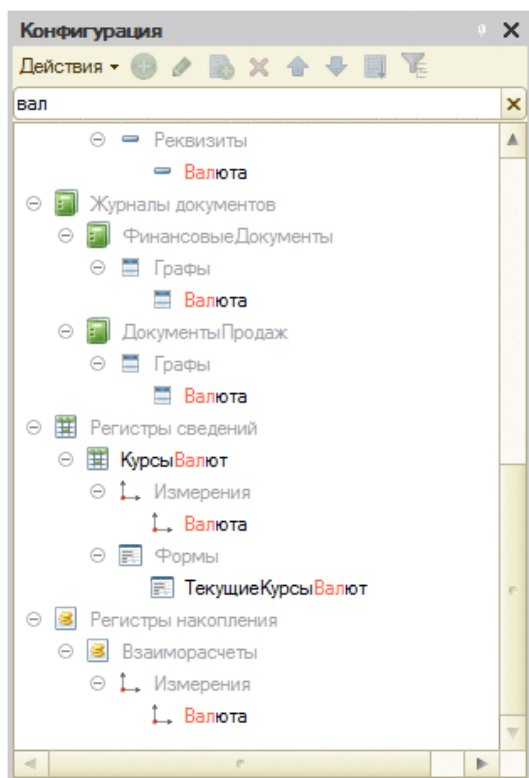


Рис. 3. Поиск в дереве конфигурации

Наименования объектов конфигурации, а также подчиненных объектов (реквизитов, табличных частей, форм и т. д.) можно перетаскивать в модули и текстовые документы с помощью мыши стандартным образом.

2.8. Настройка сортировки дерева метаданных

Данный режим предназначен для сортировки объектов конфигурации. За один раз могут быть отсортированы однотипные объекты конфигурации, подчиненные одному объекту конфигурации.

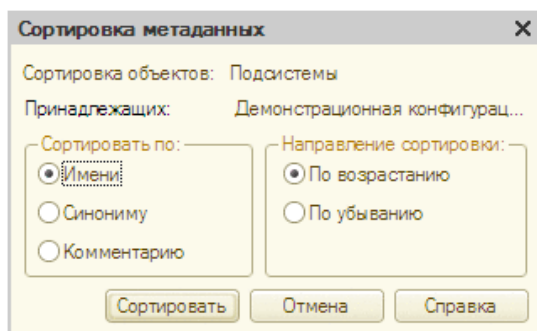


Рис. 4. Сортировка метаданных

Например, формы конкретного справочника.

- **Сортировать по** – выбор свойства, по которому будет выполняться сортировка:
 - **Имени**. Сортировка будет выполнена по именам объектов конфигурации.
 - **Синониму**. Сортировка будет выполнена по синонимам объектов конфигурации.
 - **Комментарию**. Сортировка будет выполнена по комментариям объектов конфигурации.
- **Направление сортировки** – выбор направления сортировки:
 - **По возрастанию**. Будет выполняться сортировка в порядке возрастания.
 - **По убыванию**. Будет выполняться сортировка по убыванию.

2.9. Создание и удаление объекта конфигурации

2.9.1. Создание объекта конфигурации

Управление большинством объектов конфигурации выполняется в окне **Конфигурация**. В этом разделе будут изложены общие приемы создания объектов конфигурации, применимые к объектам конфигурации любых типов.

Для создания нового объекта конфигурации необходимо выполнить следующие действия:

- в дереве конфигурации выделить наименование типа объекта конфигурации или любого из существующих объектов конфигурации того типа, который должен быть у создаваемого объекта;
- выбрать пункт **Действия – Добавить** окна **Конфигурация**.

Создание нового подчиненного объекта можно также производить следующим образом:

- открыть окно редактирования объекта (выбрать пункт **Действия – Изменить** окна **Конфигурация**);
- указать нужный вид подчиненного объекта;
- нажать кнопку **Добавить**.

В результате этих действий на текущей ветви дерева конфигурации появится новый объект, а на экран для редактирования свойств этого объекта будет автоматически вызвана палитра свойств, если палитра еще не была открыта (о работе с палитрой свойств см. [здесь](#)). Для объектов, имеющих широкий набор редактируемых свойств, дополнительно к палитре свойств может вызываться окно редактирования (о работе в окне см. [здесь](#)).

Новому объекту конфигурации присваивается условное имя, состоящее из слова, соответствующего типу создаваемого объекта, и числа – порядкового номера нового объекта конфигурации. Например, для нового справочника имя будет начинаться со слова «**Справочник**». Палитра свойств объекта будет содержать значения свойств, задаваемые по умолчанию.

В конфигураторе контролируются имена объектов перед обновлением конфигурации базы данных и перед формированием поставки.

Для объектов, которые могут иметь подчиненные объекты (например, справочник может иметь реквизиты, табличные части, формы и макеты), производится создание нужного числа и состава подчиненных объектов. Их формирование и настройка производятся с использованием различных средств конфигуратора.

Для облегчения создания некоторых составных частей (запросов, макетов и процедур печати, движений регистров, ввода на основании) объектов в конфигураторе имеются различные конструкторы – вспомогательные инструменты, облегчающие процесс проектирования (подробнее см. [здесь](#)).

Для тех типов объектов, которые могут иметь формы, в системе имеются конструкторы форм – вспомогательные инструменты, облегчающие разработку форм объектов. **Конструктор форм** запускается при создании новой формы (о работе с конструктором форм см. [здесь](#)).

Для редактирования форм предназначен **редактор форм** (о работе с ним см. [здесь](#)).

Редактирование макетов, которые основаны на табличном документе, производится **редактором табличных документов** (см. [здесь](#)). Для формирования макетов можно использовать **конструктор печати** (см. [здесь](#)) и **конструктор выходных форм** (см. [здесь](#)).

Редактирование макетов, которые основаны на текстовом документе, производится **редактором текстовых документов** (см. [здесь](#)).

Программы на встроенном языке, располагающиеся в модулях, создаются с использованием **текстового редактора** (см. [здесь](#)).

Для создания нового объекта можно использовать механизм перетаскивания объектов с помощью мыши. При перетаскивании объекта (как в пределах «своей» ветки, так и за ее пределы) создается новый объект. При этом

производится проверка возможности использования исходных свойств объекта в результирующем. Если результирующий объект поддерживает свойства, то они копируются. Пример успешного копирования: перенос реквизитов, макетов справочника в документ. При копировании того же справочника в объект типа **Стиль** успешно перенесутся только имя, синоним и комментарий.

Если исходный объект содержит подчиненные объекты (реквизиты, формы, макеты и др.), то при перетаскивании на «подобный» уровень (например, перетаскивается объект **Справочник. Валюты** в ветвь **Документы**) скопируются реквизиты, формы, макеты и табличные части. При этом некоторые свойства составных частей могут быть изменены (например, для документа отсутствует свойство **Родитель**).

2.9.2. Удаление объекта конфигурации

Для удаления объекта конфигурации нужно указать его в дереве конфигурации и выбрать пункт **Действия – Удалить** окна **Конфигурация**. Если на объект нет ссылок в других объектах конфигурации, объект удаляется.

Если на объект есть ссылки (объект используется), то выдается сообщение: **Объект не может быть удален, так как на него есть ссылки в других объектах!** А в окне сообщений выводится список использования данного объекта.

На [рис. 5](#) приведен пример окна сообщений при попытке удаления документа **ПриходТовара**.

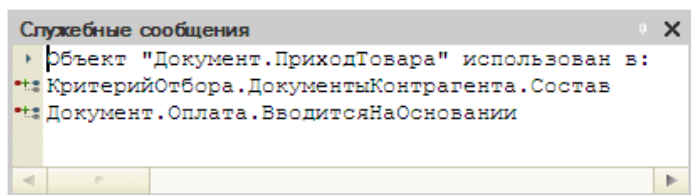


Рис. 5. Службные сообщения

Для перехода к объекту, в котором используется ссылка на удаляемый объект, нужно дважды щелкнуть мышью наименование этого объекта в окне сообщений.

В зависимости от настройки **При удалении выполнять проверку в модулях** (подробнее см. [здесь](#)), при удалении объекта конфигурации будет выполняться проверка использования удаляемого объекта в модулях. Если в результате проверки будет обнаружено использование удаляемого объекта в текстах на встроенном языке, то на экран будет выдан список использований и будет запрошено подтверждение удаления объекта.

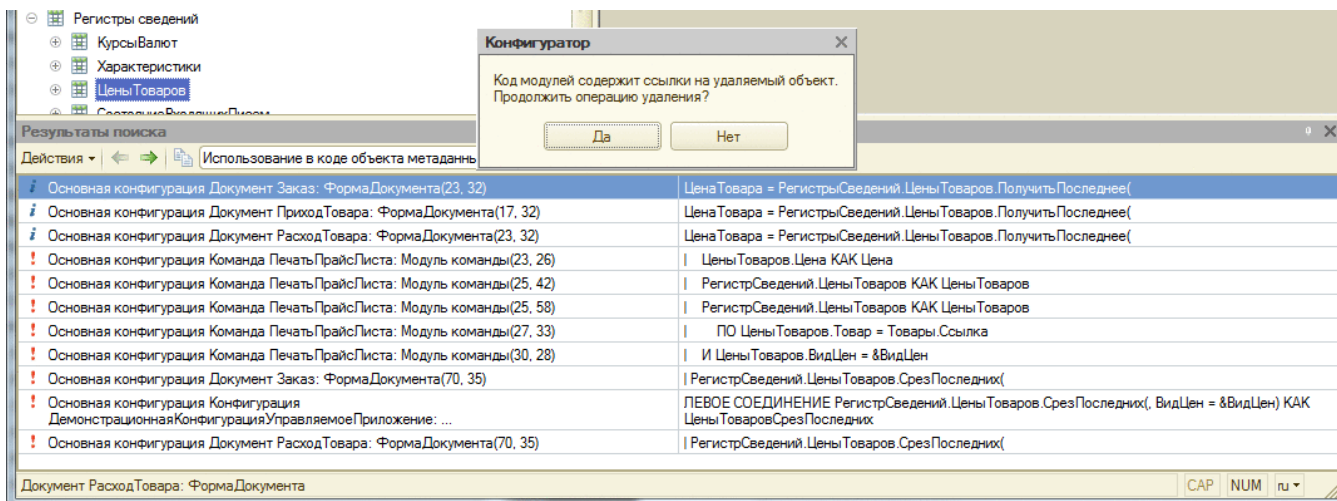


Рис. 6. Проверка использования удаляемого объекта в модулях

2.9.3. Ссылки на объект конфигурации

С точки зрения взаимосвязи объекты конфигурации делятся на несвязанные (например, реквизит **Закупочная цена**, имеющий тип **Число**) и связанные (например, реквизит **Единица измерения** справочника **Номенклатура**, ссылающийся на справочник **Единицы измерения**).

Часто бывает полезно знать, какие объекты конфигурации ссылаются на данный объект, а также выяснить, какие объекты связаны с данным объектом. Для этих целей используются команды **Действия – Поиск ссылок на объект** и **Действия – Поиск ссылок в объекте** окна **Конфигурация**.

На экран выводится диалог (см. [рис. 7](#)).

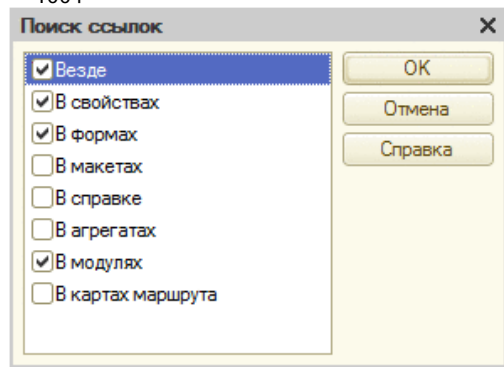


Рис. 7. Ссылки на объект

В нем нужно указать область поиска (свойства, подчиненные объекты) и нажать кнопку [ОК](#).

Если ссылки есть, то при выполнении команды в окно сообщений выводится список ссылок. Если ссылок нет, то на экран выводится одно из предупреждений:

- [Ссылка на данный объект не существует](#),
- [Ссылка в объекте не обнаружено](#).

Команда [Действия – Поиск ссылок в объекте](#) выводит перечень всех агрегатных типов, которые использованы в объекте.

Для перехода по ссылке необходимо дважды щелкнуть мышью соответствующую строку с наименованием объекта-ссылки в окне сообщений.

2.10. Палитра свойств

Процесс редактирования объекта конфигурации заключается в изменении свойств объекта, чтобы добиться требуемого поведения объекта в процессе использования.

Палитра свойств – это окно в виде набора свойств, которые можно определить для объекта конфигурации. Состав свойств зависит от типа редактируемого объекта.

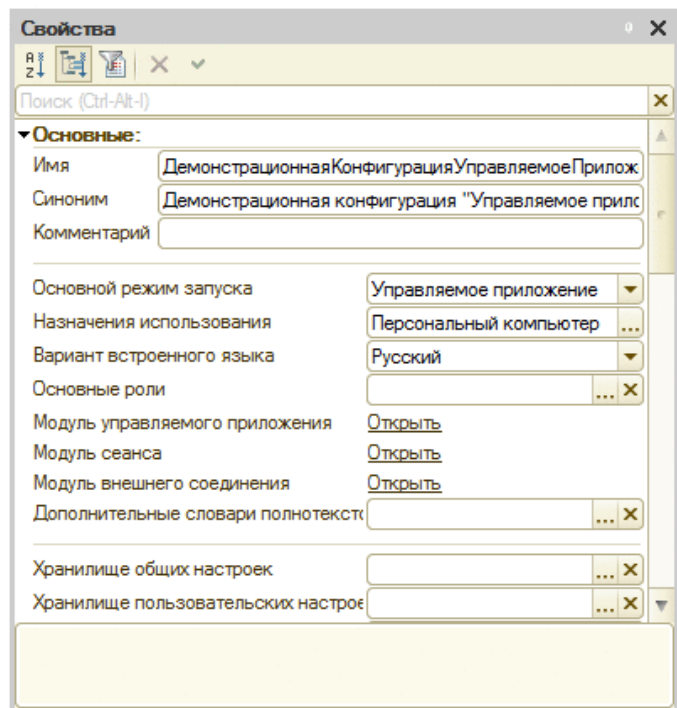


Рис. 8. Палитра свойств

В этом разделе описываются основные приемы работы с палитрой свойств.

Для вызова палитры свойств нужно указать объект конфигурации и выбрать пункт [Правка – Свойства](#).

В палитре свойств свойства объекта группируются по категориям. Число категорий и состав свойств, размещенных в каждой категории, зависит от вида рассматриваемого объекта. Например, для реквизита документа палитра свойств содержит только категории свойств [Основные](#), [Характеристики](#) и [Тип данных](#). Состав свойств категории также зависит от вида объекта. Даже в случае выбора одинаковых видов объектов состав свойств определяется индивидуальными настройками конкретных объектов. Например, для иерархического справочника в состав свойств табличного поля категории

Использование дополнительно включаются свойства [Дерево](#), [Иерархический просмотр](#) и другие.

Состав свойств также зависит от выбранных значений других свойств. Например, в свойстве [Тип](#) при выборе примитивного типа [Число](#) добавляются свойства, характеризующие выбранный тип: [Длина](#), [Точность](#) и [Неотрицательное](#).

При необходимости быстро найти какое-либо свойство в палитре свойств можно воспользоваться полем поиска. Доступ в это поле выполняется с помощью сочетания клавиш [Ctrl+Alt+I](#). Поиск выполняется по представлению свойств в палитре свойств (см. [рис. 9](#)) и начинается после ввода двух непробельных символов. Для того чтобы отключить наложенный фильтр, необходимо нажать кнопку очистки или клавишу [ESC](#), находясь в поле поиска.

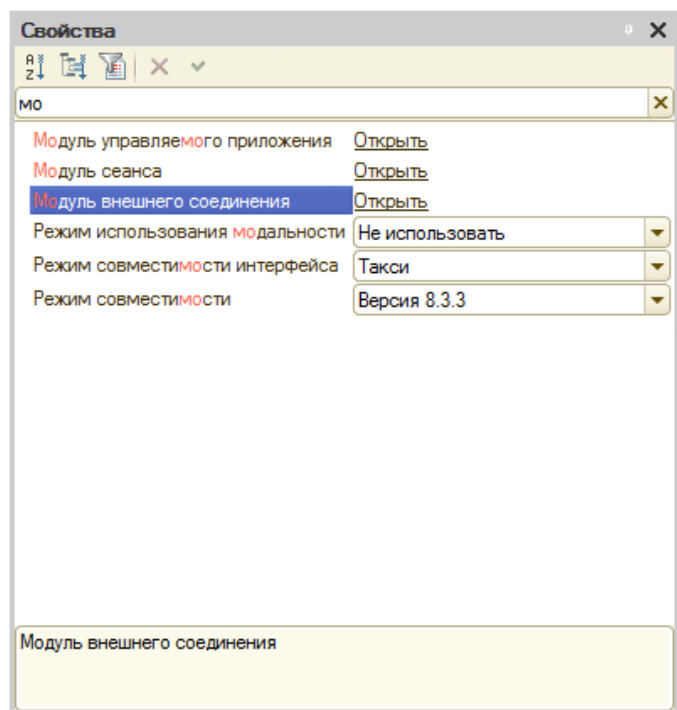


Рис. 9. Поиск в палитре свойств

Панель инструментов палитры свойств состоит из пяти кнопок (см. [рис. 10](#)). С помощью первых трех кнопок производится управление показом свойств.



Рис. 10. Кнопки окна свойств

Далее описываются кнопки панели:

- [Сортировка по алфавиту](#). Производит вывод свойств объекта, отсортированных по алфавиту (кнопка нажата). Сами категории не показываются. Если включено отображение имен свойств, то сортировка выполняется по имени, в противном случае – по представлению свойства. Если у свойства не задано имя, то сортировка всегда выполняется по представлению свойства. Отображение имен свойств управляется пунктом [Отображать имена свойств](#) контекстного меню окна свойств.
- [Сортировка по категориям](#). Производит вывод свойств объекта, отсортированных по категориям (кнопка нажата).
- [Показывать только важные](#). При нажатой кнопке выводятся только важные свойства объекта; при отжатой – все

- **Отменить редактирование.** Отмена изменений в текстовом поле свойства.
- **Сохранить.** Записать изменения в текстовое поле свойства.

Категории свойств можно располагать в виде закладок или списком. При расположении свойств списком сами свойства могут быть представлены в виде списка по категориям или по алфавиту (при этом наименования категорий не показываются).

Для выбора способа показа категорий (должен быть включен показ категорий – кнопка **Сортировка по категориям** нажата) в любом свободном месте окна палитры свойств нужно открыть контекстное меню и выбрать необходимый способ показа (**Закладками** или **Списком**). Если выбран способ показа **Закладками**, то кнопки (**Сортировка по алфавиту** и **Сортировка по категориям**) становятся недоступными.

При выборе способа показа **Закладками** для перехода к свойствам другой категории необходимо щелкнуть мышью соответствующую закладку.

Для возврата к режиму показа категорий списком в окне свойств контекстного меню нужно выбрать пункт **Списком**.

Если выбран способ показа **Списком**, то свойства объекта располагаются по категориям или по алфавиту. Для расположения по алфавиту следует нажать кнопку **Сортировка по алфавиту** панели инструментов палитры свойств. Все свойства показываются подряд в алфавитном порядке.

Для расположения по категориям нужно нажать кнопку **Сортировка по категориям** панели инструментов. Все свойства группируются по категориям в виде списка (одна под другой). Наименование категории показывается **полужирным** шрифтом. Слева от наименования расположена кнопка управления показом свойств, входящих в категорию. Кнопка позволяет скрыть или показать набор свойств.

Двойной щелчок мыши по наименованию категории свойств приводит к сворачиванию всех остальных категорий и показу свойств данной категории.

При нажатой кнопке **Показывать только важные** производится показ только важных (основных) свойств объекта (независимо от режима показа списком или закладками, по категориям или по алфавиту). Для просмотра всех свойств нужно снова нажать кнопку **Показывать только важные**.

В зависимости от вида объекта его свойства бывают доступными или недоступными для изменения. Например, в текстовом документе свойства объекта предназначены только для показа числа строк и символов, а также статуса документа, но они недоступны для изменения.

Каждое свойство в палитре свойств имеет наименование, а также развернутое пояснение. Режим получения развернутого описания устанавливается с помощью команды контекстного меню **Пояснение**, вызываемого в любом свободном месте палитры свойств (вне полей ввода). Если этот режим установлен, то при выборе свойства в нижней части палитры свойств выводится развернутое пояснение. Помимо пояснительного описания может выводиться имя свойства для доступа к значению из программного модуля.

Для свойств, которые могут быть доступны с помощью средств встроенного языка, можно настроить их показ в виде наименования или в виде имен свойств. Например, свойство с наименованием **Тип значения** имеет имя **ТипЗначения**. Режим показа можно изменить с помощью команды **Отображать имена свойств** контекстного меню. Имена свойств показываются в соответствии с выбранным вариантом встроенного языка.

ВНИМАНИЕ! Имена свойств объектов метаданных и объекта метаданных **Конфигурация** не отображаются.

Способ ввода значений в палитре свойств зависит от типа редактируемого свойства.

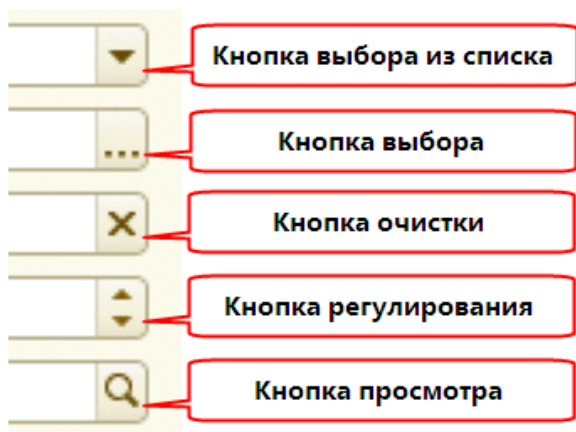


Рис. 11. Возможные действия

Для текстовых реквизитов – это обычный ввод текста (можно использовать буфер обмена), для свойств типа «флажок» – щелчок мышью. Значения некоторых свойств выбираются из списков. Поля таких свойств имеют **кнопку списка выбора**

(см. [рис. 11](#)). Если поле свойства имеет **кнопку выбора** (см. [рис. 11](#)), то при ее нажатии открывается окно, в котором производится выбор значения свойства (или его просмотр в случае, когда редактирование объекта невозможно, например, для не захваченных в хранилище конфигурации объектов или для объектов, находящихся на поддержке без возможности редактирования). К таким свойствам относится, например, выбор файла картинки, определение цвета и другие.

По **кнопке просмотра** (см. [рис. 11](#)) для текстовых данных вызывается окно редактирования строки на разных языках, для событий – процедура модуля формы, обрабатывающая данное событие, для свойств из категории **Представление** – существующая форма объекта данных. По **кнопке очистки** (см. [рис. 11](#)) производится сброс значения указанного свойства. **Кнопка регулирования** (см. [рис. 11](#)) позволяет увеличивать или уменьшать числовое значение на 1 в сторону больших или меньших значений.

Кнопки в полях могут комбинироваться.

При начале редактирования любого текстового поля палитры свойств становятся доступными кнопки панели инструментов палитры свойств **Отменить редактирование** (см. [рис. 10](#)) и **Сохранить** (см. [рис. 10](#)). Нажатие кнопки отмены отменяет произведенные изменения. Нажатие кнопки сохранения сохраняет изменение.

В палитре свойств могут располагаться ссылки, с помощью которых открываются связанные с выбранным объектом различные формы. Например, вызов справочной информации (описание) объекта конфигурации, различных форм, процедур модуля формы. Такие ссылки изображаются подчеркнутым текстом. Нажатие ссылки открывает окно, связанное с этой ссылкой.

Если просматриваются свойства объекта, редактирование которого запрещено (например, объект не захвачен в хранилище (см. [здесь](#))), то допускается открытие диалога редактирования типа в режиме «только просмотр».

2.11. Окно «Дополнительно»

Для удобства просмотра и установки основных интерфейсных свойств объекта конфигурации можно использовать окно **Дополнительно**. Для его открытия нужно выбрать пункт **Правка – Дополнительно**.

Для просмотра свойств достаточно в окне **Конфигурация** выбрать требуемый объект. Его свойства будут сразу показаны в окне **Дополнительно**.

Информация в окне распределена по закладкам.

Для объектов метаданных допустимо включение объекта в те или иные подсистемы независимо от взаимной подчиненности подсистем. Объект может быть отнесен одновременно и к «родительской», и к подчиненной ей подсистемам. Это выполняется на закладке **Подсистемы**.

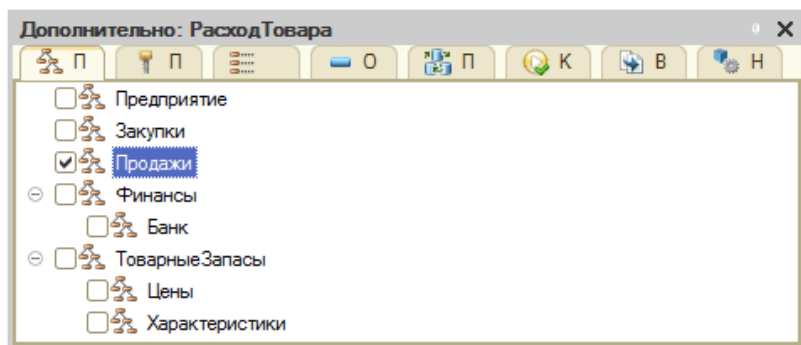


Рис. 12. Закладка «Подсистемы»

На закладке **Права** представлен список ролей и права каждой роли по данному объекту.

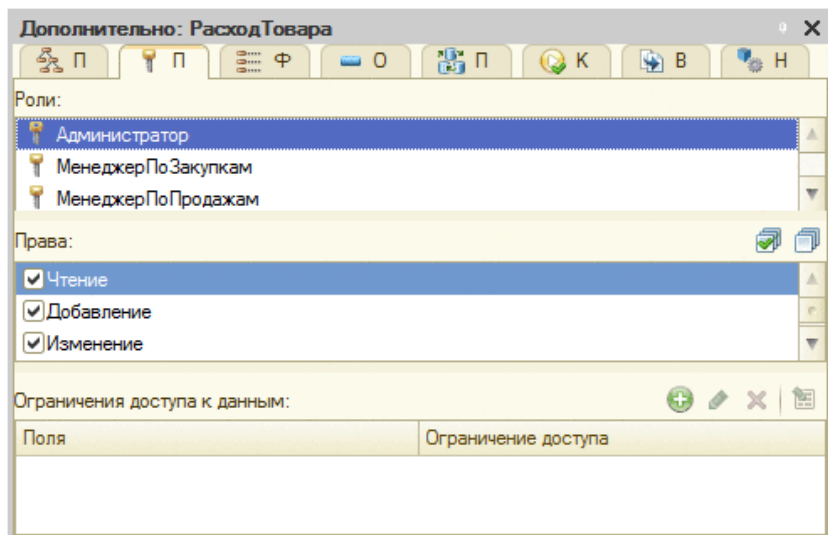


Рис. 13. Закладка «Права»

В табличном поле [Ограничение доступа к данным](#) редактируются ограничения доступа к данным на уровне отдельных полей и записей (подробнее см. [здесь](#)).

На закладке [Функциональные опции](#) представлен список функциональных опций, существующих в системе. Можно отметить те функциональные опции, к которым привязан данный объект метаданных.

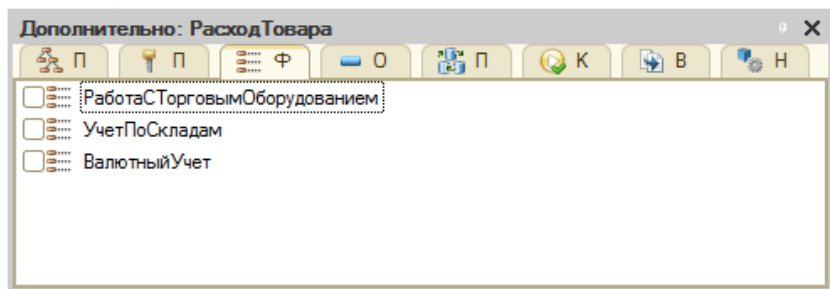


Рис. 14. Закладка «Функциональные опции»

Закладка [Общие реквизиты](#) позволяет указывать, в состав каких общих реквизитов входит текущий объект. Правила редактирования в этом окне одинаковы с правилами редактирования в окне [Состав](#) свойства общего реквизита.

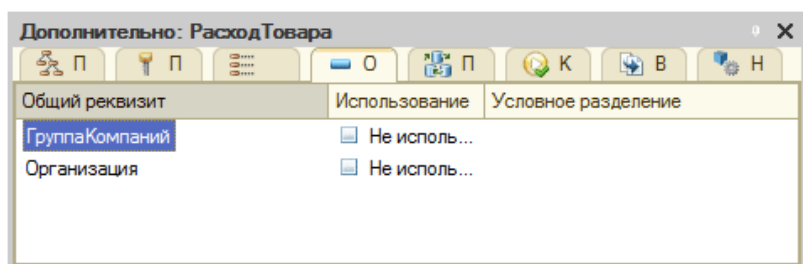


Рис. 15. Закладка «Общие реквизиты»

На закладке [Планы обмена](#) представлен список планов обмена. В списке отметками указаны те планы обмена, в которых производится учет изменений по данному объекту.

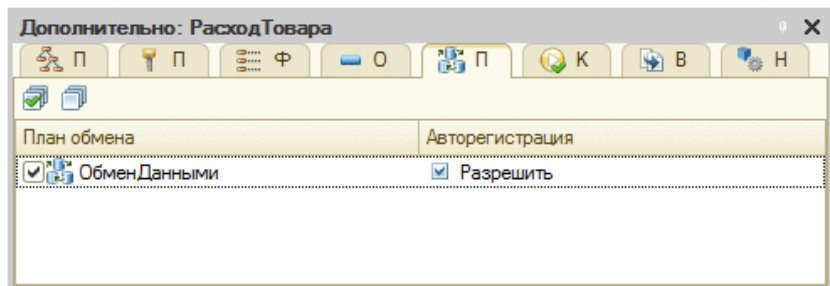


Рис. 16. Закладка «Планы обмена»

На закладке [Командный интерфейс](#) можно редактировать видимость стандартных и пользовательских команд выбранного объекта метаданных в разрезе различных подсистем. Команды, входящие в начальную страницу, собраны в узел [Рабочий стол](#).

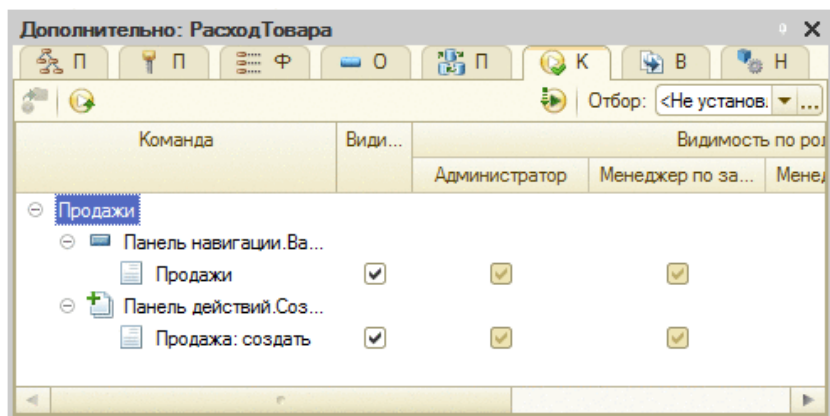


Рис. 17. Закладка «Командный интерфейс»

На закладке [Ввод на основании](#) представлен список объектов, на основании которых может вводиться данный объект, и

список объектов, являющихся основанием планов обмена.

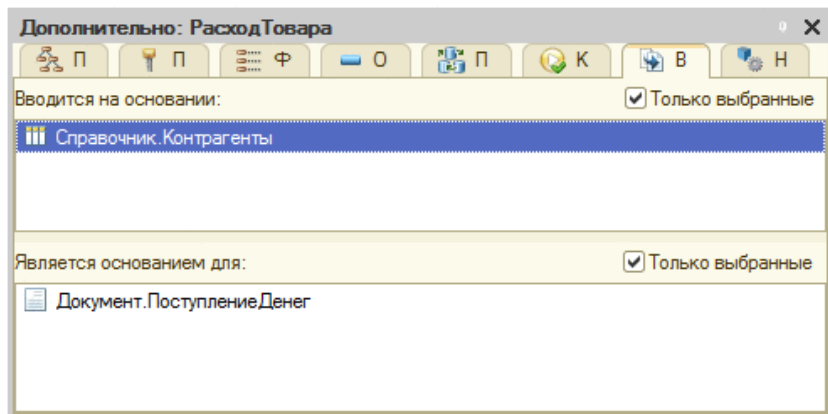


Рис. 18. Закладка «Ввод на основании»

С помощью флажка **Только выбранные** можно отображать только выбранные объекты или все объекты, которые могут вводиться на основании, где выбранные объекты отмечены флажками.

На закладке **Настройка поставки** можно выбрать правило поставки для выбранного объекта конфигурации, аналогично тому, как это делается в специальном диалоге настройки поставки (см. [здесь](#)).

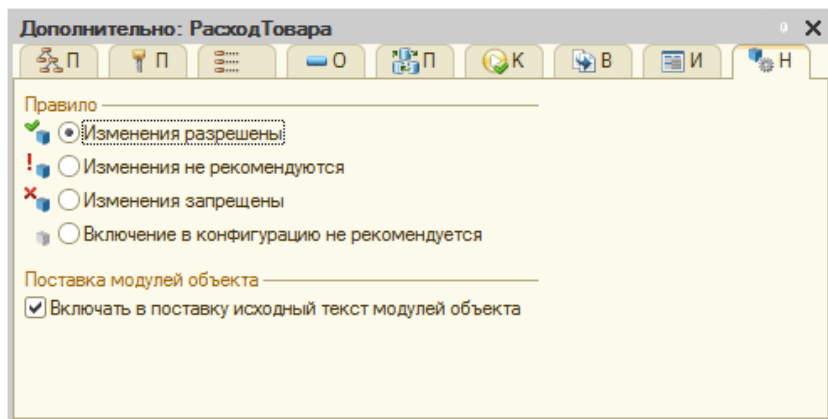


Рис. 19. Закладка «Настройка поставки»

2.12. Окно редактирования объекта

Для основных объектов конфигурации (справочники, документы, журналы документов и др.) редактирование свойств объектов, управление составом подчиненных объектов, настройку взаимодействия объектов удобно производить с помощью окна редактирования объекта.

В большинстве случаев окно редактирования объекта вызывается при выборе пункта **Действия – Изменить** окна **Конфигурация**.

Редактируемые свойства располагаются на нескольких закладках. Каждая закладка содержит набор реквизитов для настройки свойств объекта определенного вида. Так, на закладке **Основные** вводятся свойства **Имя**, **Синоним** и **Комментарий**; на закладке **Подсистемы** указывается, в каких подсистемах используется данный объект. Переход по закладкам производится с помощью кнопок **Далее >** и **< Назад**. Кроме того, нужная закладка может быть выбрана указателем мыши. В форме есть кнопка **Действия**, при нажатии которой на экран выводится контекстное меню объекта. С помощью команд этого меню можно открыть нужную форму, модуль объекта (если есть), вызвать нужный конструктор и другие действия.

Состав закладок и набор управляющих элементов на однотипных закладках может меняться в зависимости от типа объекта конфигурации.

Например, для объекта **Справочник** окно редактирования выглядит следующим образом:

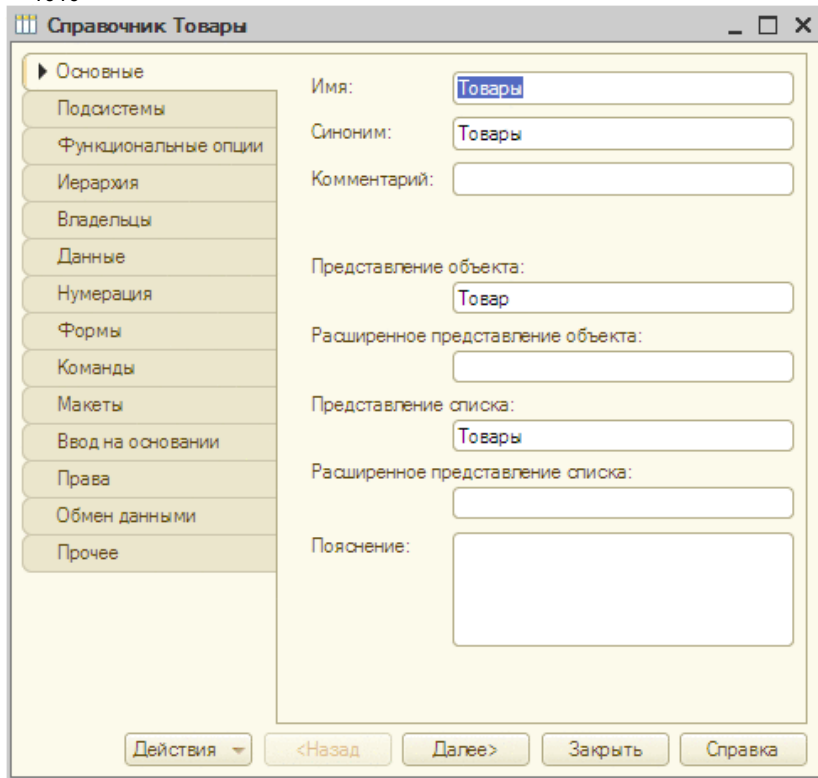


Рис. 20. Закладка «Основные»

На закладке **Основные** указываются свойства **Имя**, **Синоним** и **Комментарий**. Кроме того, на данной закладке задаются свойства, участвующие в формировании представления объекта в командном интерфейсе (подробнее см. [здесь](#)).

Если выполняется изменение имени объекта (редактируется свойство **Имя**), то система выполняет поиск возможных вхождений старого имени по конфигурации. Поиск выполняется в строковых константах, запросах (в том числе используемых в системе компоновки данных и динамических списках) и текстах ограничений доступа к данным.

В процессе поиска обнаруживаются случаи фактического использования старого наименования – случаи, когда факт использования не требует уточнения, например в программном коде присутствует конструкция **Справочники.Номенклатура** (при переименовании этого справочника), а также случаи возможного использования старого наименования – случаи, когда однозначно определить факт использования затруднительно. Если в процессе поиска были обнаружены только случаи фактического использования, то все замены выполняются автоматически (включая собственно переименование объекта). Если в процессе поиска обнаруживаются случаи возможного использования, то дальнейшие действия настраиваются с помощью диалога настройки параметров конфигулятора (подробнее см. [здесь](#)).

Работа со списком использований в коде объекта метаданных выглядит следующим образом.

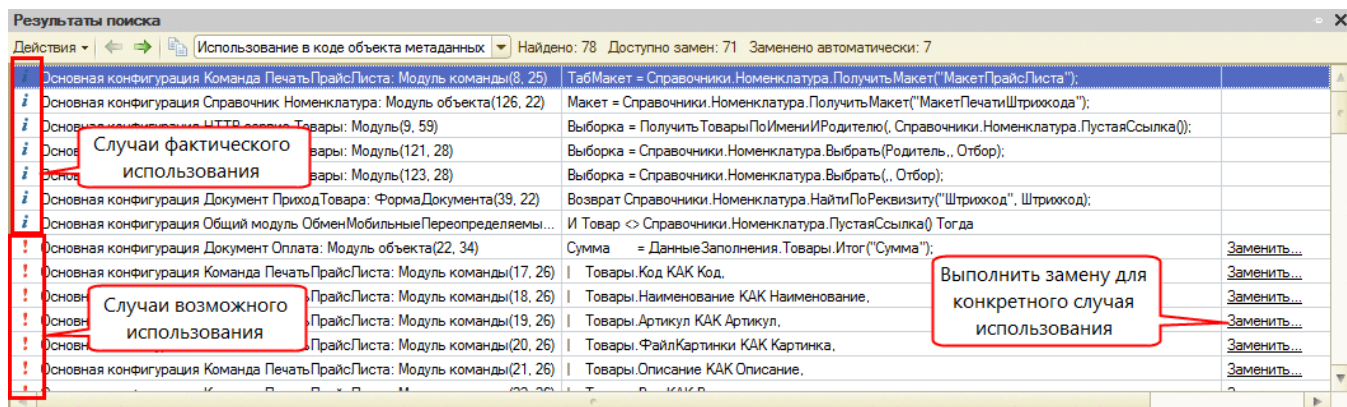


Рис. 21. Результат поиска

В левой колонке отмечается случай использования объекта конфигурации, представленный в этой строке: «i» описывает случай фактического использования, а «!» описывает случай возможного использования.

В крайней правой колонке отображается гиперссылка **Заменить...**, при нажатии на которую будет выполнена замена старого имени объекта конфигурации на новое имя в одном, конкретном месте. После замены гиперссылка пропадет. Если в этой колонке сразу отсутствует гиперссылка, то замена уже выполнена системой.

Для ускорения процесса замены можно выделить несколько строк в результатах поиска и выбрать в контекстном меню команду **Заменить...**

На закладке **Подсистемы** указывается, в каких подсистемах используется данный объект. Допустимо включение объекта в те или иные подсистемы независимо от взаимной подчиненности подсистем. Объект может быть отнесен одновременно и к «родительской», и к подчиненной подсистемам. Отнесение объекта к подсистемам определяет, в каких фрагментах командного интерфейса будут отображаться команды редактируемого объекта. Подробнее о командном интерфейсе см. [здесь](#).

На закладке **Данные** создаются реквизиты, ресурсы, измерения, табличные части и реквизиты табличных частей, а также другие подчиненные объекты (в зависимости от типа объекта). Кроме того, на закладке могут содержаться другие элементы управления для настройки свойств, характерных для конкретных типов объектов.

Так, для справочника определяется длина наименования и кода, указывается тип кода и основное представление элемента справочника.

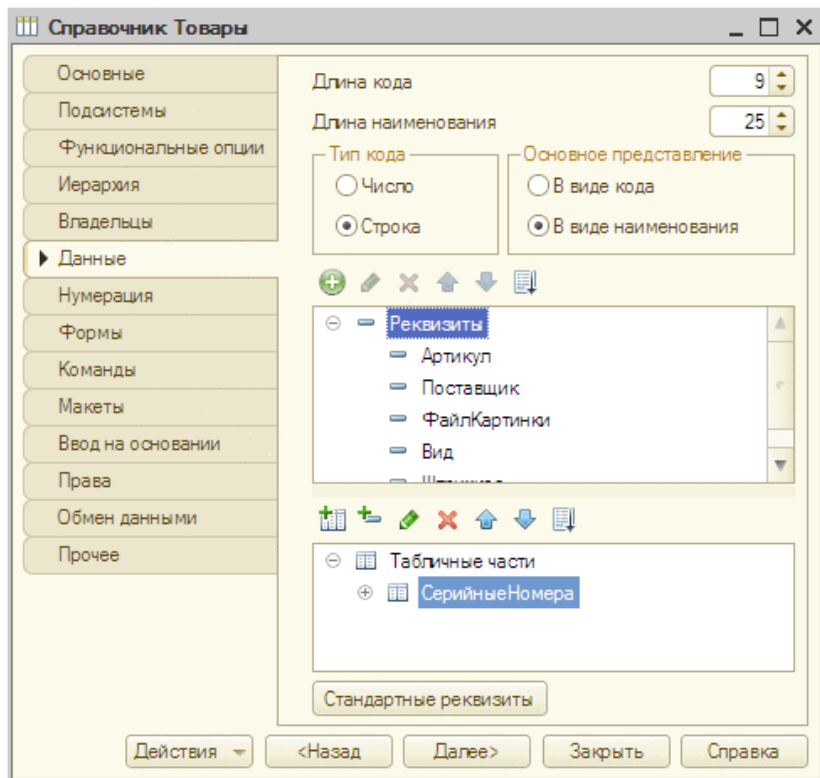


Рис. 22. Закладка «Данные»

С помощью кнопок панели инструментов, расположенной над списками подчиненных объектов, производится добавление, удаление и упорядочивание этих объектов. Свойства подчиненных объектов задаются в палитре свойств.

Если окно редактирования открыто для объекта, редактирование которого запрещено (например, объект не захвачен в хранилище), то допускается открытие диалога редактирования типа в режиме «только просмотр».

На закладке **Формы** ведется управление формами объекта и выбираются основные формы.

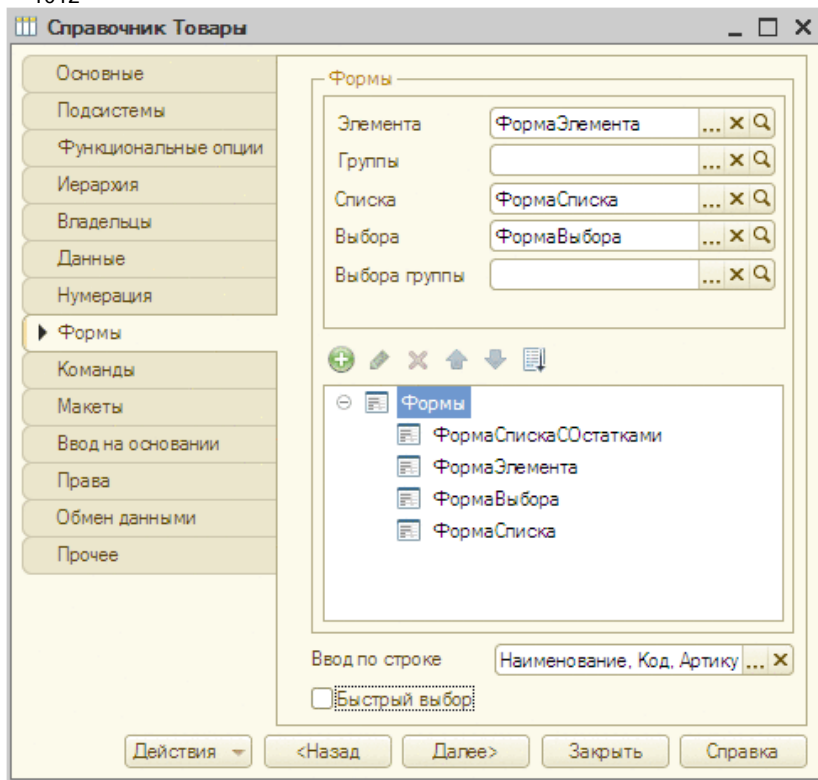


Рис. 23. Закладка «Формы»

Подробнее про основные и дополнительные формы см. [здесь](#).

В свойстве **Ввод по строке** указываются те реквизиты объекта, по которым система будет выполнять поиск информации. Подробнее про ввод по строке см. [здесь](#).

Свойство **Быстрый выбор** отвечает за режим выбора по умолчанию. Подробнее о работе данного свойства см. [здесь](#).

На закладке **Команды** имеется возможность задать пользовательские команды, связанные с данным объектом. Описание видов команд см. [здесь](#).

При добавлении новой формы запускается конструктор форм, с помощью которого производится выбор вида формы, подбор состава размещаемых в форме реквизитов и построение собственно формы. Подробнее о работе с конструктором форм см. [здесь](#). Основные приемы редактирования формы см. [здесь](#).

На закладке **Макеты** ведется управление макетами объекта.

При добавлении нового макета запускается конструктор макетов, с помощью которого создается макет. Подробнее о работе с конструктором макетов см. [здесь](#).

Ниже списка макетов располагается кнопка **Конструкторы**, при нажатии которой открывается подменю для выбора вида конструктора (состав конструкторов зависит от типа объекта):

- При выборе пункта **Конструктор печати** запускается конструктор печати, с помощью которого создается макет и процедура для печати. Подробнее о работе с конструктором печати см. [здесь](#).
- При выборе пункта **Конструктор выходной формы** запускается конструктор выходных (отчетных) форм (см. [здесь](#)).

На закладке **Права** определяются права по объектам данного типа для каждой созданной роли.

Для прикладных объектов (справочники, документы, планы видов характеристик, планы счетов, планы видов расчета, регистры, бизнес-процессы и задачи) на закладке **Планы обмена** указывается список объектов типа **ПланОбмена**. Следует установить пометку для тех планов обмена, в которых учитываются изменения редактируемого объекта.

На закладке **Прочее** расположены кнопки открытия модуля объекта, модуля менеджера, справочной информации, а также может располагаться кнопка **Предопределенные** для открытия списка предопределенных элементов объекта (для справочников, планов видов характеристик, планов счетов, планов видов расчетов). Также на закладке могут располагаться реквизиты управления блокировкой (см. [здесь](#)) и настройки использования полнотекстового поиска (см. [здесь](#)). Эти реквизиты присутствуют только для следующих прикладных объектов:

- справочники,
- документы,
- планы видов характеристик,

- планы счетов,
- планы видов расчета,
- регистры,
- бизнес-процессы,
- задачи.

Некоторые объекты могут содержать специальные закладки, относящиеся только к данному типу объектов:

- для объекта типа [Справочник](#) – это закладки [Иерархия](#), [Владельцы](#), [Нумерация](#);
- для объекта [Документ](#) – это [Нумерация](#), [Движения](#), [Журналы](#) и [Последовательности](#);
- для объекта [ПланыВидовХарактеристик](#) – [Иерархия](#);
- для объекта [ПланыВидовРасчетов](#) – [Расчет](#) и [Ввод на основании](#);
- для объекта [ПланыСчетов](#) – [Субконто](#);
- для объекта [РегистрыРасчета](#) – [Перерасчеты](#);
- для объекта [Задача](#) – [Адресация](#);
- для всех регистров – [Регистраторы](#);
- для объектов, изменения которых могут учитываться планами обмена, – [Обмен данными](#).

2.13. Создание раздела справочной информации

К некоторым объектам конфигурации можно «прикрепить» текст, объясняющий назначение и порядок использования объекта. Такой текст называется пользовательским описанием. При работе с системой «1С:Предприятие» пользователь может вывести описание на экран для просмотра.

Создание и корректировка пользовательского описания выполняются при помощи встроенного HTML-редактора, который можно вызвать из палитры свойств объекта по ссылке [Открыть свойства Справочная информация](#). Создание и редактирование текста описания производятся средствами HTML-редактора. Чтобы установить название, в тексте главы следует создать заголовок первого уровня, используя тег `<H1>` языка разметки. Следует иметь в виду, что тег `<STYLE>` (и его содержимое), который может быть указан при ручном редактировании текста справочной информации вне тега `<BODY>`, будет проигнорирован при отображении главы справки в режиме «1С:Предприятие». Если требуется применить какое-либо специальное оформление для страницы справочной информации, его определение следует выполнять внутри тега `<BODY>`.

Если установлено свойство [Включать в содержание справки](#), то для данного объекта в содержание справки может включаться элемент по значению синонима объекта. Независимо от установки данного свойства в список индексов (для поиска в справке) включается строка по значению синонима объекта. Следует помнить, что появление объекта в дереве содержания зависит от значения одноименного свойства у подсистем, в состав которых входит объект (см. [здесь](#)).

2.14. Работа с конфигурацией базы данных

2.14.1. Дерево объектов конфигурации базы данных

Для просмотра структуры конфигурации базы данных, свойств, форм, макетов и другой информации об объектах можно открыть окно конфигурации базы данных. Для этого нужно выбрать пункт [Конфигурация – Конфигурация базы данных – Открыть окно конфигурации БД](#). По виду оно не отличается от окна [Конфигурация](#).

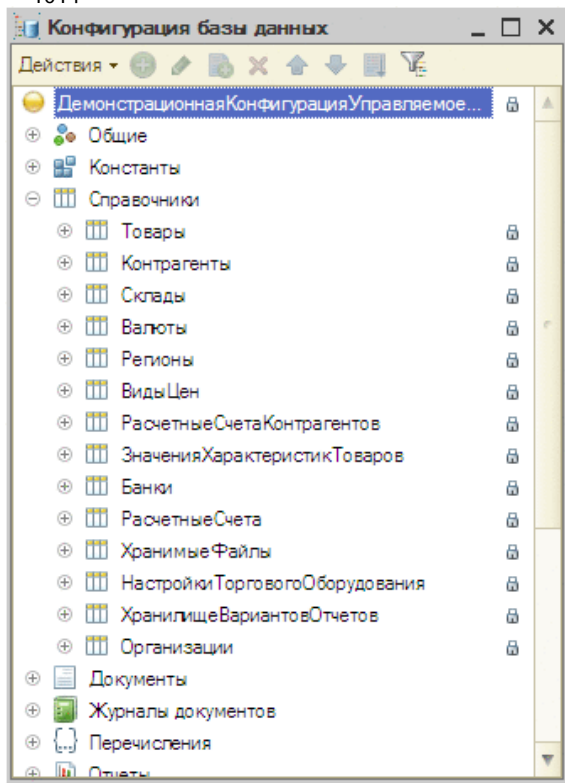


Рис. 24. Конфигурация базы данных

Приемы работы с объектами конфигурации базы данных совпадают с приемами работы в окне [Конфигурация](#) с тем лишь отличием, что все объекты доступны только для чтения (просмотра).

2.14.2. Обновление конфигурации базы данных

В процессе редактирования конфигурации могут быть созданы новые, изменены существующие или удалены имеющиеся объекты. Текущая структура базы данных может быть отлична от структуры конфигурации. Отличие конфигураций показывается в заголовке окна [Конфигурация](#) символами **<!>**.

ПРИМЕЧАНИЕ. Знак отличия **<!>** конфигураций появляется только после сохранения изменений в основной конфигурации. Однако после сохранения основной конфигурации можно продолжить внесение изменений, и в этом случае в заголовке окна [Конфигурация](#) будут присутствовать признаки изменения для обеих конфигураций.

Чтобы выполнить приведение в соответствие конфигурации и конфигурации базы данных, необходимо произвести обновление конфигурации базы данных. Для этого нужно выбрать пункт [Конфигурация – Обновить конфигурацию базы данных](#). Если основная конфигурация еще не была сохранена, то сначала конфигуратор выполнит ее сохранение, а потом произведет обновление конфигурации базы данных.

Если при обновлении конфигурации базы данных было открыто окно сообщений, то оно очищается.

ВНИМАНИЕ! Обновление конфигурации базы данных может потребовать прекращения работы всех пользователей.

Перед обновлением можно сравнить конфигурации, а также провести их объединение.

Если на момент выполнения обновления конфигурации базы данных выполнялась отладка, то после сохранения текущей конфигурации на экран выводится вопрос: [Для обновления конфигурации базы данных необходимо прекратить отладку. Продолжить?](#) При ответе **Да** отладка прекращается, и конфигурация базы данных обновляется. При ответе **Нет** не производится обновление, и отладка не прекращается.

Обновление конфигурации базы данных требует монопольного доступа конфигуратора к информационной базе. В зависимости от наличия пользователей, работающих с базой данных, и их режимов работы, возможно несколько вариантов поведения системы:

- конфигуратор выдает сообщение об ошибке исключительной блокировки в том случае, если:
 - используется файловый вариант базы данных;
 - есть сеансы, подключенные к информационной базе без использования веб-сервера;
 - нет сеансов, работающих через веб-сервер;
 - обновление конфигурации требует реструктуризации базы данных.

- конфигуратор предлагает завершить все сеансы и повторить обновление, если:
 - обновление конфигурации требует реструктуризации базы данных;
 - с файловым вариантом информационной базы работают веб-клиенты или тонкие клиенты, подключенные через веб-сервер;
- в остальных случаях конфигуратор предлагает выполнить динамическое обновление.

ПРИМЕЧАНИЕ 1. При выдаче диагностических сообщений указываются характеристики сеансов, которые мешают выполнению действия. Если количество сеансов меньше или равно 5, то выводится подробный список сеансов (с указанием имени компьютера, типа приложения и т. д.), в противном случае выводится общее число сеансов.

ПРИМЕЧАНИЕ 2. Работа информационной базы в монопольном режиме не переводит базу данных Microsoft SQL Server в однопользовательский (single user) режим.

ПРИМЕЧАНИЕ 3. Для ускорения процесса реструктуризации информационной базы, при использовании СУБД Microsoft SQL Server, рекомендуется режим восстановления для базы данных устанавливать в значение **Простой** или **С неполным протоколированием**. Смену режима можно выполнять или перед выполнением реструктуризации или на постоянной основе, если не требуется выполнять восстановление базы данных на произвольный момент времени. **Перед сменой режима восстановления базы данных необходимо выполнить резервное копирование базы данных!**

Ошибка монопольного доступа

Если система не может получить монопольный доступ, то имеется возможность только дождаться, пока пользователи будут отключены от информационной базы, и повторить операцию обновления.

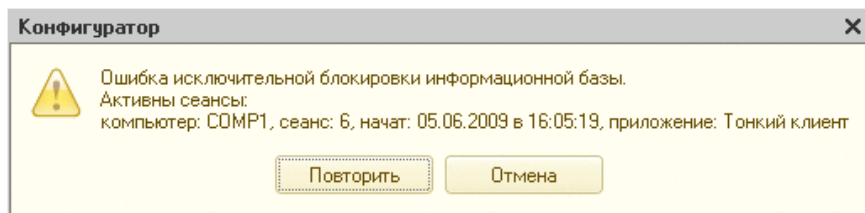


Рис. 25. Ошибка исключительной блокировки

Завершение сеансов и попытка обновления

Если для обновления конфигурации базы данных необходимо завершить все сеансы, то пользователю выдается сообщение (см. [рис. 26](#)).

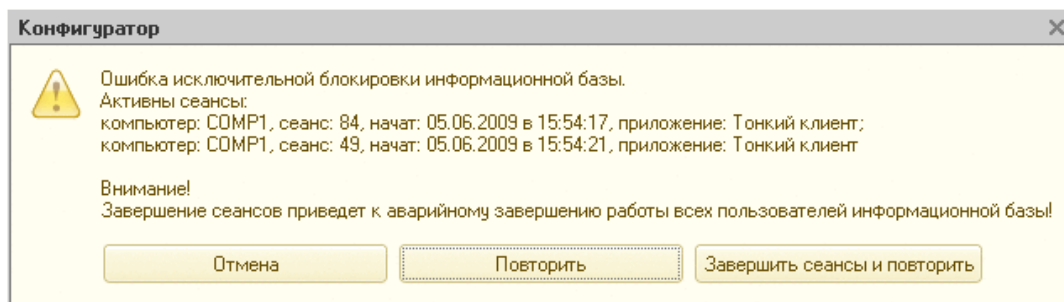


Рис. 26. Отключение сеансов для обновления

Если выбрана команда **Завершить сеансы и повторить**, то у пользователя запрашивается подтверждение выбранного действия (**Завершение сеансов приведет к аварийному завершению работы пользователей! Выполнить завершение сеансов?**) и в случае утвердительного ответа происходит попытка завершения работы всех сеансов информационной базы. Затем выполняется попытка повторного сохранения конфигурации базы данных.

Завершение всех сеансов приведет к аварийному завершению работы всех клиентских приложений.

Возможны ситуации, когда завершение работы сеанса невозможно. Например, файловый вариант информационной базы опубликован на веб-сервере, доступ к которому требует клиентского сертификата или на веб-сервере настроена аутентификация и для доступа требуется ввод имени пользователя и пароля. При этом к информационной базе подключения выполняются только с использованием веб-сервера. Конфигуратор не поддерживает аутентифицированный доступ к веб-серверу и поэтому не может выполнить попытку завершения сеансов. В том случае, если Конфигуратор не смог завершить сеансы доступа к информационной базе, попытку обновления конфигурации базы данных можно либо выполнить позже, либо завершить работу сеансов другими способами.

Динамическое обновление

Если есть возможность выполнить динамическое обновление, пользователю выдается специальное сообщение (см. [рис. 27](#)).

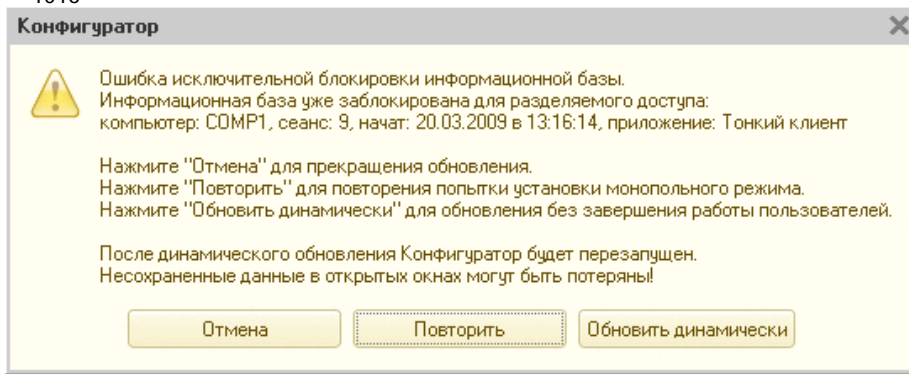


Рис. 27. Динамическое обновление

Если выбрана команда **Обновить динамически**, то выполняется обновление без завершения работы пользователей. Предполагается, что выполненные изменения будут записаны динамически в виде версии изменений конфигурации (конфигурация базы данных при этом не изменяется). Допускается выполнение повторных изменений основной конфигурации. Если при очередной попытке обновления конфигурации базы данных может быть установлен монопольный режим работы, конфигуратор осуществляет обновление конфигурации базы данных с учетом всех изменений (как текущих, так и предыдущих).

ПРИМЕЧАНИЕ. Если используется клиент-серверный вариант работы с информационной базой, то после обновления конфигуратор будет перезапущен. При этом все несохраненные изменения в текстовых, табличных и других документах будут утеряны.

Если было выполнено динамическое обновление, то работающие в этот момент пользователи продолжают работать со старой конфигурацией. Для того чтобы начать работать с обновленной конфигурацией, пользователю необходимо перезапустить систему «1С:Предприятие». Для контроля и оповещения пользователей о произведенных динамических изменениях следует использовать метод глобального контекста `КонфигурацияБазыДанныхИзмененаДинамически()`.

ПРИМЕЧАНИЕ. После выполнения обновления конфигурации базы данных все версии, созданные динамическим обновлением, будут удалены.

Если были обнаружены изменения, требующие реструктуризации базы данных, то на экран выводится диалог со списком таких изменений для подтверждения обновления.

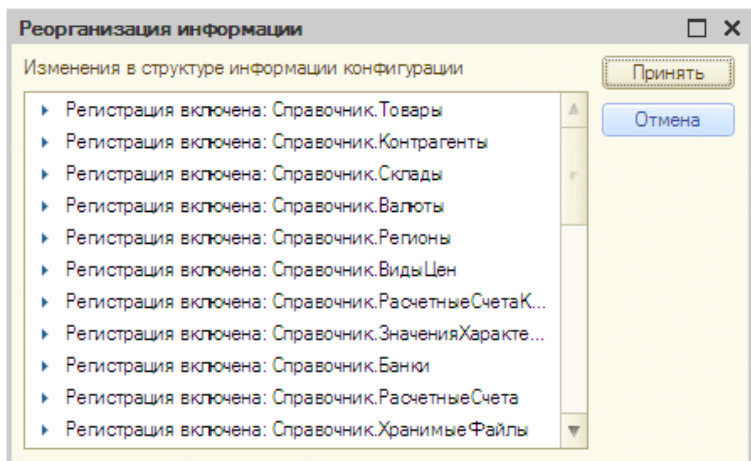


Рис. 28. Реорганизация информации

Для подтверждения сохранения нужно нажать кнопку **Принять**, для отказа – кнопку **Отмена**.

2.14.3. Фоновое обновление конфигурации базы данных

2.14.3.1. Общее описание

Обновление конфигурации базы данных, которое связано с реструктуризацией базы данных, выполняемое для информационных баз большого объема, может занимать длительное время. Во время выполнения обновления, работать с информационной базой не возможно.

Для того чтобы свести потери времени на эту операцию к минимуму, существует специальный режим, выполняющий обновление конфигурации базы данных в фоновом режиме. Фоновое обновление конфигурации базы данных характеризуется следующими особенностями:

- Доступно только в клиент-серверном варианте информационной базы;
- Может выполняться при закрытом конфигураторе.

- Большая часть фонового обновления конфигурации базы данных выполняется без монопольного доступа к базе данных (включая выполнение операции реструктуризации базы данных).
- Во время фонового обновления недоступны следующие операции:
 - Редактирование конфигурации.
 - Отладка прикладного решения.
 - Выполнение операции обновления конфигурации базы данных.
 - Не допускается использование методов `УстановитьРежимАгрегатов()`, `УстановитьИспользованиеАгрегатов()`, `ПерестроитьИспользованиеАгрегатов()`.
 - Изменение содержимого плана счетов или плана видов расчета, если связанный с ним регистр бухгалтерии или регистр расчета участвует в операции фонового обновления. Попытка изменить содержимое таких планов счетов или планов вида расчета вызывает ошибку.
- Фоновое обновление конфигурации можно поставить «на паузу» длительностью не более 48 часов. Если «пауза» продлится более 48 часов – фоновое обновление будет отменено.
- Не поддерживается работа фонового обновления для конфигураций, находящихся в режиме совместимости с версией 8.1 (см. [здесь](#)).
- Не поддерживается работа фонового обновления конфигурации базы данных при работе на СУБД IBM DB2 9.1.

Процесс фонового обновления конфигурации состоит из нескольких этапов:

- Фаза **обработки**:
 - Выполняется длительное время.
 - Запуск данной фазы может быть выполнен любым способом:
 - Интерактивно, из конфигуратора;
 - Из встроенного языка (с помощью соответствующих методов);
 - С помощью режима пакетного запуска конфигуратора.
 - Во время выполнения фазы пользователи могут работать с информационной базой.
 - Выполняется реструктуризация основного объема данных для следующих объектов конфигурации:
 - Справочники,
 - Документы,
 - Журналы документов,
 - Регистры сведений,
 - Регистры накоплений,
 - Регистры бухгалтерии,
 - Регистры расчета,
 - Последовательности,
 - Планы счетов,
 - Бизнес-процессы,
 - Задачи.
 - Во время выполнения фазы обработки, система фиксирует все измененные данные для вышеперечисленных объектов, по аналогии с механизмами обмена данными.
- Фаза **актуализации**:
 - Запускается автоматически, после окончания фазы обработки, с интервалом 1 минута.

- Во время выполнения фазы пользователи могут работать с информационной базой.

- Фаза состоит из автоматически повторяющихся итераций. Каждая итерация анализирует изменения, накопленные с момента выполнения предыдущей итерации (или завершения фазы обработки) и выполняет реструктуризацию накопленных изменений.

- Итерации завершаются в момент перехода к следующей фазе.

- Фаза **принятия изменений**:

- Требуется монопольный доступ к информационной базе.

- Во время выполнения фазы пользователи **не** могут работать с информационной базой.

- Первым шагом этой фазы выполняется актуализация данных, накопленных с момента проведения последней, перед текущей фазой, итерации фазы актуализации.

- Затем выполняется реструктуризация данных, не участвующих в фазах обработки и актуализации. В этих данных не предполагается большого объема изменений, и их реструктуризация выполняется быстро.

- Следующим шагом выполняется принятие всех изменений, выполненных в базе данных.

- После этого обновление конфигурации базы данных завершается.

Если фоновое обновление запущено так, что не требуется выполнения реструктуризации базы данных, то в этом случае все обновление выполняется в фазе принятия изменений, переход к которой возможен сразу после начала фонового обновления.

Во время фонового обновления имеется возможность остановить работу сервера или поставить процесс фонового обновления «на паузу».

После остановки сервера или после аварийного завершения рабочего процесса, обслуживающего системное фоновое задание, выполняющее обновление, создание первого сеанса будет происходить несколько большее время, нежели обычно. Это связано с восстановлением фонового обновления. Однако сам процесс фонового обновления находится в приостановленном состоянии. Для продолжения работы необходимо возобновить работу фонового обновления. Такое поведение реализовано для того, чтобы не произошло заикливание системы в том случае, если причиной аварийного завершения рабочего процесса, обслуживающего фоновое обновление, является собственно фоновое обновление.

После восстановления работоспособности сервера, фоновое обновление конфигурации базы данных продолжает работу следующим образом:

- Если работа была прервана в фазе обработки, то процесс продолжается с последнего объекта конфигурации, обработка которого не была завершена.

- Если работа была прервана в фазе актуализации – незаконченная итерация начинается заново.

СОВЕТ. Рекомендуется отменять запущенный процесс фонового обновления конфигурации базы данных в тех случаях, когда не планируется завершение фонового обновления конфигурации базы данных (например, процесс поставлен на бесконечную «паузу»). Выполнение данной рекомендации положительно скажется на характеристиках системы, т. к. не будет происходить регистрация изменений для выполнения фазы актуализации.

При выполнении операции фоновой реструктуризации следует учитывать некоторые особенности работы системы:

- Если регистр бухгалтерии или накопления добавлен в состав разделителя, то данный регистр обрабатывается во время фазы принятия изменений.

- Если изменен тип независимого разделителя (см. [здесь](#)), то все объекты, входящие в состав этого разделителя, обрабатываются во время фазы принятия изменений.

- Если изменяется тип измерения, входящего в основной отбор независимого регистра сведений, то такой регистр обрабатывается во время фазы принятия изменений.

2.14.3.2. Диалог фонового обновления конфигурации базы данных

Для вызова операции фонового обновления конфигурации базы данных нужно выбрать пункт меню [Конфигурация – Конфигурация базы данных – Фоновое обновление конфигурации базы данных](#).

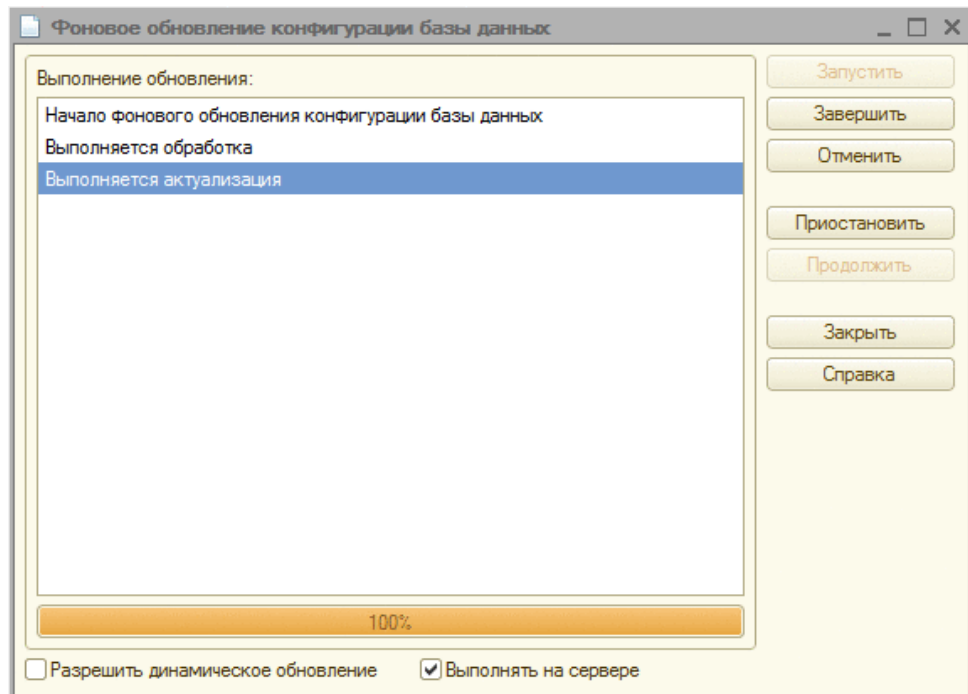


Рис. 29. Диалог фонового обновления конфигурации базы данных

Нажатие кнопки **Запустить** запускает фазу обработки. При этом в окне служебных сообщений отображается текст:

[Копировать в буфер обмена](#)

Старт фонового обновления конфигурации БД прошел успешно
Конфигурация недоступна для редактирования. Выполняется фоновое обновление конфигурации базы данных.

После запуска фонового обновления конфигурация блокируется от изменения. Во время выполнения любой фазы, работу фонового обновления можно приостановить с помощью кнопки **Приостановить**. Для того, чтобы продолжить выполнение фонового обновления служит кнопка **Продолжить**.

После окончания фазы обработки начинается фаза актуализации, во время выполнения которой можно перевести систему к выполнению фазы принятия изменений с помощью кнопки **Завершить** или отказаться от обновления с помощью кнопки **Отменить**.

СОВЕТ. Рекомендуется отменять запущенный процесс фонового обновления конфигурации базы данных в тех случаях, когда не планируется завершение фонового обновления конфигурации базы данных (например, процесс поставлен на бесконечную «паузу»). Выполнение данной рекомендации положительно скажется на характеристиках системы, т. к. не будет происходить регистрация изменений для выполнения фазы актуализации.

Флажок **Разрешить динамическое обновление** служит для того, чтобы определить, нужно ли при нажатии на кнопку **Запустить** попытаться выполнить динамическое обновление вместо запуска процесса фонового обновления конфигурации базы данных. Если этот флажок установлен, то сразу же после нажатия на кнопку **Запустить** будет производиться проверка на возможность выполнить динамическое обновление конфигурации. В случае, когда конфигурация допускает такую возможность, вместо фонового обновления конфигурации базы данных будет производиться динамическое обновление.

Флажок **Выполнять на сервере** служит для того, чтобы определить, где будет выполняться запуск, завершение и отмена. Если этот флажок сброшен, то данные действия будут выполняться на стороне клиентского приложения, а если установлен – то на стороне сервера «1С:Предприятия». Кроме того, если установлен данный флажок, то можно выполнять обновление конфигурации только при наличии права **ОбновлениеКонфигурацииБазыДанных**, без установленного права **Администрирование**.

2.14.4. Сохранение конфигурации базы данных в файл

Для сохранения конфигурации базы данных в файл на диск нужно выбрать пункт **Конфигурация – Конфигурация базы данных – Сохранить конфигурацию БД в файл**. На экран выводится стандартный диалог выбора файла. Необходимо выбрать каталог и указать имя файла, в который будет записана конфигурация базы данных.

Сохраненный файл конфигурации базы данных необходим для операции сравнения и объединения конфигураций (см. [здесь](#)).

2.14.5. Сравнение конфигурации и конфигурации базы данных

Если в процессе внесения изменений в конфигурацию требуется получить отчет об отличиях от конфигурации базы данных, то следует выбрать пункт **Конфигурация – Конфигурация базы данных – Сравнить, объединить с конфигурацией БД**.

В случае необходимости можно восстановить измененные объекты.

2.14.6. Отказ от изменений в конфигурации

Для отказа от изменений в конфигурации достаточно выбрать пункт [Конфигурация – Конфигурация базы данных – Вернуться к конфигурации БД](#).

ПРИМЕЧАНИЕ. Пункты меню [Сохранить конфигурацию БД в файл...](#) и [Вернуться к конфигурации БД](#) доступны даже в случае закрытой редактируемой конфигурации. Команда [Вернуться к конфигурации БД](#) по-прежнему недоступна, когда информационная база подключена к хранилищу конфигурации.

2.15. Запуск «1С:Предприятия»

В конфигураторе предусмотрен запуск режима 1С:Предприятие. Для этого нужно выбрать пункт [Сервис – 1С:Предприятие](#). Часто необходимо запустить «1С:Предприятие» в режиме отладки. Для этого существует команда [Отладка – Начать отладку](#) (подробнее про отладчик см. [здесь](#)).

Если конфигурация была модифицирована (были произведены изменения), то конфигуратор выводит вопрос: [Редактируемая конфигурация отличается от конфигурации базы данных. Обновить конфигурацию базы данных?](#) Для сохранения внесенных изменений следует выбрать кнопку [Да](#).

Если выбрана кнопка [Нет](#), то режим 1С:Предприятие запускается без сохранения конфигурации.

В случае отказа на экран выводится вопрос: [Конфигурация базы данных не соответствует сохраненной конфигурации. Продолжить?](#) Если выбрана кнопка [ОК](#), то запускается режим 1С:Предприятие с прежней конфигурацией базы данных. Если выбрана кнопка [Отмена](#), то запуск режима 1С:Предприятие не производится.

2.16. Выгрузка и загрузка файлов конфигурации

Механизм выгрузки и загрузки файлов конфигурации позволяет осуществлять выборочную выгрузку/загрузку некоторых свойств объектов конфигурации (модулей, макетов и справочной информации). Для выгрузки свойств нужно выбрать пункт [Конфигурация – Выгрузить файлы конфигурации](#).

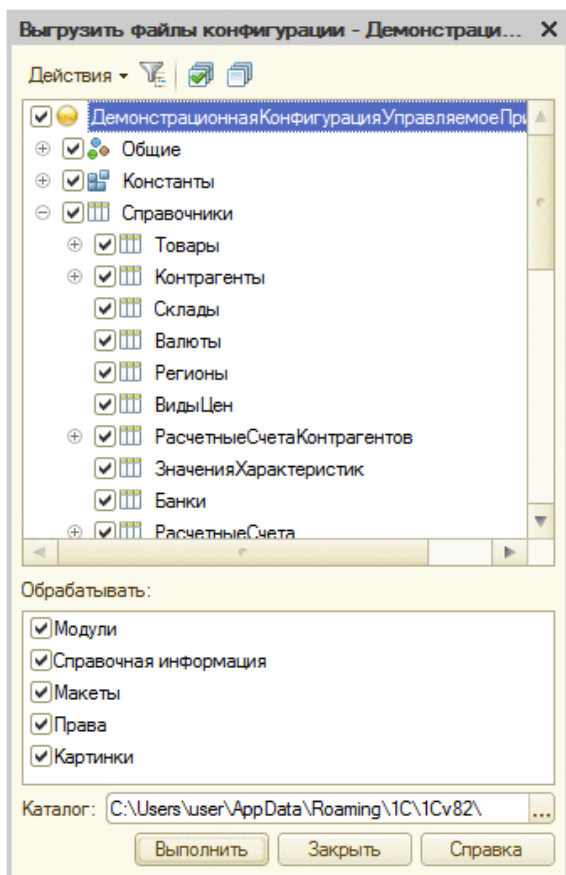


Рис. 30. Выгрузить файлы конфигурации

В открывшемся окне следует отметить те объекты конфигурации, которые нужно выгрузить, укажите вид выгружаемых данных и укажите каталог, в который будет осуществлена выгрузка. Для начала выгрузки нажмите кнопку [Выполнить](#). Выгружаемые данные записываются в файлы с именами, соответствующими названию выгружаемого свойства. Расширение файла соответствует типу данных: .

- [htm](#) – для справки и макетов HTML-документа;
- [txt](#) – для модулей и макетов текстового документа;

- **mxl** – для макетов табличного документа;
- **geo** – для макетов географической схемы;
- **grs** – для макетов графической схемы;
- **bin** – для макетов двоичных данных.

Картинки выгружаются в том формате, в котором они хранятся в конфигурации. Загружаются картинки всех поддерживаемых форматов (см. [здесь](#)).

Для выгрузки/загрузки прав доступа необходимо выбрать требуемые роли (или все) и указать пункт **Права** в списке **Обрабатывать**. Права доступа выгружаются в формате XML. Файл выгрузки включает в себя собственно права доступа, ограничения доступа к данным и шаблоны ограничений доступа.

При выгрузке файлов справки и макетов HTML-документов формируется отдельный файл для каждого языка, для которого определен макет. При этом код языка включается в имя файла перед его расширением: например, [Документ.РасходТовара.Справка.ru.htm](#) означает файл со справочной информацией на русском языке документа [РасходТовара](#).

Загрузка данных осуществляется выбором пункта **Конфигурация – Загрузить файлы конфигурации**. Действия в окне аналогичны действиям, описанным для режима выгрузки данных.

При загрузке производится попытка загрузить файлы справки и макеты HTML-документов для каждого языка, определенного в прикладном решении. Если при загрузке обнаружен только один файл (без указания в наименовании кода языка), то он загружается для текущего языка конфигурации.

2.17. Выгрузка и загрузка конфигурации

Данный механизм позволяет осуществлять выгрузку всех объектов конфигурации в xml-файлы и выполнить обратную загрузку. Все объекты конфигурации выгружаются в один или несколько файлов. Основным файлом в выгрузке является файл [Configuration.xml](#), который содержит описание свойств собственно конфигурации.

Имеется возможность выполнить выгрузку и загрузку расширения конфигурации (см. [здесь](#)). В этом случае следует использовать команды меню **Действия – Конфигурация** (из окна с конфигурацией расширения), либо соответствующими командами в диалоге **Расширения конфигурации** (см. [здесь](#)). Выгружаться будут расширенные или контролируемые свойства.

В отдельные файлы выгружаются:

- Объекты конфигурации – объекты верхнего уровня, и подчиненные объекты, отдельно захватываемые в хранилище конфигурации.

Примеры имен файлов:

- [Document.ПродажиТоваров.xml](#) – описание объект конфигурации [Документ.ПродажиТоваров](#).
- [CommonPicture.ПодсистемаЗакупки.xml](#) – описание объекта конфигурации [ОбщаяКартинка.ПодсистемаЗакупки](#) (но не сама картинка).
- Отдельные свойства – как правило, имеющие большой объем или сложную структуру.

Примеры имен файлов:

- [CommonPicture.ПодсистемаЗакупки.Picture.xml](#) – описание картинки (но не сама картинка).
- [DataProcessor.Путеводитель.Template.ГлавнаяСтраница.xml](#) – описание макета, расположенного в обработке (но не сам макет).
- Значения отдельных свойств, являющиеся файлами известных форматов (но не XML). К таким файлам можно отнести модули форм, макеты, картинки.

Примеры имен файлов:

- [CommonPicture.ПодсистемаЗакупки.Picture.Picture.png](#) – картинка, сохраненная в общие картинки под именем [ПодсистемаЗакупки](#).
- [Catalog.Товары.Form.ФормаЭлемента.Form.Module.txt](#) – модуль управляемой формы элемента справочника [Товары](#).
- [DataProcessor.ПомощникСозданияОбменаДанными.Form.Форма.Form.КартинкаВторойИнформационнойБазы.Picture.png](#) – абсолютная картинка содержащаяся в свойстве [Picture](#) элемента управляемой формы [КартинкаВторойИнформационнойБазы](#).

Объекты, не являющиеся объектами разработки сохраняются в файле родительского объекта, например, файл `DataProcessor.ПечатьЭтикетокИЦенников.Template.КомпонентаПечатиШтрихкодовLinux32.Template.bin` означает собственно внешнюю компоненту печати этикеток и ценников, расположенную в двоичном макете `КомпонентаПечатиШтрихкодовLinux32`.

Специальным образом выгружаются следующие элементы:

- Модуль объекта – выгружается в виде текстового файла (если таковой может быть получен) или в бинарном виде.

Пример имени файла:

- `DataProcessor.ЖурналРегистрации.ObjectModule.txt` – модуль объекта метаданных `Обработка.ЖурналРегистрации`.
- HTML – выгружается в виде набора файлов: корневой файл свойства, файлы HTML страниц, файлы картинок (если таковые имеются). Корневой файл содержит список HTML страниц входящих в документ. HTML файлы являются страницами входящими в документ. Каждая из них соответствует конкретному языку, определенному в конфигурации. Справка по объектам конфигурации является частным случаем выгрузки HTML.

Примеры имен файлов (для свойства `Справка`):

- `Catalog.Товары.Help.xml` – корневой файл внешнего свойства справка объекта метаданных `Справочник.Товары`.
- `Catalog.Товары.Help.ru.html`, `Catalog.Товары.Help.en.html` –страницы справки для русского и английского языков соответственно.
- `Catalog.Товары.Help.Картинка.png` – файл картинки, содержащийся в HTML документе со справкой.
- Конфигурации поставщиков – выгружается как файл информации о конфигурациях поставщиков во внутреннем формате (не XML) и файлы конфигурации (`.cf`) для каждой конфигурации поставщика.

Примеры имен файлов:

- `Configuration.ParentConfigurations` – файл информации о конфигурациях поставщиков.
- `Configuration.ParentConfigurations.УправлениеТорговлей.cf` – конфигурация поставщика `УправлениеТорговлей`.

При выгрузке все ссылки на объекты метаданных, ссылки на поля, идентификаторы типов и т. д. выгружаются в виде имен (`Catalog.Товары`). Все идентификаторы выгружаются на английском языке – упоминание справочника `Товары` будет выглядеть `Catalog.Товары`, а не `Справочник.Товары`.

Если в объектах конфигурации существуют ссылки на отсутствующие объекты конфигурации, то такие ссылки выгружаются в виде текстового представления идентификатора и при загрузке также будут восстановлены в ссылку на отсутствующий объект конфигурации (для сохранения идентичности двух конфигураций). Для реквизитов и элементов управляемой формы выгружаются внутренние идентификаторы, используемые, в частности, для сохранения и восстановления настроек.

При выгрузке ролей выгружаются значения только тех прав, которые отличаются от значений по умолчанию в момент выгрузки. При определении значений прав по умолчанию используются свойства роли `Устанавливать права для новых объектов` и `Устанавливать права для реквизитов и табличных частей по умолчанию`. Значения этих флажков являются значениями по умолчанию для прав доступа на объекты конфигурации и прав доступа на реквизиты и табличные части, соответственно.

ПРИМЕЧАНИЕ. Обычные формы и интерфейсы обычного приложения выгружаются в двоичном виде.

При загрузке, если обнаружены ошибки в файлах, то будет сформировано окно со списком ошибок. Если ошибки не являются критичными, то можно нажать кнопку `Принять`. Выбрав строку с ошибкой – можно открыть файл, содержащий ошибку.

2.18. Расширения конфигурации

Открытие списка подключенных расширений конфигурации выполняется с помощью команды главного меню `Конфигурация – Расширения конфигурации`. В открывшемся окне можно создавать новые расширения и управлять подключенными. Подробнее описание работы с расширениями с помощью этого окна см. [здесь](#).

2.19. Работа с мобильным приложением

Описание работы с мобильным приложением см. [здесь](#).

2.20. Отчет по конфигурации

В конфигураторе можно вывести в текстовом или табличном виде информацию обо всех объектах конфигурации. Для этого следует выбрать пункт `Конфигурация – Отчет по конфигурации`.

На экран выводится диалог. В нем нужно выбрать тип (текстовый или табличный) и имя файла, в который предполагается сохранить описание структуры конфигурации.

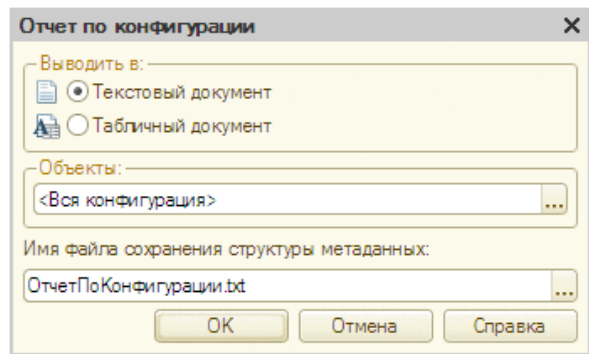


Рис. 31. Отчет по конфигурации

Создание описания объектов конфигурации для сложных конфигураций может занимать продолжительное время.

Если требуется получить отчет по отдельным объектам конфигурации, то в поле **Объекты** нужно нажать кнопку выбора и в открывшемся диалоге выбора объектов установить флажки у нужных объектов.

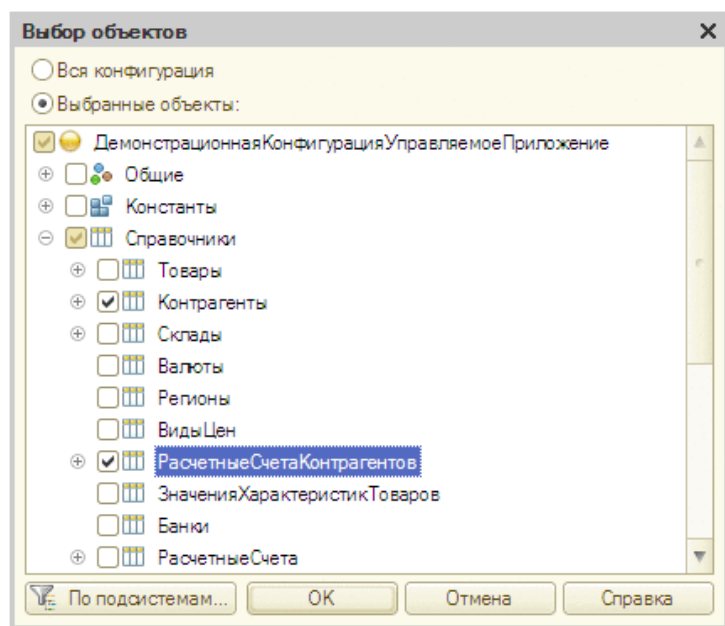


Рис. 32. Выбор объектов для отчета по конфигурации

Допускается отбор по подсистемам.

После окончания создания описания конфигурации будет открыто окно с описанием в выбранном формате (текстовом или табличном).

2.21. Глобальный поиск и замена

Режим глобального поиска и замены предназначен для поиска определенной строки во всех модулях, диалогах, табличных документах, описаниях конфигурации и внешних файлах (внешние отчеты и обработки, табличные документы). Найденный текст может быть заменен другим. Этот режим может быть использован, например, для поиска всех вызовов некоторой глобальной процедуры или обращения к какому-либо реквизиту в разных модулях.

Вызов режима поиска осуществляется выбором пункта **Правка – Глобальный поиск**, а режима замены – выбором пункта **Правка – Глобальная замена**.

В этих режимах используется один и тот же диалог. Если выбран режим поиска, то становятся недоступными реквизиты режима замены. Поэтому для краткости рассмотрим процедуру глобальной замены, а потом укажем особенности режима поиска.

На экран будет выдан диалог для задания параметров поиска.

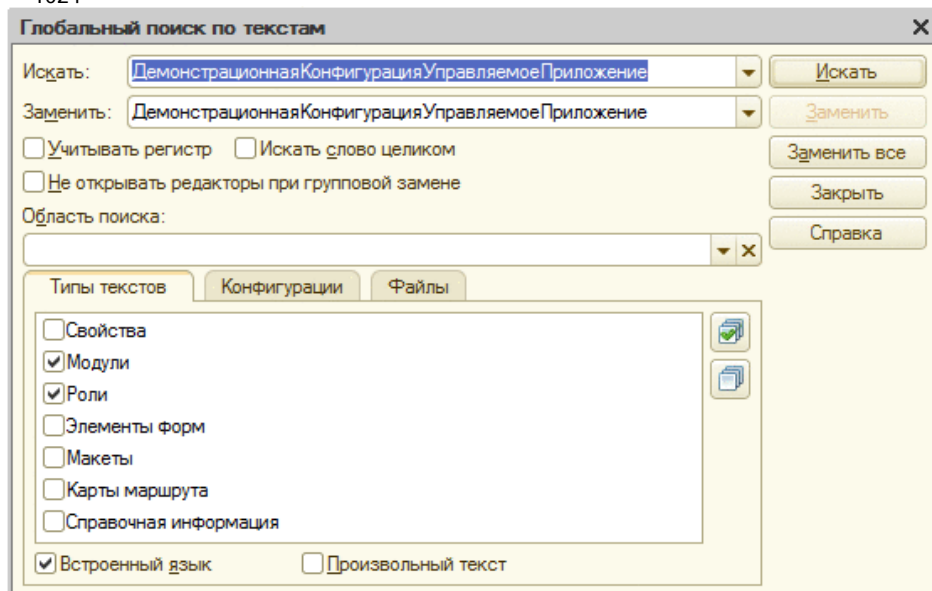


Рис. 33. Окно глобальной замены

В поле **Искать** этого диалога следует ввести образец для поиска или выбрать один из образцов, которые были использованы ранее в операциях поиска из списка истории.

В поле **Заменить** нужно ввести текст, на который следует произвести замену найденного текста, или выбрать один из образцов, которые были использованы ранее в операциях замены из списка истории.

Чтобы различать при поиске прописные и строчные буквы, требуется установить флажок **Учитывать регистр**. При установленном флажке **Искать слово целиком** будут найдены только целые слова, а не части слов.

Если не требуется открытия редакторов при групповой замене (по кнопке **Заменить все**), то нужно установить флажок **Не открывать редакторы при групповой замене**. При любом состоянии флажка редактор будет открываться при нажатии клавиши **Искать** или **Заменить**.

Ниже расположена панель, на закладках которой указывается, где следует искать указанный образец.

На закладке **Типы текстов** помечаются типы объектов, в которых будет произведен поиск. Если конфигурация редактируется для режима запуска **Управляемое приложение** (см. [здесь](#)), то из списка объектов будут исключены пользовательские интерфейсы.

На закладке **Конфигурации** можно указать с точностью до объекта разделы конфигураций, в которых будет произведен поиск.

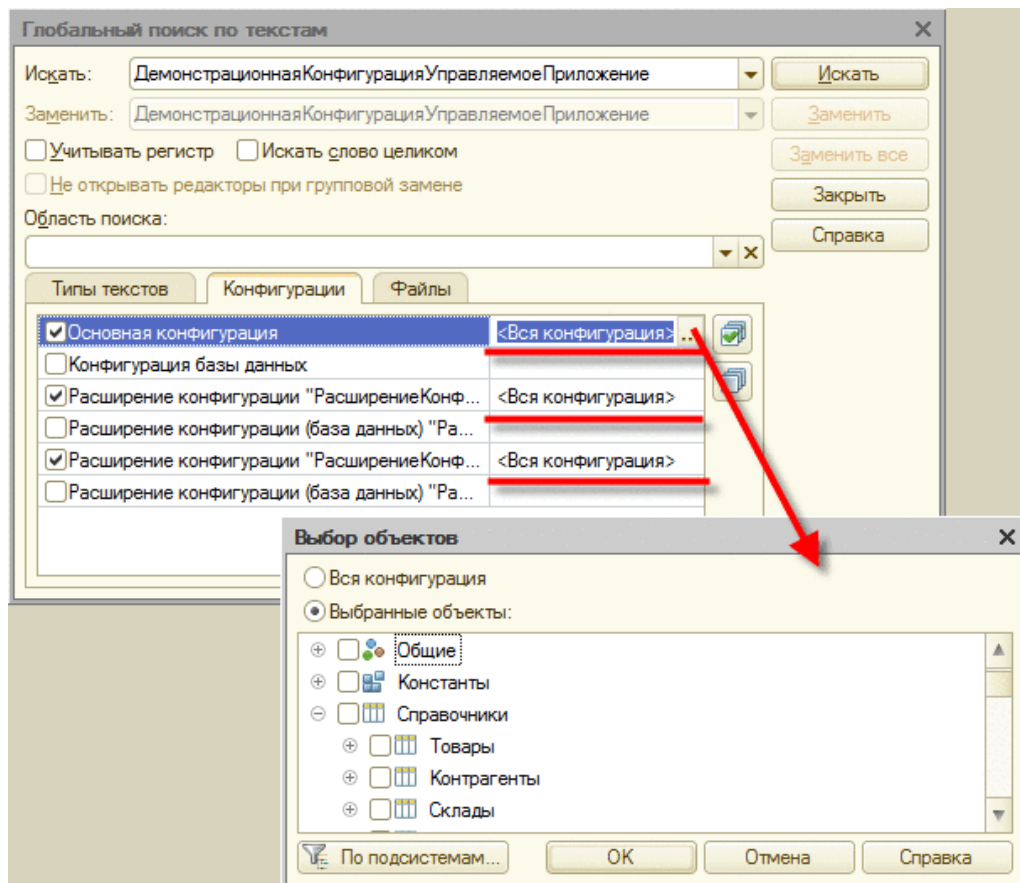


Рис. 34. Глобальный поиск

В список конфигураций помимо основной включается конфигурация базы данных, конфигурации хранилища, конфигурации расширений (если они открыты) и конфигурации расширений, сохраненных в базу данных (для открытых расширений). Конфигурации хранилища должны быть открыты перед вызовом режима поиска или замены. Конфигурации базы данных (основной и расширений) доступны только в случае использования глобального поиска.

Для указания набора объектов нужно установить переключатель **Выбранные объекты** и пометить те объекты, в которых будет произведен поиск. При первом запуске в списке по умолчанию установлены пометки всех объектов. Чтобы снять установку, следует снять флажок в строке с наименованием конфигурации. Затем можно указывать конкретные объекты для поиска.

На закладке **Файлы** можно указать каталог и типы файлов, в которых может производиться поиск. Могут быть просмотрены следующие типы просматриваемых файлов: конфигурации, расположенные в файлах (сохраненные, файлы поставки), внешние отчеты и обработки, текстовые и табличные документы. Если каталог не указан (реквизит **Каталог** не заполнен), то поиск в файлах не осуществляется. Поиск может также производиться в открытых документах тех же типов. Для этого следует установить флажок **Искать в открытых документах**.

Выборную совокупность настроек можно сохранить для дальнейшего использования. Для этого в поле **Область поиска** нужно указать имя настройки. Для использования прежней настройки достаточно выбрать имя настройки в выпадающем списке. Сохраняются следующие настройки: настройки на закладке **Типы текстов**, состав объектов только для основной конфигурации на закладке **Конфигурации** и настройки на закладке **Файлы**.

Если был запущен режим поиска, то для начала поиска нужно нажать кнопку **Искать**.

В режиме глобального поиска можно прервать процесс нажатием **Ctrl + Break**.

На экран в окно **Результаты поиска** будет выведен список найденных вхождений исходного текста.

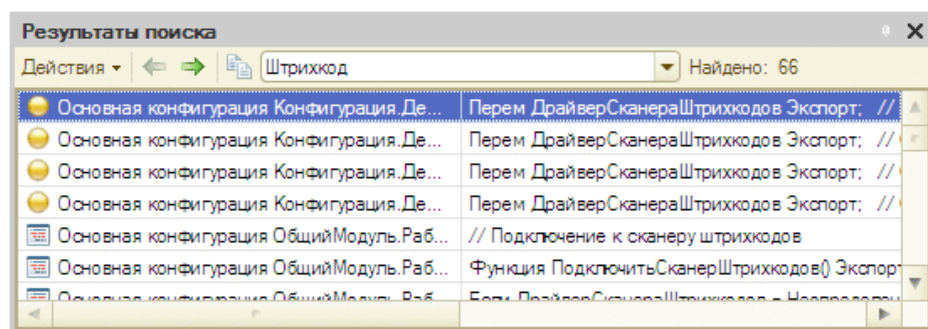


Рис. 35. Результат глобального поиска

Если какой-либо модуль имеет ограничение доступа (см. [здесь](#)), то перед поиском исходного текста в данном модуле система запрашивает пароль доступа. Необходимо ввести правильный пароль или отказаться от ввода пароля. Если пароль не введен, то просмотр в данном модуле не производится.

Результат поиска можно просмотреть, а к каждому найденному значению можно перейти, если выбрать в результате поиска нужную строчку и нажать клавишу **Enter**. Для просмотра следующего или предыдущего найденного значения можно воспользоваться пунктами [Действия – Следующая позиция](#) и [Действия – Предыдущая позиция](#).

Результат поиска (весь список) можно запомнить в буфер обмена с помощью команды [Копировать](#) контекстного меню окна или с помощью соответствующей кнопки панели инструментов окна результатов поиска, а также вывести в табличный или текстовый документ.

Ширину колонок можно изменить стандартным приемом – с помощью указателя мыши при нажатой клавише **Ctrl**.

Если был запущен режим замены, то в поле [На текст](#) указывается образец текста, на который следует заменить исходный текст, указанный в поле [Заменить](#).

Если перед заменой требуется посмотреть исходный текст, то для начала поиска следует нажать кнопку [Искать](#). На экран выводится результат первого найденного исходного текста. Если снова нажать кнопку [Искать](#), то текущий текст будет пропущен и на экран будет выведено очередное вхождение исходного текста в текущем окне или другое окно, содержащее исходный текст.

Групповая замена (без подтверждения каждой замены) будет выполнена при нажатии на кнопку [Заменить все](#). Если в этом случае не требуется открытия объектов, в которых обнаружено вхождение исходного текста, то следует установить флажок [Не открывать редакторы при групповой замене](#).

ВНИМАНИЕ! Во время просмотра результатов поиска изменить условия поиска нельзя.

Структура выбора области поиска (типы текста, список объектов конфигураций, файлов и открытые документы) запоминается и при следующем открытии диалога восстанавливается. Если требуется сохранить несколько областей, то каждой области в реквизите [Область поиска](#) нужно присвоить имя. При повторном открытии окна поиска в списке областей достаточно выбрать нужную и выполнить поиск.

2.22. Настройка рабочей области конфигуратора

Для создания и редактирования конфигурации требуется одновременное использование различных окон. Например, просмотр, выбор, добавление и удаление объектов конфигурации производятся в окне [Конфигурация](#); редактирование свойств объектов и их составных частей производится в палитре свойств; получение справочной информации по встроенному языку – в окне синтакс-помощника; сообщения и результаты поиска выводятся в окне сообщений. Кроме того, каждый объект конфигурации в общем случае может состоять из различных частей, каждая из которых редактируется в отдельном окне.

Одновременное открытие различных служебных окон заметно сужает рабочую область, предназначенную для редактирования прикладных объектов (формы, модули и макеты) и редактирования общих объектов конфигурации (модуль приложения и общие модули, макеты, стили, интерфейсы и др.).

Для расширения рабочей области и удобства работы можно воспользоваться некоторыми рекомендациями по настройке различных панелей конфигуратора, наличию и поведению различных служебных окон, использованию режимов показа окон.

2.22.1. Настройка панелей

Конфигуратор спроектирован таким образом, чтобы максимально использовать рабочую область за счет автоматического выбора нужных панелей инструментов для каждого вида окна. Так, при редактировании табличного документа конфигуратор предоставляет панель инструментов, предназначенную для выполнения команд по редактированию табличного документа; при переходе в окно, содержащее модуль формы, конфигуратор закрывает панель инструментов табличного документа и показывает панель инструментов текстового редактора.

Пользователь может самостоятельно настроить состав панелей инструментов и их размещение на экране. Если при настройке стандартной командной панели на ней размещают команды, которые отсутствуют на этой панели по умолчанию, то возможна ситуация, когда после перезапуска конфигуратора порядок команд на командной панели будет изменен. В этом случае рекомендуется скрыть стандартную панель, а вместо нее создать дополнительную командную панель, разместив на ней требуемые команды в нужном порядке. Порядок команд на ней изменяться не будет.

Помимо панелей инструментов внизу располагается [Панель окон](#) и [Панель состояния](#). Каждая из них занимает отдельную строку и не может быть перемещена в другое место. Если какая-либо панель не нужна, то ее можно скрыть с помощью контекстного меню в любом месте любой панели. Панель показывается, если слева от ее названия установлен флажок, и скрыта, если флажок снят. Для изменения режима показа панели достаточно в контекстном меню выбрать строку с наименованием панели.

Чтобы панель окон не занимала постоянно часть рабочей области программы, можно установить режим [Автоматически прятать](#). При работе панель окон скрыта. Чтобы она появилась, нужно подвести указатель мыши к месту размещения панели.

2.22.2. Окно «Конфигурация»

Окно [Конфигурация](#) может быть закрыто и открыто в любой момент времени. После выбора всех требуемых для работы окон окно [Конфигурация](#) может быть закрыто. Закрытие окна не приводит к окончанию работы с конфигурацией. Его можно открыть в любой момент.

Первоначальное состояние окна [Конфигурация – Прикрепленное](#). В этом состоянии оно «перекрывает» все окна, находящиеся в состоянии [Обычное](#) (большинство окон показываются в этом состоянии). Чтобы использовать область, занимаемую окном [Конфигурация](#), его состояние можно изменить на [Обычное](#) (в этом случае другие окна будут показываться поверх окна [Конфигурация](#)) или на [Прячущееся](#) (если окно не нужно, оно скрывается автоматически, а при подведении к нему указатель мыши раскрывается).

2.22.3. Использование режимов показа окон

В режиме [Обычный](#) окно может располагаться (его видно) только в пределах свободной рабочей области конфигуратора. Использование других режимов позволяет расширить рабочую область или рациональнее ее использовать. Кроме того, можно «вынести» окно за пределы рабочей области.

Каждое окно (кроме калькулятора) в конфигураторе может быть переведено в режим [Прячущееся](#). Этот режим позволяет без лишних действий выбрать нужное окно для просмотра и редактирования, переведя указатель мыши в строку с заголовком окна. Когда просмотр закончен, для сворачивания окна достаточно просто перевести указатель мыши на любое другое окно или заголовок другого прячущегося окна. Целесообразно в этом режиме использовать окна, работа с которыми носит кратковременный характер (синтакс-помощник, окно [Конфигурация](#), окно сообщений и результатов поиска, а также окно прикладных объектов, табличных и текстовых документов, открываемых в основном для просмотра).

Режим состояния окна [Свободное](#) позволяет поместить данное окно в любое место экрана, независимо от размеров и положения окна программы (конфигуратора).

В режиме [Прикрепляемое](#) окно может быть прикреплено к другому окну, находящемуся в этом состоянии, или к одной из сторон окна, а также расположено поверх другого прикрепляемого окна (совмещенные окна).

Рекомендуется совмещать такие окна, просмотр которых одновременно не требуется. Например, окно палитры свойств и окно синтакс-помощника или [Табло](#) и [Стек вызовов](#) во время отладки.

Подробнее о режимах показа окон см. [здесь](#).

Глава 20. Механизм полнотекстового поиска в данных

20.1. Общие сведения

Механизм полнотекстового поиска в данных системы «1С:Предприятие» позволяет осуществлять поиск в базе данных с указанием поисковых операторов (**И**, **ИЛИ**, **НЕ**, **РЯДОМ** и др.).

Механизм полнотекстового поиска основан на использовании двух составляющих:

- полнотекстового индекса, который создается в базе данных и затем периодически, по мере необходимости, обновляется;
- средств выполнения полнотекстового поиска.

Объектами полнотекстового поиска являются данные следующих объектов конфигурации:

- планы обмена,
- справочники,
- документы,
- планы видов характеристик,
- планы счетов,
- планы видов расчета,
- регистры сведений,
- регистры накопления,
- регистры бухгалтерии,
- регистры расчета,
- бизнес-процессы,
- задачи.

Для каждого из перечисленных объектов конфигурации реализовано свойство **Полнотекстовый поиск**, которое позволяет включать или исключать данные объекта в/из полнотекстового индексирования.

Для фиксации изменений объектов полнотекстового поиска система «1С:Предприятие» ведет журнал регистрации изменений. Запись в этот журнал осуществляется в процессе записи объектов в базу данных. В файлы регистрации изменений попадают только те объекты, которые настроены для полнотекстового индексирования. Если в результате отката транзакции запись объекта в базу данных будет отменена, то запись в журнале регистрации изменений останется.

Полнотекстовое индексирование выполняется в привилегированном режиме (в контексте сервера) и не требует монопольного захвата базы данных. В процессе полнотекстового индексирования выполняется чтение файлов регистрации изменений и получение измененных объектов из базы данных, а также транслитерация слов и замещение букв в словах, введенных латиницей, на кириллицу (при этом индекс хранит обе формы слова). Индексированию подлежат только реквизиты следующих типов:

- строка,
- дата,

- число,
- ссылочные типы,
- хранилище значения.

Для каждого объекта и реквизита в полнотекстовый индекс добавляется:

- имя объекта метаданных или его реквизита;
- синоним объекта метаданных или его реквизита (на всех языках конфигурации);
- представление объекта метаданных (на всех языках конфигурации).

Стандартные и пользовательские реквизиты индексируются на всех языках, что позволяет проводить поиск на всех языках конфигурации (например, русский и английский).

ПРИМЕЧАНИЕ. При полнотекстовом индексировании и поиске слова с буквой «ё» буквы «ё» заменяются на «е».

О расположении файлов индекса полнотекстового поиска можно прочитать в книге «1С:Предприятие 8.3. Руководство администратора».

В результате полнотекстового индексирования создается основной индекс, а в ходе последующего изменения данных базы данных – дополнительный, который содержит информацию по данным, измененным после последнего обновления основного индекса.

Поиск по основному индексу выполняется эффективно, в то время как поиск по дополнительному индексу выполняется медленнее. Поэтому в процессе индексирования предусмотрена возможность слияния индексов, которая позволяет добавить результат последних изменений данных к основному индексу. При этом следует учитывать, что эта операция может оказаться довольно длительной (если размер основного индекса большой), поэтому рекомендуется выполнять ее в периоды минимальной нагрузки на систему (в ночное время или по выходным).

Полнотекстовый поиск инициируется пользователем в контексте клиентского приложения, а выполняется в контексте сервера. Это означает, что в файловом варианте работы поиск будет выполняться на клиентском компьютере, а в клиент-серверном – в кластере серверов системы «1С:Предприятие».

Полнотекстовый поиск в данных осуществляется в соответствии с правами пользователя (в том числе и в соответствии с ограничениями прав доступа на уровне записей и полей базы данных). При этом поиск можно вести и по словам, которые были введены в базу данных с ошибками: например, если в слове вместо русской «с» стоит английская «с», если при наборе слова была нечаянно сменена раскладка клавиатуры и в результате слово приняло вид, например, «системf» вместо «система», эти слова все равно попадут в результат поиска. Также следует помнить, что механизм полнотекстового поиска в данных осуществляется всегда с начала слова.

Результаты поиска возвращаются частями, размер которых определяется при выполнении команды полнотекстового поиска.

Ранжирование полученных результатов осуществляется в порядке следующих приоритетов:

- «вес» объекта метаданных: чем больше ссылок на этот объект в реквизитах других объектов, тем выше его «вес»;
- дата объекта (более новые объекты будут находиться в начале).

20.2. Использование механизма

Полнотекстовый поиск в данных осуществляется средствами встроенного языка.

Свойство глобального контекста `ПолнотекстовыйПоиск` возвращает менеджер полнотекстового поиска –

объекта типа `МенеджерПолнотекстовогоПоиска`.

Методы менеджера полнотекстового поиска позволяют:

- получить информацию о состоянии полнотекстового индекса;
- выполнить полнотекстовое индексирование;
- инициировать процесс выполнения полнотекстового поиска в данных.

Для получения информации о состоянии полнотекстового индекса предназначены следующие методы:

- `ПолучитьРежимПолнотекстовогоПоиска()` – возвращает значение `Истина`, если использование полнотекстового поиска разрешено, или `Ложь` – в противном случае;
- `ДатаАктуальности()` – дата последнего момента, когда были проиндексированы все данные и не было информации о новых объектах для индексирования;
- `ИндексАктуален()` – возвращает значение `Истина`, если индекс полнотекстового поиска полностью соответствует текущему состоянию информационной базы;
- `ОбновлениеИндексаЗавершено()` – возвращает значение `Истина`, если слияние полнотекстового индекса не требуется.

Для выполнения полнотекстового индексирования предназначены следующие методы:

- `УстановитьРежимПолнотекстовогоПоиска()` – устанавливает режим полнотекстового поиска (`Разрешить` или `Запретить`). Если поиск был запрещен, то вызов этого метода с параметром `Истина` автоматически очищает имеющийся полнотекстовый индекс.
- `ОбновитьИндекс()` – обновляет индекс полнотекстового поиска. Если индексы отсутствовали, выполняет полную переиндексацию всей базы данных. В параметрах метода передаются условия индексирования:
 - `РазрешитьСлияние` – если передается значение `Истина`, будет выполнено слияние основного и дополнительного индексов.
 - `Порционное` – значение `Истина` указывает, что индексирование будет выполняться порциями из 10 тысяч объектов. После индексирования данных одной порции процесс завершается. Время индексирования одной порции сильно зависит от данных. Например, на типовой конфигурации «Управление производственным предприятием» занимает 3–5 минут.
 - `ОчиститьИндекс()` – удаляет все файлы полнотекстового индекса. Этот метод рекомендуется использовать в тех случаях, когда данные были полностью или почти полностью обновлены (например, при выполнении загрузки информационной базы). После очистки индекса нужно выполнить индексирование (если требуется).

Инициирование процесса полнотекстового поиска в данных выполняется с помощью метода `СоздатьСписок()`. В метод передаются два параметра:

- `СтрокаПоиска` – строка, содержащая поисковое выражение;
- `РазмерПорции` – число, задающее количество объектов, которые будут возвращены в одной части полнотекстового поиска.

Описание синтаксиса поисковых выражений см. [здесь](#).

Метод `СоздатьСписок()` возвращает объект `СписокПолнотекстовогоПоиска`, который позволяет выполнять полнотекстовый поиск и получать результаты полнотекстового поиска. Этот объект может быть использован многократно для выполнения поиска с различными условиями. Свойства объекта `СтрокаПоиска` и `РазмерПорции` позволяют изменять используемое поисковое выражение и размер порции получаемых данных.

Для выполнения полнотекстового поиска и получения «первых» результатов используется метод `ПерваяЧасть()`, который заполняет список первыми найденными элементами, в соответствии с размером порции. Для получения последующих результатов полнотекстового поиска используются методы `СледующаяЧасть()` и `ПредыдущаяЧасть()`, которые могут принимать в качестве параметра текущую начальную позицию и заполняют список полнотекстового поиска результатами поиска. Если параметр `ТекущаяНачальнаяПозиция` не указан, то используется имеющееся у объекта `СписокПолнотекстовогоПоиска` значение начальной позиции, которое, например, можно получить с помощью метода `НачальнаяПозиция()`. Использование параметра предпочтительнее, т. к. повышает скорость работы полнотекстового поиска.

Метод `Количество()` содержит количество элементов в текущей части (для последней части оно будет меньше либо равно размеру порции), а метод `ПолноеКоличество()` содержит полное количество элементов, найденных в результате полнотекстового поиска.

Метод `СледующаяЧасть()` заполняет список следующими элементами в соответствии со значением размера порции. При этом текущая позиция увеличивается на количество данных, содержащихся в полученной части. Если данных для получения очередной порции нет (достигнут конец данных), будет вызвано исключение, которое может быть обработано конструкцией

`Попытка ... Исключение ... КонецПопытки.`

Метод `ПредыдущаяЧасть()` заполняет список предыдущими найденными элементами в соответствии с размером порции. Если данных для получения очередной порции нет (достигнуто начало данных), будет вызвано исключение, которое может быть обработано конструкцией

`Попытка ... Исключение ... КонецПопытки.`

Метод `СлишкомМногоРезультатов()` возвращает значение `Истина`, если во время поиска, по соображениям производительности, было произведено усечение количества результатов, что может сказаться на точности поиска (не все объекты будут найдены). Анализ значения, возвращаемого этим методом, имеет смысл выполнять при получении последней порции найденных данных для информирования пользователя о том, что получены не все результаты, которые содержатся в базе данных.

Свойство `ПолучатьОписание` содержит признак получения описания для результатов поиска. Если он установлен в значение `Истина`, то для каждого из результатов поиска будет заполнено значение `Описание`, помогающее понять контекст найденных слов. В то же время установка этого свойства в значение `Ложь` ускоряет поиск.

Список полнотекстового поиска представляет собой коллекцию элементов списка полнотекстового поиска, которую можно обойти с помощью оператора `Для Каждого ... Из ... Цикл.`

Каждый элемент списка полнотекстового поиска представляется объектом `ЭлементСпискаПолнотекстовогоПоиска` и содержит следующие свойства:

- **Значение** – идентифицирует данные (объект или набор записей), в которых найдено поисковое выражение;
- **Метаданные** – объект метаданных, описывающий те данные, в которых найдено поисковое выражение;
- **Представление** – текстовое представление найденного объекта;
- **Описание** – содержит (с новой строки) пары `<реквизит> : <значение>`, где:
 - `<реквизит>` – реквизит объекта, в значении которого найдено выражение поиска;
 - `<значение>` – значение этого реквизита.

Результат поиска с помощью метода `ПолучитьОтображение()` можно представить в виде объекта `ЧтениеXML` или в виде строки с текстом в формате HTML, в котором средствами HTML найденные слова подсвечены (жирный шрифт и цвет фона).

20.3. Использование дополнительных словарей

Дополнительные словари морфологии и синонимов для полнотекстового поиска расширяют системные словари и могут содержать в себе особые термины и слова, которые используются при работе с конфигурацией.

В качестве дополнительных словарей могут выступать макеты из двоичных данных и текстовые макеты, а также константы строкового типа и типа [ХранилищеЗначения](#). Указание дополнительных словарей выполняется в свойстве [Дополнительные словари](#) корневого объекта метаданных.

При этом они должны иметь следующее содержание:

[Копировать в буфер обмена](#)

```
<?xml version="1.0"?>
<Dictionary>
  <Words>
    <lemma>рунет</lemma><forms>рунета рунете рунетом</forms>
    <lemma>Ванесса</lemma><forms>Ванессе Ванессы Ванессу</forms>
  </Words>
  <Synonyms>
    <item>ошибка баг сбой</item>
    <item>стрим поток</item>
  </Synonyms>
</Dictionary>
```

Элемент Dictionary

Предназначен для хранения собственно словаря. Словарь может содержать две секции:

- дополнительные слова (леммы) и их формы;
- ряды синонимов.

Элемент Words

Содержит слова в следующем виде:

- в элементе `lemma` слово хранится в его основной форме (именительном падеже);
- в элементе `forms` содержатся падежные формы слова.

По умолчанию нечеткий поиск не производится. Для того чтобы выполнить нечеткий поиск, следует использовать оператор «*». Пример: поиск `руне*` найдет и `рунет`, и `рунета`, и `рунете`, и `рунетом`.

Элемент Synonyms

Хранятся наборы синонимов. Каждый набор заключается в теги `item`.

Для того чтобы синонимы участвовали в полнотекстовом поиске, следует использовать оператор «!». При этом поиск `!ошибка` найдет и `ошибка`, и `баг`, и `сбой`. Описание синтаксиса поисковых выражений см. [здесь](#).

Элементы `Synonyms` и `Words` могут идти в произвольном порядке.

Словари загружаются системой при первом вызове поиска или индексировании. Если в макете содержатся ошибки, то заполнение словарей будет остановлено на месте ошибки.

В случае изменения дополнительного словаря, необходимо выполнить перезапуск системы, чтобы новый словарь был подключен системой. В зависимости от варианта системы, перезапущены должны быть разные компоненты:

- файловый вариант информационной базы с прямым подключением – необходимо перезапустить все клиентские приложения, подключенные к информационной базе;

- файловый вариант с подключением через веб-сервер – необходимо перезапустить расширение веб-сервера (фактически перезапуска требует веб-сервер);
- клиент-серверный вариант (с любым видом подключения) – необходимо перезапустить сервер «1С:Предприятия».

При этом индекс автоматически не обновляется, и нужно перестроить индекс вручную, хотя при поиске используются новые словари.

20.4. Мобильная платформа

Полнотекстовый поиск на мобильной платформе имеет некоторые особенности и ограничения.

Полнотекстовый поиск не поддерживает:

- Поиск с учетом морфологии.
- Подключение дополнительных словарей.
- Операцию извлечения текста (объект `ИзвлечениеТекст`).

При использовании полнотекстового поиска на мобильной платформе следует помнить о следующих особенностях:

- Обновление индекса полнотекстового поиска выполняется автоматически, после создания или изменения объекта конфигурации, для которых установлено использование полнотекстового поиска.
- Для обновления индекса полнотекстового поиска создается специальное фоновое задание. Фоновое задание создается в том случае, если не выполняется серверный вызов и не выполняется другое фоновое задание. Обновление выполняется порциями по 10 000 объектов.
- В том случае, если все объекты в информационной базе проиндексированы, то мобильная платформа запускает слияние индексов по истечении 20 секунд бездействия.

Если требуется гарантированно выполнить индексирование данных, например, после синхронизации информационной базы с информационной базой на персональном компьютере, то следует явным образом выполнить обновление индекса полнотекстового поиска с помощью метода

`ОбновитьИндекс(Истина, Ложь)` менеджера полнотекстового поиска.

Глава 21. Механизм временного хранилища, работа с файлами и картинками

21.1. Временное хранилище

В «1С:Предприятии» существует механизм работы с временным хранилищем, обеспечивающий хранение некоторых данных, привязанных к сеансу. Кроме того, реализован механизм работы с файлами, который обеспечивает обмен файлами между информационной базой и клиентским приложением. Особенностью данного механизма является то, что он ориентирован на использование в тонком клиенте и веб-клиенте и разработан с учетом ограничений на работу с файлами, накладываемых веб-браузерами.

Временное хранилище – это специализированное хранилище информации, в которое может быть помещено значение. Основное назначение – это временное хранение информации при клиент-серверном взаимодействии до ее переноса в базу данных. Механизм временного хранилища совместно с механизмом работы с файлами предоставляет набор, с помощью которого можно поместить данные, хранящиеся локально у пользователя, во временное хранилище информационной базы, перенести эту информацию из временного хранилища в базу данных и получить ее обратно на компьютер пользователя. Наиболее распространенные прикладные задачи, решаемые этими механизмами, – это хранение сопроводительной информации, например, изображений товаров, связанных с договорами документов и т. п. Механизмы временного хранилища и работы с файлами часто используются совместно, но могут использоваться и по отдельности.

Необходимость во временном хранилище возникает, например, из-за того, что в модели работы веб-браузера требуется передать выбранный пользователем файл непосредственно на сервер без возможности его хранения на клиенте. При передаче файла он помещается во временное хранилище и потом может быть использован при записи объекта в базу данных.

Можно использовать временное хранилище как универсальное хранилище с контролируемым временем жизни данных:

- Если при помещении данных во временное хранилище была выполнена привязка данных к некоторой форме, то время жизни сохраненных данных зависит от продолжительности жизни формы, к которой привязаны данные. При удалении объекта формы временное хранилище будет очищено от всей связанной с ней информации.
- Если при помещении данных во временное хранилище привязка к форме не была выполнена, то очистка временного хранилища будет выполнена в следующих случаях:
 - При следующем запросе формы.
 - При следующем серверном вызове из клиентского общего модуля.
 - При контекстном и неконтекстном клиентских вызовах из формы.
 - При серверном вызове из модуля команды. Если вызов сервера осуществляется для помещения значения во временное хранилище, то очистка не производится. Очистка производится после того, как вызов закончил свою работу.

То есть можно поместить одно или несколько значений во временное хранилище, а в следующем вызове это значение использовать. При этом после использования и перед тем, как серверный вызов будет окончен, помещенное значение будет автоматически удалено.

Наиболее типичная прикладная задача, решаемая временным хранилищем, – обеспечение доступа к файлам или картинкам до того, как объект будет записан в информационную базу, например, в форме элемента.

Данные, помещенные в хранилище, идентифицируются уникальным адресом, который в дальнейшем можно использовать в операциях записи, чтения или удаления. Этот адрес выдают методы записи значения во временное хранилище. Отдельный метод во встроенном языке позволяет определить, является ли переданный адрес адресом, указывающим на данные во временном хранилище.

Временное хранилище, сформированное в одном сеансе, недоступно из другого сеанса. Исключением является возможность передачи данных из фонового задания в сеанс, инициировавший это фоновое задание, с помощью временного хранилища (подробнее см. [здесь](#)).

21.2. Особенности работы с файлами в асинхронном режиме и веб-клиенте

Работа с файлами в асинхронном режиме присутствует только на стороне клиентского приложения. В общем случае рекомендуется использовать синхронные методы для работы на стороне сервера и асинхронные – для работы на стороне клиентского приложения.

Рассмотрим пример работы с файлами в асинхронной технике: удаление всех файлов в каталоге временных файлов. Для синхронной техники такое действие будет выглядеть следующим образом:

[Копировать в буфер обмена](#)

```
&НаКлиенте
Процедура Удалить(Команда)
    УдалитьФайлы(КаталогВременныхФайлов(), ПолучитьМаскуВсеФайлыКлиента());
    Сообщить("Удаление успешно завершено");
КонецПроцедуры
```

Асинхронная техника будет выглядеть по-другому:

[Копировать в буфер обмена](#)

```
&НаКлиенте
Процедура Удалить(Команда)
    ОбратныйВызов = Новый ОписаниеОповещения("ПолучитьКаталогВременныхФайловЗавершение", ЭтотОбъект, , "УдалитьОшибка", ЭтотОбъект);
    НачатьПолучениеКаталогаВременныхФайлов(ОбратныйВызов);
КонецПроцедуры
&НаКлиенте
Процедура ПолучитьКаталогВременныхФайловЗавершение(ИмяКаталогаВременныхФайлов, ДополнительныеПараметры) Экспорт
    ОбратныйВызов = Новый ОписаниеОповещения("УдалитьЗавершение", ЭтотОбъект, , "ФайловаяОшибка", ЭтотОбъект);
    НачатьУдалениеФайлов(ОбратныйВызов, ИмяКаталогаВременныхФайлов, ПолучитьМаскуВсеФайлыКлиента());
КонецПроцедуры
&НаКлиенте
Процедура УдалитьЗавершение(ДополнительныеПараметры) Экспорт
    Сообщить("Удаление успешно завершено");
КонецПроцедуры
&НаКлиенте
Процедура ФайловаяОшибка(ИнформацияОбОшибке, СтандартнаяОбработка, ДополнительныеПараметры) Экспорт
    Сообщить("При выполнении файловой операции обнаружена ошибка: " + КраткоеПредставлениеОшибки(ИнформацияОбОшибке));
КонецПроцедуры
```

Во-первых, следует обратить внимание, что практически все файловые операции стали асинхронными. Даже такие простые действия, как получение каталога временных файлов. И необходимость получения такого каталога приводит к первому асинхронному вызову.

Затем происходит вызов собственно удаления всех файлов. И обработка завершения удаления требует второй обработчик, в котором должны располагаться действия, которые должны происходить после удаления всех файлов в каталоге временных файлов. В данном примере это выдача сообщения о завершении операции.

Если при выполнении файловых операций произойдет ошибка – управление будет передано обработчику `ФайловыеОшибки()`. В нем можно корректно обработать сложившуюся ситуацию.

Работа с файлами в веб-клиенте имеет ряд особенностей:

1. Без установки расширений, в любом веб-браузере, доступны только методы `ПолучитьФайл()` и `НачатьПомещениеФайла()`. Метод `ПоместитьФайл()` не поддерживается при работе с веб-браузером Google Chrome и в том случае, если для конфигурации запрещено использование модальных методов (свойство конфигурации `Режим использования модальности`, см. [здесь](#)).
2. При установке расширения работы с файлами (см. [здесь](#)) во всех веб-браузерах, кроме Google Chrome, доступы как синхронные, так и асинхронные методы работы с файлами (синхронность или асинхронность зависит от свойства конфигурации `Режим использования модальности`, см. [здесь](#)). Соответствие методов см. [здесь](#).
3. Для веб-браузера Google Chrome доступно только асинхронное расширение работы с файлами. Синхронная работа не поддерживается.

ПРИМЕЧАНИЕ 1. Для корректной работы расширения работы с файлами в веб-браузере Microsoft Internet Explorer рекомендуется использование библиотеки Microsoft Core XML Services (MSXML) версий 4.0 или 6.0.

ПРИМЕЧАНИЕ 2. Расширение работы с файлами для веб-браузера Microsoft Internet Explorer устанавливается в каталог `%APPDATA%\1C\1CEWebExt\FileSystemExtIE`.

21.3. Способы работы с файлами и временным хранилищем

В данном разделе описаны наиболее частые варианты использования механизма работы с временным хранилищем и файлами.

21.3.1. Сохранение данных из файла во временное хранилище

21.3.1.1. Помещение одного файла

Метод `ПоместитьФайл()` помещает файл из локальной файловой системы во временное хранилище. Метод может принимать адрес во временном хранилище, по которому нужно сохранить файл. Если же адрес не определен или является пустой строкой, то будет создан новый адрес и метод вернет его через специальный параметр.

ПРИМЕЧАНИЕ. Объем данных, перемещаемых между клиентом и сервером, за один вызов метода `ПоместитьФайл()`, не должен превышать 4 Gb (в сериализованном виде).

Если параметр, определяющий интерактивный режим работы, равен `Истина`, то метод отобразит стандартное диалоговое окно выбора файла, в котором можно выбрать файл для помещения в хранилище. В этом случае метод также вернет адрес выбранного файла.

В качестве результата метод возвращает `Ложь`, если пользователь в интерактивном режиме отказался от совершения операции в диалоге выбора файла.

21.3.1.2. Помещение набора файлов

Метод `ПоместитьФайлы()` помещает во временное хранилище несколько файлов за один вызов. Существует несколько различных способов использования этого метода:

- предварительно сформировать список добавляемых файлов, например, когда предварительно выбрал файлы, которые необходимо поместить в информационную базу;
- передать в метод маску поиска файлов, например, когда требуется поместить в информационную базу все файлы определенного вида, например, все картинки;
- передать в метод предварительно подготовленный объект `ДиалогВыбораФайла` в режиме открытия файлов.

После окончания работы метод может вернуть список реально добавленных файлов.

Копировать в буфер обмена

```
// СписокФайлов – реквизит формы типа СписокЗначений,
// содержащий список добавляемых файлов
МассивФайлов = Новый Массив;
Для каждого ЭлементСписка Из СписокФайлов Цикл
    МассивФайлов.Добавить(Новый ОписаниеПередаваемогоФайла(ЭлементСписка, ));
КонiecЦикла;
ПомещенныеФайлы = Новый Массив;
Результат = ПоместитьФайлы(МассивФайлов, ПомещенныеФайлы, , Ложь, УникальныйИдентификатор);
```

ПРИМЕЧАНИЕ 1. Для использования метода `ПоместитьФайлы()` в веб-клиенте требуется подключение расширения работы с файлами (см. [здесь](#)).

ПРИМЕЧАНИЕ 2. Объем данных, перемещаемых между клиентом и сервером, за один вызов метода `ПоместитьФайлы()`, не должен превышать 4 Gb (в сериализованном виде).

21.3.2. Помещение данных во временное хранилище

Метод `ПоместитьВоВременноеХранилище()` схож с методом `ПоместитьФайл()`, за исключением того, что данные для записи во временное хранилище представляются не в виде пути в файловой системе, а в виде значения. Точно так же, если не указан существующий адрес во временном хранилище, создается новый адрес. Адрес возвращается как результат функции. Так же как и для файлов, размещенные данные обязательно принадлежат какой-либо форме и автоматически удаляются после ее удаления.

ПРИМЕЧАНИЕ. Объем данных, перемещаемых между клиентом и сервером, за один вызов метода `ПоместитьВоВременноеХранилище()`, не должен превышать 4 Gb (в сериализованном виде).

21.3.3. Получение данных из временного хранилища

При записи объекта в информационную базу может понадобиться извлечь данные из временного хранилища и поместить их, например, в реквизит объекта информационной базы. Для этого существует специальный метод – `ПолучитьИзВременногоХранилища()`. Этот метод извлекает данные из временного хранилища и возвращает их в качестве результата выполнения. Для получения данных необходимо указать адрес во временном хранилище. Этот адрес возвращают методы помещения данных во временное хранилище в случае их успешного выполнения (см. предыдущие разделы).

ВНИМАНИЕ! При получении на сервере значения из временного хранилища следует учитывать то, что оно получается по ссылке. В действительности, ссылка эта указывает на значение, которое хранится в кеше. В течение 20 минут, с момента помещения в хранилище или же с момента последнего обращения, значение сохранится в кеше, а затем записывается на диск и из кеша удаляется. При следующем обращении значение загружается с диска и снова помещается в кеш.

После десериализации и восстановления значения из временного хранилища ссылки не восстанавливаются. Значение в кеше восстанавливается с диска. Но после сериализации/десериализации восстановить ссылки на другие объекты внутри значения невозможно.

21.3.4. Удаление данных из временного хранилища

После того как данные сохранены в реквизите объекта информационной базы, данные во временном хранилище можно удалить. Для этого есть метод `УдалитьИзВременногоХранилища()`, который производит удаление. Метод принимает в параметре адрес во временном хранилище.

21.3.5. Проверка адреса на принадлежность временному хранилищу

Адрес может указывать как на временное хранилище, так и на реквизит в информационной базе. Для проверки его типа существует метод `ЭтоАдресВременногоХранилища()`.

Он проверяет, что переданный адрес является адресом, указывающим на хранилище. Возвращает *Истина*, если адрес указывает на временное хранилище.

21.3.6. Получение адреса реквизита

После того как данные помещены в реквизит объекта информационной базы, может потребоваться получить доступ к ним с помощью файловых методов.

Но прежде чем получить данные, например из реквизита, необходимо получить адрес этого реквизита. Для этого существует метод `ПолучитьНавигационнуюСсылку()`.

Он может вернуть адрес значения в информационной базе по исходным параметрам. Для этого необходимо передать ключ объекта (это может быть как ссылка на объект, так и ключ записи регистра сведений) и имя реквизита. Если нужно получить адрес значения, хранимого в реквизите табличной части, то к имени реквизита в параметре, задающем имя реквизита, необходимо добавить имя табличной части и точку «.».

Например:

[Копировать в буфер обмена](#)

[Товары.Изображение](#)

21.3.7. Получение файла из информационной базы

21.3.7.1. Получение одного файла

Метод `ПолучитьФайл()` получает файл из информационной базы и сохраняет его в локальную файловую систему пользователя. Первый параметр определяет адрес файла в реквизите объекта информационной базы или во временном хранилище файлов. Сохранение не произойдет, если у пользователя, от имени которого выполняется операция, отсутствует право *Просмотр* на реквизит объекта информационной базы. Второй параметр определяет место сохранения получаемого файла. В неинтерактивном режиме необходимо указать путь. В интерактивном режиме параметр является опциональным.

По умолчанию метод выполняется в интерактивном режиме. Это значит, что будет сформировано диалоговое окно, в котором можно указать действие с полученным файлом: запустить его или сохранить в указанное пользователем место файловой системы. Если выбран интерактивный режим, а параметр *Имя файла* не указан, то операция открытия файла недоступна. Метод возвращает значение типа *Булево. Ложь* означает, что пользователь выбрал отмену операции в диалоговом окне сохранения файлов в интерактивном режиме.

21.3.7.2. Получение набора файлов

Метод `ПолучитьФайлы()` позволяет получить и сохранить в локальной файловой системе пользователя несколько файлов, хранящихся в информационной базе. В качестве параметра передается список выгружаемых файлов.

[Копировать в буфер обмена](#)

```
// СписокФайлов – список значений, содержащий ссылки на элементы
// справочника, в котором расположены выгружаемые файлы
// Представлением списка значений является имя выгружаемого файла
МассивФайлов = Новый Массив;
Для каждого ЭлементСписка Из СписокФайлов Цикл
    Файл = Новый Файл(Строка(ЭлементСписка.Значение));
    ПолучаемыйФайл = Новый ОписаниеПередаваемогоФайла;
    ПолучаемыйФайл.Имя = ЭлементСписка.Представление;
    ПолучаемыйФайл.Хранение = ПолучитьНавигационнуюСсылку(ЭлементСписка.Значение, "Данные");
    МассивФайлов.Добавить(ПолучаемыйФайл);
КонецЦикла;
ПолученныеФайлы = Новый Массив;
Результат = ПолучитьФайлы(МассивФайлов, ПолученныеФайлы, ПутьВыгружаемыхФайлов, Ложь);
Если НЕ Результат Тогда
    Сообщение = Новый СообщениеПользователю;
    Сообщение.Текст = "Ошибка получения файлов!";
    Сообщение.Сообщить();
КонецЕсли;
```

После окончания работы метод может вернуть список реально выгруженных файлов с указанием полного имени каждого сохраненного файла.

ПРИМЕЧАНИЕ. Для использования метода `ПолучитьФайлы()` в веб-клиенте требуется подключение расширения работы с файлами.

В случае если свойство `Имя` объекта `ОписаниеПередаваемогоФайла` содержит абсолютный путь к файлу, файл будет сохранен по этому пути, без учета параметра `РасположениеФайлов`.

В качестве параметра `РасположениеФайлов` может выступать или путь в локальной файловой системе, или объект `ДиалогВыбораФайлов` в режиме выбора каталога или сохранения файлов. Если в качестве значения параметра `РасположениеФайлов` указан объект `ДиалогВыбораФайлов` в режиме сохранения файлов, то:

- диалог будет вызываться для каждого передаваемого файла, кроме тех файлов, для которых свойство `Имя` объекта `ОписаниеПередаваемогоФайла` содержит абсолютный путь;
- в качестве начального имени файла в диалоге будет использоваться значение свойства `Имя` объекта `ОписаниеПередаваемогоФайла`;
- если для какого-либо файла будет выбран отказ от сохранения, то свойство `Имя` объекта `ОписаниеПереданногоФайла` будет содержать пустую строку;
- метод `ПолучитьФайлы()` вернет значение `Истина`, если хотя бы один файл получен успешно;
- следует учитывать, что собственно получение файлов выполняется после того, как пользователь ответит на вопросы по указанию имени и пути всех получаемых файлов;
- если у пользователя, от имени которого выполняется метод `ПолучитьФайлы()`, отсутствует право `Просмотр` хотя бы на один реквизит объекта информационной базы, откуда выполняется получение файлов – неудачей завершится вся операция.

Пример:

[Копировать в буфер обмена](#)

```
ПередаваемыеФайлы = Новый Массив;
Описание = Новый ОписаниеПередаваемогоФайла("Описание", АдресФайла);
ПередаваемыеФайлы.Добавить(Описание);
ВыборФайла = Новый ДиалогВыбораФайла(РежимДиалогаВыбораФайла.Сохранение);
ВыборФайла.Заголовок = "Сохранение архива";
ВыборФайла.Расширение = "zip";
ВыборФайла.Фильтр = "Архив(*.zip)|*.zip|Все файлы|*.*";
ВыборФайла.ИндексФильтра = 0;
ПолучитьФайлы(ПередаваемыеФайлы, , ВыборФайла, Ложь);
```

Если выбран интерактивный режим выбора каталога сохранения файлов, то веб-клиент будет дополнительно запрашивать разрешение на сохранение файлов, указанных с абсолютными путями. В случае неинтерактивного сохранения (путь к каталогу указан в соответствующем параметре) запрос будет выполнен для всего списка сохраняемых файлов.

21.3.8. Пример использования файловых методов

[Копировать в буфер обмена](#)

```
// Получение в интерактивном режиме файла с диска
// и помещение его во временное хранилище.
&НаКлиенте
Процедура ВыбратьФайлСДискаИЗаписать()
    Переименовать ВыбранноеИмя;
    Переименовать АдресВременногоХранилища;
    НовыйОбъект = Объект.Ссылка.Пустая();
    Если ПоместитьФайл(АдресВременногоХранилища, "", ВыбранноеИмя, Истина) Тогда
        Объект.ИмяФайла = ВыбранноеИмя;
        ПоместитьФайлОбъекта(АдресВременногоХранилища);
    КонецЕсли;
КонецПроцедуры
// Копирование файла из временного хранилища в реквизит
// справочника, запись объекта, удаление файла из временного
// хранилища.
&НаСервере
Процедура ПоместитьФайлОбъекта(АдресВременногоХранилища)
    ЭлементСправочника = РеквизитФормыВЗначение("Объект");
    ДвоичныеДанные = ПолучитьИзВременногоХранилища(АдресВременногоХранилища);
    ЭлементСправочника.ДанныеФайла = Новый ХранилищеЗначения(ДвоичныеДанные, Новый СжатиеДанных());
    Файл = Новый Файл(ЭлементСправочника.ИмяФайла);
    ЭлементСправочника.ИмяФайла = Файл.Имя;
    ЭлементСправочника.Записать();
    Модифицированность = Ложь;
    УдалитьИзВременногоХранилища(АдресВременногоХранилища);
    ЗначениеВРеквизитФормы(ЭлементСправочника, "Объект");
КонецПроцедуры
// Считывание файла из реквизита и сохранение его
// на локальном диске в интерактивном режиме.
&НаКлиенте
Процедура ПрочитатьФайлИСохранитьНаДиск()
    Адрес = ПолучитьНавигационнуюСсылку(Объект.Ссылка, "ДанныеФайла");
    ПолучитьФайл(Адрес, Объект.ИмяФайла, Истина);
КонецПроцедуры
```

21.3.9. Разрешение на выполнение группы операций с файлами

При выполнении некоторых операций в веб-клиенте может потребоваться получить разрешение на несколько операций по работе с файлами. Например, необходимо выполнить получение документа из информационной базы и затем открыть сохраненный документ с помощью ассоциированного приложения.

Для выполнения этой операции потребуются отвечать на вопрос о сохранении документа и на вопрос о необходимости запуска. Если операций будет больше, вопросов пользователю также станет больше.

Для уменьшения количества вопросов можно воспользоваться методом `ЗапроситьРазрешениеПользователя()`. При использовании этого метода пользователю отображается список всех операций, которые планируется выполнить, и предлагается разрешить выполнение группы операций. Если пользователь разрешил выполнение, то запрошенные операции будут выполняться без дополнительных запросов пользователю. Если разрешение не предоставлено, операции будут выполняться в обычном режиме: один запрос на одну операцию.

ПРИМЕЧАНИЕ. Для использования метода `ЗапроситьРазрешениеПользователя()` в веб-клиенте требуется подключить расширение работы с

Рассмотрим пример использования метода:

[Копировать в буфер обмена](#)

```
Если ПодключитьРасширениеРаботыСФайлами() Тогда
    Ссылка = ПолучитьНавигационнуюСсылку(Объект.Ссылка, "ДанныеФайла");
    // Формирование описания передаваемых файлов (в данном случае файл один)
    ПередаваемыеФайлы = Новый Массив;
    Описание = Новый ОписаниеПередаваемогоФайла(Объект.ИмяФайла, Ссылка);
    ПередаваемыеФайлы.Добавить(Описание);
    // Готовим объект для получения информации о полученных файлах
    ПереданныеФайлы = Новый Массив;
    // Определяем прочие параметры методов
    КаталогСохранения = "c:\temp";
    Интерактивно = Ложь;
    ИмяПереданногоФайла = КаталогСохранения + "\" + Объект.ИмяФайла;
    // Готовим описание методов для получения разрешений
    Методы = Новый Массив;
    Методы.Добавить(Новый Массив);
    Методы[0].Добавить("ПолучитьФайлы");
    Методы[0].Добавить(ПередаваемыеФайлы);
    Методы[0].Добавить(ПереданныеФайлы);
    Методы[0].Добавить(КаталогСохранения);
    Методы[0].Добавить(Интерактивно);
    Методы.Добавить(Новый Массив);
    Методы[1].Добавить("ЗапуститьПриложение");
    Методы[1].Добавить(ИмяПереданногоФайла);
    Если Не ЗапроситьРазрешениеПользователя(Методы) Тогда
        Предупреждение("Пользователь отказал в разрешении.");
        Возврат;
    КонецЕсли;
    ПолучитьФайлы(ПередаваемыеФайлы, ПереданныеФайлы, КаталогСохранения, Интерактивно);
    ЗапуститьПриложение(ИмяПереданногоФайла);
Иначе
    Предупреждение("Выполнение не поддерживается. Не установлено расширение работы с файлами.");
КонецЕсли;
```

Следует отметить несколько особенностей применения метода `ЗапроситьРазрешениеПользователя()`.

- Разрешение запрашивается только для следующих методов:

- `ПолучитьФайлы()`,
- `ПоместитьФайлы()`,
- `НайтиФайлы()`,
- `ЗапуститьПриложение()`,
- `КопироватьФайл()`,
- `ПереместитьФайл()`,
- `УдалитьФайлы()`.

- Разрешение запрашивается для конкретного набора параметров метода. Если при реальном исполнении метода работы с файлами значения параметров будут отличаться от тех, для которых получено разрешение, это разрешение не будет действовать и пользователь получит отдельный запрос на подтверждение выполнения операции.

- Если необходимо выполнить две (или более) одинаковые операции с файлами (даже с одинаковым набором параметров), следует указать соответствующее количество элементов в массиве параметров метода `ЗапроситьРазрешениеПользователя()`. Например, если необходимо дважды получить из информационной базы один и тот же файл и поместить его в фиксированное место файловой системы, следует запросить разрешение на две операции.

- Если запрашивается разрешение на операцию, при выполнении которой выполняется интерактивная операция (например, функции `ПолучитьФайлы()` в качестве параметра передается объект `ДиалогВыбораФайла`), то такая операция исключается из запроса.

Полученные разрешения сохраняются либо до выполнения разрешенного вызова, либо до окончания исполнения встроенного языка.

ПРИМЕЧАНИЕ. В толстом и тонком клиентах `ЗапроситьРазрешениеПользователя()` метод всегда возвращает значение `Истина`, без взаимодействия с пользователем.

21.3.10. Работа с временным хранилищем в фоновом задании

В механизме работы с временным хранилищем есть возможность передать данные из фоновое задания в сеанс, инициировавший фоновое задание. Для такой передачи следует в родительском сеансе поместить во временное хранилище пустое значение (с помощью метода `ПоместитьВоВременноеХранилище()`), указав какой-либо идентификатор создаваемого временного хранилища (параметр `Адрес`). Затем полученный адрес передать в фоновое задание через параметры фонового задания. Далее, если в фоновом задании этот адрес использовать в качестве значения параметра `Адрес` метода `ПоместитьВоВременноеХранилище()`, то результат будет скопирован в сеанс, из которого было запущено фоновое задание.

Данные, помещенные во временное хранилище в фоновом задании, не будут доступны из родительского сеанса до момента завершения фонового задания.

21.3.11. Поддержка адресов в поле картинки

Элемент формы `Поле` вида `Поле картинки` поддерживает отображение картинки, заданной адресом значения (которое может быть картинкой или двоичными данными) во временном хранилище или в базе данных.

Для этого в свойстве `Данные` элемента формы необходимо задать реквизит строкового типа. Значение этого реквизита и будет интерпретироваться как адрес картинки.

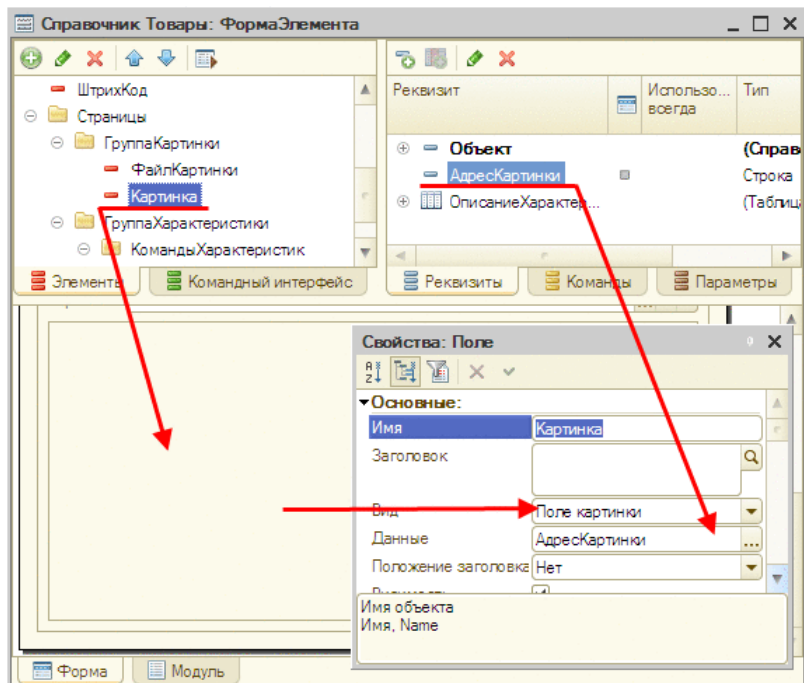


Рис. 341. Отображение картинки на форме

Копировать в буфер обмена

```
// Пример 1
// Привязка поля картинки к адресу картинки во временном
// хранилище. АдресКартинки – реквизит формы строкового типа
ПоместитьФайл(АдресКартинки, ИсходноеИмя, ВыбранноеИмя, Истина, УникальныйИдентификатор);
// Пример 2
// Получение адреса картинки из реквизита объекта
// информационной базы
ФайлКартинки = Объект.ФайлКартинки;
Если Не ФайлКартинки.Пустая() Тогда
    АдресКартинки = ПолучитьНавигационнуюСсылку(ФайлКартинки, "ДанныеФайла");
Иначе
    АдресКартинки = "";
Конечесли;
```

21.3.12. Доступ к стандартным каталогам

При использовании системы требуется некоторое место в файловой системе, где можно хранить различные файловые данные, которые с одной стороны являются временными, а с другой – должны храниться достаточно длительное время. К таким файлам можно отнести черновики для систем управления документооборотом, внешние компоненты, работающие на стороне клиентского компьютера и т. д.

Для хранения таких файлов предназначен специальный каталог, который привязан к конкретному пользователю конкретной информационной базы. Один и тот же пользователь, работающий с двумя информационными базами, будет иметь доступ к двум разным каталогам хранения данных пользователя. Местоположение данного каталога определяется с помощью метода `РабочийКаталогДанныхПользователя()`. Если каталог отсутствует, то он создается при первом обращении к нему. Если создание каталога невозможно – система вызывает исключение.

ПРИМЕЧАНИЕ. Метод `РабочийКаталогДанныхПользователя()` не доступен на стороне сервера.

В операционной системе имеется специальный каталог, выделенный для постоянного хранения данных пользователя. Это могут быть какие-то отчеты, печатные формы документов и т. д. В этот каталог помещаются данные, которые в дальнейшем могут быть отправлены внешним потребителям. Для доступа к этому каталогу предназначен метод `КаталогДокументов()`. Физическое расположение каталога зависит от операционной системы, где исполняется приложение:

- ОС Windows – каталог документов текущего пользователя ОС.
- ОС Linux – домашний каталог текущего пользователя ОС.
- ОС MacOS – системный каталог документов.
- ОС MacOS X – каталог документов текущего пользователя.

Глава 22. Журнал регистрации

Для выполнения административных задач часто требуется выяснить, какие события происходили в определенный момент времени или какие действия выполнял тот или иной пользователь.

Для этих целей предназначен журнал регистрации. В этом журнале могут фиксироваться различные события. С его помощью администратор может получить историю работы пользователей с системой.

Журнал регистрации не хранится в базе данных и не сохраняется при операциях загрузки/выгрузки информационной базы.

При работе пользователей система «1С:Предприятие» фиксирует в журнале основные действия, выполняемые пользователем по модификации данных информационной базы, доступу и отказу доступа к данным, выполнению регламентных операций, подключению и отключению от системы и т. д.

Кроме средств интерактивной работы с журналом, которые предоставляются в конфигураторе, доступна также и программная работа с журналом.

В данной главе описываются средства программной работы с журналом регистрации.

22.1. Управление детализацией

Имеется возможность управления уровнем детализации сообщений, которые попадают в журнал регистрации. Для этого существуют методы `ПолучитьИспользованиеЖурналаРегистрации()` и `УстановитьИспользованиеЖурналаРегистрации()`.

[Копировать в буфер обмена](#)

```
// установим регистрацию всех событий журнала
Уровни = Новый Массив;
Уровни.Добавить(УровеньЖурналаРегистрации.Ошибка);
Уровни.Добавить(УровеньЖурналаРегистрации.Информация);
Уровни.Добавить(УровеньЖурналаРегистрации.Предупреждение);
Уровни.Добавить(УровеньЖурналаРегистрации.Примечание);
УстановитьИспользованиеЖурналаРегистрации(Уровни);
```

22.2. Запись событий

При выполнении различных операций возникает необходимость фиксировать выполняемые действия для последующего анализа. Для такой фиксации можно использовать журнал регистрации. Запись событий в него осуществляется с помощью метода глобального контекста `ЗаписатьЖурналаРегистрации()`.

ПРИМЕЧАНИЕ. С помощью данного метода невозможно записать в журнал регистрации системные события.

Имена пользовательских событий могут содержать точку в своих названиях, тем самым образуя группы пользовательских событий.

22.3. Управление регистрацией событий

Регистрацию любого события журнала регистрации можно включить или выключить. Для этого служат методы глобального контекста `ПолучитьИспользованиеСобытияЖурналаРегистрации()` и `УстановитьИспользованиеСобытияЖурналаРегистрации()` и объект `ИспользованиеСобытияЖурналаРегистрации`. Следующий пример показывает, как можно выключить регистрацию события `Ошибка выполнения` (`$_PerformError$`):

[Копировать в буфер обмена](#)

```
ИспользованиеСобытия = Новый ИспользованиеСобытияЖурналаРегистрации();
ИспользованиеСобытия.Использование = Ложь;
УстановитьИспользованиеСобытияЖурналаРегистрации("$_PerformError$", ИспользованиеСобытия);
```

Следует отметить, что существует два события, настройка регистрации которых имеет дополнительные

возможности. Это события **Доступ** (`_$Access$.Access`) и **Отказ в доступе** (`_$Access$.AccessDenied`). Работа с этими событиями будет описана более подробно.

ПРИМЕЧАНИЕ. События, связанные с транзакциями (`_$Transaction$.Begin`, `_$Transaction$.Commit`, `_$Transaction$.Rollback`), не могут быть отключены с помощью метода `УстановитьИспользованиеСобытияЖурналаРегистрации()`.

22.3.1. Настройка параметров события «Доступ»

Событие **Доступ** предназначено для регистрации фактов доступа пользователей системы к тем или иным данным. Для настройки регистрации события **Доступ** следует задать:

- необходимость регистрации события;
- перечень объектов метаданных, доступ к которым необходимо регистрировать;
- перечень полей объекта метаданных, чтение которых следует регистрировать (**поля доступа**);
- перечень полей объекта метаданных, значения которых необходимо зарегистрировать (**поля регистрации**).

Общая схема работы при этом выглядит следующим образом (при условии, что событие регистрируется). Если в процессе работы с данными (выбранного объекта метаданных) произошло чтение одного из заданных полей доступа, происходит фиксация в журнале регистрации полей регистрации по заданным правилам.

Объем данных, которые будут записаны в журнал регистрации, зависит от того, как настроена регистрация события:

Использование	Поля доступа	Поля регистрации	Результат
Не задано			Событие не регистрируется
Задано	Не заданы	Не заданы	Событие не регистрируется
Задано	Заданы	Не заданы	Событие регистрируется без детализации
Задано	Заданы	Заданы	Событие регистрируется с указанием полей регистрации

Событие формируется только в случае успешного чтения данных.

Рассмотрим настройки параметров на конкретном примере:

[Копировать в буфер обмена](#)

```

НастройкаСправочника = Новый ОписаниеИспользованияСобытияДоступЖурналаРегистрации();
// Укажем объект, доступ к которому будет регистрироваться
НастройкаСправочника.Объект = "Справочник.ФизическиеЛица";
// Укажем поля доступа
НастройкаСправочника.ПоляДоступа.Добавить("ПаспортныеДанные");
НастройкаСправочника.ПоляДоступа.Добавить("Дети.СвидетельствоОРождении");
// Укажем поля регистрации
НастройкаСправочника.ПоляРегистрации.Добавить("ПаспортныеДанные");
НастройкаСправочника.ПоляРегистрации.Добавить("Дети.ИмяРебенка");
АльтернативыПолей = Новый Массив();
АльтернативыПолей.Добавить("Фамилия");
АльтернативыПолей.Добавить("Имя");
НастройкаСправочника.ПоляРегистрации.Добавить(АльтернативыПолей);
НастройкаОбъектовМетаданных = Новый Массив();
НастройкаОбъектовМетаданных.Добавить(НастройкаСправочника);

```

В данном примере будет выполняться регистрация доступа к элементам справочника **ФизическиеЛица**.

При этом события журнала регистрации будут формироваться в том случае, если в запросе к данным справочника [ФизическиеЛица](#) будут участвовать:

- поле [ПаспортныеДанные](#);
- поле [СвидетельствоОРождении](#) табличной части [Дети](#).

Если в запросе к данным описанные поля не присутствуют, событие доступа к данным не будет записано в журнал регистрации.

После того как система определит доступ к контролируемым полям, в журнал регистрации будут записаны данные о следующих полях справочника [ФизическиеЛица](#):

- Поле [ПаспортныеДанные](#).
- Поле [ИмяРебенка](#) из табличной части [Дети](#).
- Какое-либо из полей [Фамилия](#) или [Имя](#). Какое из полей будет записано в журнал регистрации, определяется наличием этих данных в запросе к данным. Причем если используются все перечисленные поля, то в журнал регистрации попадет поле, заданное с меньшим индексом (в примере – [Фамилия](#)). Если из указанной пары в запросе используется какое-либо одно поле, в журнал регистрации попадет именно оно.

Рассмотрим еще один пример настройки события [Доступ](#):

[Копировать в буфер обмена](#)

```
НастройкаРегистраСведений = Новый ОписаниеИспользованияСобытияДоступЖурналаРегистрации( ) ;
// Укажем объект, доступ к которому будет регистрироваться
НастройкаРегистраСведений.Объект = "РегистрСведений.ОкладыСотрудников" ;
// Укажем поля доступа
НастройкаРегистраСведений.ПоляДоступа.Добавить( "Оклад" ) ;
// Укажем поля регистрации
НастройкаРегистраСведений.ПоляРегистрации.Добавить( "Сотрудник" ) ;
НастройкаОбъектовМетаданных.Добавить( НастройкаРегистраСведений ) ;
```

В данном примере при обращении к полю [РегистрСведений.ОкладыСотрудников.Оклад](#) будет происходить регистрация события [Доступ](#) со следующим составом поля [Данные.Данные](#):

- если в результате запроса будет поле [РегистрСведений.ОкладыСотрудников.Сотрудник](#) (и поле [Ссылка](#) задано в настройке полей регистрации справочника [ФизическиеЛица](#), на который ссылается поле [Сотрудник](#)), то в событие журнала регистрации будет записано:
 - в поле [Данные](#) будет записана таблица значений, содержащая колонку [СправочникФизическиеЛицаСсылка](#), в которой будут значения ссылок на объекты справочника [ФизическиеЛица](#);
 - в поле [Метаданные](#) будет записан массив, содержащий строку [РегистрСведений.ОкладыСотрудников](#), т. е. имя объекта, который послужил поводом регистрации события доступа;
 - в поле [ПредставлениеМетаданных](#) будет записан массив, содержащий строку [РегистрСведений.Окладысотрудников](#) – представление объекта метаданных;
- если поле [Ссылка](#) не участвует в полях регистрации справочника [ФизическиеЛица](#), то она не будет зафиксирована в запросе к регистру сведений;
- если в запросе будет использовано получение данных по ссылке типа [Справочник.ФизическиеЛица](#), хранящейся в регистре сведений, то регистрация будет применяться к полям, которые получены через ссылку, хранящуюся в поле регистра [Сотрудник](#):
 - [Имя](#), [Фамилия](#) и [ПаспортныеДанные](#) (все – полученные через ссылку, хранящуюся в поле регистра [Сотрудник](#)), то в журнал регистрации попадут поля [ПаспортныеДанные](#) и [Фамилия](#) (описание

регистрации для справочника **Сотрудники** можно посмотреть в предыдущем примере).

22.3.2. Настройка параметров события «Отказ в доступе»

Событие **Отказ в доступе** предназначено для регистрации фактов отказа в доступе к тем или иным данным пользователям системы. Для настройки регистрации события **Отказ в доступе**, следует задать:

- необходимость регистрации события;
- перечень объектов метаданных, для которых необходимо регистрировать поля регистрации при отказе в доступе (для остальных объектов отказы в доступе будут регистрироваться без детализации);
- перечень полей объекта метаданных, значения которых необходимо зарегистрировать (**поля регистрации**) при отказе в доступе.

Общая схема работы при этом выглядит следующим образом (при условии, что событие регистрируется). Если в процессе работы с данными (выбранного объекта метаданных) произошел отказ в доступе, происходит фиксация в журнале регистрации полей регистрации по заданным правилам.

Объем данных, которые будут записаны в журнал регистрации, зависит от того, как настроена регистрация события:

Использование	Поля регистрации	Результат
Не задано		Событие не регистрируется
Задано	Не заданы	Событие регистрируется без детализации (поле Данные не заполнено)
Задано	Заданы	Событие регистрируется с указанием полей регистрации в поле Данные

Событие формируется в следующих случаях:

- при проверке доступа к объекту данных целиком:
 - в случае нарушения прав при обращении к методам и свойствам прикладного объекта из встроенного языка или из стандартных интерфейсных функций (форм, команд);
 - в случае отказов в доступе при проверке прав на конфигурацию;
 - при возникновении события в поле **Данные** записи журнала будет помещена структура со свойством **Право**, содержащим действие, в выполнении которого было отказано;
- при проверке ограничений доступа к данным:
 - в случае нарушения условий ограничения доступа к данным;
 - при возникновении события в поле **Данные** записи журнала будет помещена структура, содержащая два свойства:
 - **Действие** – указано действие, при попытке выполнения которого произошел отказ в доступе;
 - **Данные** – содержит информацию о полях регистрации (если настроено).

ПРИМЕЧАНИЕ. При считывании данных во временную таблицу поле **Данные**.**Данные** события журнала регистрации не заполняется.

Рассмотрим настройки параметров на конкретном примере:

```

НастройкаСправочника = Новый ОписаниеИспользованияСобытияОтказВДоступеЖурналаРегистрации( );
// Укажем объект, доступ к которому будет регистрироваться
НастройкаСправочника.Объект = "Справочник.ФизическиеЛица";
// Укажем поля регистрации
НастройкаСправочника.ПоляРегистрации.Добавить ( "ПаспортныеДанные" );
НастройкаСправочника.ПоляРегистрации.Добавить ( "Дети.ИмяРебенка" );
АльтернативыПолей = Новый Массив( );
АльтернативыПолей.Добавить ( "Фамилия" );
АльтернативыПолей.Добавить ( "Имя" );
НастройкаСправочника.ПоляРегистрации.Добавить ( АльтернативыПолей );

```

При нарушении доступа к данным в операции **Чтение** будет происходить регистрация события **Отказ в доступе** и выполняться запись поля регистрации по правилам, описанным для полей регистрации события **Доступ**. Для других действий (**Изменение**, **Удаление**, **Добавление**) событие регистрируется, но запись данных не происходит.

22.4. Получение записей журнала регистрации

В процессе эксплуатации прикладного решения могут возникать ситуации, которые требуют программного анализа журнала регистрации, в том числе и получение событий журнала регистрации по некоторым критериям (отбору). Для программного получения записей журнала регистрации необходимо воспользоваться методом глобального контекста **ВыгрузитьЖурналРегистрации()**. Программно получать можно как записи журнала регистрации информационной базы, где выполняется код на встроенном языке, так и другой информационной базы (даже если с этой базой работают другие экземпляры «1С:Предприятия»), если у метода **ВыгрузитьЖурналРегистрации()** указан параметр **ИмяВходногоФайла**. Следует помнить, что такое чтение рекомендуется выполнять только в том случае, если файлы журнала регистрации другой информационной базы доступны локально для экземпляра «1С:Предприятия», выполняющего чтение журнала. Если такого доступа нет, то необходимо разрабатывать другие способы получения журнала регистрации.

Для задания условий отбора выборки необходимо воспользоваться параметром **Фильтр** метода **ВыгрузитьЖурналРегистрации()**. Данный параметр может содержать либо одиночное значение отбора, заданное с помощью объекта **Структура** особого формата (формат структуры подробно описан в синтаксис-помощнике: **Глобальный контекст – Процедуры и функции работы с журналом регистрации – ВыгрузитьЖурналРегистрации()**), либо массив таких структур.

При этом если отбор задан одним элементом типа **Структура**, то в результирующую выборку попадут все записи, которые удовлетворяют всем условиям, заданным в структуре (условие сочетается «по И»). Если отбор задан массивом, то в результирующую выборку попадут записи, которые удовлетворяют хотя бы одному условию, перечисленному в элементах массива (условия сочетаются «по ИЛИ»).

Обратим особое внимание на возможность задания отбора по полю **Данные** записи журнала регистрации. В качестве отбора по этому полю могут выступать значения типа **Строка**, **Число**, **Дата**, **Ссылка**, **Структура** (для событий **Доступ**, **Отказ в доступе**, **Аутентификация**, **Ошибка аутентификации**, **Добавление пользователя**, **Изменение пользователя** и **Удаление пользователя**), а также массив структур.

Значением элемента структуры, переданной в параметр **Фильтр**, может выступать конкретное значение, **Структура** или **Массив**. Проверка соответствия данных записи журнала на соответствие параметрам фильтра выполняется следующим образом:

- Если передано конкретное значение, то производится сравнение значений в данных записи и в значении фильтра.
- Если передан массив значений, то выполняется сравнение значения из данных записи журнала регистрации и всех значений массива, переданного в качестве значения фильтра. Условие считается выполненным, если значение, указанное в записи журнала, совпадает хотя бы с одним значением переданного массива.
- Если в качестве значения передана структура, то условие считается выполненным при условии, что для всех элементов структуры, переданной в качестве элемента фильтра, будет установлено соответствие с данными записи журнала регистрации.

Рассмотрим это подробнее.

1. В поле **Данные** записи журнала регистрации находится структура, в поле **Данные** фильтра также указана структура. В этом случае запись будет удовлетворять условию, если в поле **Данные** записи журнала присутствуют все ключи из одноименного поля фильтра и значения для этих ключей соответствуют друг другу.
2. В поле **Данные** записи журнала регистрации находится таблица значений, в поле **Данные** фильтра указана структура. В этом случае запись будет удовлетворять условию, если в таблице значений есть все колонки с именами, равными именам ключей структуры фильтра, и есть строки, в которых значения соответствующих колонок равны значениям структуры из фильтра.
3. В поле **Данные** записи журнала регистрации находится значение типа **Структура**, **Массив** или **ТаблицаЗначений**, в поле **Данные** фильтра указан массив. В этом случае запись будет удовлетворять условию, если в данных записи журнала будет найдено хотя бы одно значение из массива, переданного в качестве значения фильтра.
4. В поле **Данные** записи журнала регистрации находится какое-либо значение (не структура, не массив и не таблица значений), в поле **Данные** фильтра указан массив. В этом случае запись будет удовлетворять условию, если значение из записи журнала равно хотя бы одному значению из переданного массива.
5. В поле **Данные** записи журнала находится элемент со значением типа **ТаблицаЗначений**, в поле **Данные** фильтра находится элемент со значением типом **Массив**, при этом ключи элементов совпадают. В этом случае запись журнала регистрации будет удовлетворять условию, если хотя бы одна ячейка таблицы значений равна хотя бы одному значению из переданного массива.
6. В элементе поля **Данные** записи журнала находится значение типа **Структура**, **Массив** или **ТаблицаЗначений**, в поле **Данные** фильтра находится значение типа **Структура** или **Массив**. Запись журнала будет удовлетворять условию, если во вложенных данных будет обнаружена запись, удовлетворяющая условию по пунктам 1 и 5. Однако если в фильтре задана структура, то поиск элементов в структуре или колонках таблицы значений записи журнала регистрации, соответствующих элементам этой структуры, будет выполняться только на конкретном уровне вложенности, соответствующем уровню вложенности в фильтре. Таким образом, отдельные значения и значения из массивов в фильтре ищутся на всю глубину, а элементы структуры фильтра только на конкретном уровне.

Если при формировании таблицы значений, находящейся в поле **Данные** записи журнала регистрации, присутствуют колонки с одинаковым именем, то происходит объединение этих колонок в одну. При этом количество строк в этой таблице увеличивается таким образом, чтобы в результирующей колонке присутствовали все уникальные значения из одноименных колонок. Значения, указанные в остальных колонках, дублируются в создаваемых строках.

Покажем это на примере. Допустим, прикладное решение содержит запрос, который возвращает таблицу со следующими колонками:

- **СправочникСсылка.Товары**,
- **СправочникСсылка.Товары**,
- **Строка**.

При формировании записи журнала регистрации будет выполнено следующее преобразование:

- Первые две колонки будут объединены в одну.
- Для каждой строки, в которой значения **Товары.Ссылка** и **Номенклатура.Ссылка** не равны, будет создана еще одна строка.
- Строки будут отличаться значением объединенной колонки, где будут располагаться значения ссылок. Остальные значения колонок будут одинаковыми.

Исходная таблица

--	--	--

Ссылка	Ссылка	Артикул
Сосиски	Перец	16-AB-1675

Результирующая таблица

Ссылка	Артикул
Сосиски	16-AB-1675
Перец	16-AB-1675

Также приведем примеры, показывающие работу отборов при получении записей журнала регистрации:

Пример 1:

- значение поля **Данные** записи журнала регистрации для события **Аутентификация**:
 - тип: **Структура**:
 - ключ: **ПользовательОС**, значение: **Ivanov**;
- значение поля **Данные** фильтра метода **ВыгрузитьЖурналРегистрации()**:
 - тип: **Структура**:
 - ключ: **ПользовательОС**, значение: **Ivanov**;
- результат: запись соответствует.

Пример 2:

- значение поля **Данные** записи журнала регистрации для события **Доступ**:
 - тип: **Структура**:
 - ключ: **Данные**, значение: **ТаблицаЗначений**:

Фамилия	Город	Телефон
Петров	Тула	111-22-33
Иванов	Москва	222-33-44

- значение поля **Данные** фильтра метода **ВыгрузитьЖурналРегистрации()**:
 - тип: **Массив**:
 - элемент 1: тип **Структура**:
 - ключ: **ПользовательОС**, значение: **Ivanov**;
 - элемент 2: тип **Структура**:
 - ключ: **Данные**, значение: **Структура**;
 - ключ: **Фамилия**, значение: **Иванов**;
 - ключ: **Город**, значение: **Москва**;

- результат: запись соответствует отбору, т. к. при сравнении элемента 2 массива отбора найдено соответствие с данными записи (строка 2). В записи журнала регистрации есть таблица значений, у которой существуют колонки *Имя* и *Город* и оба значения в колонках равны соответствующим значениям в структуре отбора.

Пример 3:

- значение поля *Данные* записи журнала регистрации для события *Добавление пользователя*:
 - тип: *Структура*:
 - ключ: *Роли*, значение: *Массив*:
 - значение 1: *Роли.Администратор*;
 - значение 2: *Роли.Кладовщик*;
 - значение 3: *Роли.Продавец*;
 - значение поля *Данные* фильтра метода *ВыгрузитьЖурналРегистрации()*:
 - тип: *Структура*:
 - ключ: *Роли*, значение: *Роли.Кладовщик*;
 - результат: запись соответствует отбору, т. к. в поле *Данные* существует элемент *Роли*, а среди значений элемента есть значение *Роли.Кладовщик*.

Пример 4:

- значение поля *Данные* записи журнала регистрации для события *Добавление пользователя*:
 - тип: *Структура*:
 - ключ: *Роли*, значение: *Массив*:
 - значение 1: *Роли.Администратор*;
 - значение 2: *Роли.Кладовщик*;
 - значение 3: *Роли.Продавец*;
 - значение поля *Данные* фильтра метода *ВыгрузитьЖурналРегистрации()*:
 - тип: *Структура*:
 - ключ: *Роли*, значение: *Массив*:
 - значение 1: *Роли.Менеджер*;
 - значение 2: *Роли.Продавец*;
 - результат: запись соответствует отбору, т. к. в поле *Данные* существует элемент *Роли*, а среди значений элемента есть значение, перечисленное в массиве отбора (значение *Роли.Продавец*).

Пример 5:

- значение поля *Данные* записи журнала регистрации для события *Доступ*:
 - тип: *Структура*:

- ключ: `Данные`, значение: `ТаблицаЗначений`:

Фамилия	Город
Петров	Тула
Иванов	Москва

- значение поля `Данные` фильтра метода `ВыгрузитьЖурналРегистрации()`:

- тип: Массив:

- тип `Структура`:

- ключ: `Данные`, значение: `Тула`;

- результат: запись соответствует отбору, т. к. искомое значение (`Тула`) найдено в одной из ячеек таблицы значений, находящейся в свойстве `Данные` поля `Данные` записи журнала регистрации.

Для поля `Метаданные` записи журнала регистрации, у которого в событии хранится массив метаданных, запись соответствует отбору, если хотя бы один элемент массива из поля `Метаданные` отбора равен хотя бы одному элементу массива записи журнала.

22.5. Дополнительные методы

Для облегчения интерактивного формирования отборов журнала регистрации полезно использовать метод `ПолучитьЗначенияОтбораЖурналаРегистрации()`. С помощью данного метода можно получать доступные значения отборов для следующих параметров отбора: `Пользователь`, `Компьютер`, `ИмяПриложения`, `Событие`, `Метаданные`, `РабочийСервер`, `ОсновнойIPПорт`, `ВспомогательныйIPПорт`. Так, при необходимости установить отбор по событию журнала регистрации можно вначале получить список событий, которые реально присутствуют в данном журнале, а затем выбрать из полученного списка.

Для формирования представлений событий журнала регистрации на языке интерфейса платформы служит метод `ПредставлениеСобытияЖурналаРегистрации()`.

В том случае, если необходимо выполнить копирование фрагмента журнала регистрации с определенным фильтром, следует использовать метод `СкопироватьЖурналРегистрации()`. При необходимости выполнить удаление некоторого набора записей, отвечающих заданному фильтру (включая полную очистку журнала регистрации), следует использовать метод `ОчиститьЖурналРегистрации()`. Следует помнить, что очистка журнала регистрации поддерживается только для журнала регистрации, который работает в формате `.lgd`.

Глава 23. Механизм криптографии

23.1. Общая информация

При использовании системы «1С:Предприятие» в системах автоматизации может быть необходимо обеспечить проверку, что тот или иной документ, хранящийся в системе, не был изменен (например, к объекту базы данных **Договор** прикладывается документ, содержащий текст самого договора). Также может возникнуть необходимость обеспечить передачу какой-либо подписанной информации или организовать утверждение какого-либо документа в рамках системы. Возможны ситуации, когда требуется обеспечить передачу информации по открытым каналам так, чтобы с ней было невозможно ознакомиться, перехватив передачу (зашифровать данные).

Для подобных ситуаций в системе «1С:Предприятие» реализован механизм криптографии, базирующийся на асимметричном шифровании (используется пара ключей – **открытый** и **закрытый**).

ВНИМАНИЕ! Механизм криптографии «1С:Предприятия» не содержит реализации собственно алгоритмов криптографии. Он обеспечивает набор объектов, позволяющих взаимодействовать с внешними модулями криптографии сторонних производителей.

Открытый ключ предназначен для передачи по открытым каналам. **Закрытый ключ** для распространения не предназначен и должен быть максимально защищен.

Чтобы зашифровать данные, необходимо знать открытые ключи получателей. Для расшифровки нужно иметь закрытый ключ, являющийся парой для открытого, указанного при осуществлении шифрования. Для генерации цифровой подписи нужен закрытый ключ, а для проверки подписи – открытый ключ подписавшего (чаще всего открытый ключ включен в саму подпись).

Для того чтобы подтвердить, что конкретный открытый ключ действительно принадлежит некоторому субъекту, служит **удостоверяющий центр**, который предоставляет такую гарантию посредством подписи.

Сертификат – это открытый ключ, подписанный удостоверяющим центром. Ввиду необходимости защиты закрытый ключ существует в контейнере. **Контейнер ключей** может включать помимо закрытого ключа открытый ключ (например, в виде сертификата).

При работе с механизмом криптографии следует учитывать, где выполняется эта работа. Причина заключается в том, что схемы работы с криптографией на стороне клиентского приложения и на стороне сервера существенно отличаются. На стороне сервера используются так называемые синхронные методы работы с криптографией, в то время как на стороне клиентского приложения используются асинхронные методы работы.

Для работы в веб-клиенте должно быть установлено расширение работы с криптографией. Особенностью веб-клиента является выполнение запроса разрешения пользователя на выполнение некоторых операций: доступ к файловой системе и к закрытому ключу.

23.2. Основные понятия

При описании механизмов будут использоваться некоторые термины.

Электронная цифровая подпись (далее – **ЭЦП**) – последовательность данных, полученная в результате криптографического преобразования исходной информации с использованием закрытого ключа ЭЦП. Позволяет подтверждать целостность и неизменность этой информации, а также ее авторство при условии использования открытого ключа ЭЦП и его сертификата.

Центр по удостоверению подлинности электронной цифровой подписи (далее – **Удостоверяющий центр**) – юридическое лицо или выделенное подразделение юридического лица, обладающее полномочиями на удостоверение принадлежности конкретного открытого ключа ЭЦП определенному пользователю.

Владелец сертификата ключа – физическое лицо, на имя которого удостоверяющим центром изготовлен сертификат открытого ключа и которое владеет соответствующим закрытым (секретным) ключом.

Сертификат – электронный документ, включающий открытый ключ и информацию о владельце данного ключа, заверенную с помощью ЭЦП удостоверяющим центром.

Модуль криптографии – библиотека функций, в которой реализованы непосредственно криптографические алгоритмы или через которую осуществляется доступ к механизмам шифрования.

23.3. Общая схема использования механизмов криптографии

При работе с криптографическими механизмами можно выделить два основных варианта использования:

- шифрование/расшифровка данных;
- подпись/проверка подписи данных.

Общая схема работы при шифровании/расшифровке выглядит следующим образом:

1. Создается необходимый объект доступа к криптографической функциональности (объект **МенеджерКриптографии**).
2. Выбирается необходимый сертификат, который является **открытым** ключом принимающей стороны (объект **СертификатКриптографии**). С помощью этого сертификата будет выполняться **шифрование** данных.
3. Выполняется шифрование необходимого файла или двоичных данных с помощью методов **Зашифровать()** созданного объекта типа **МенеджерКриптографии**.
4. Зашифрованные данные готовы для передачи по открытым каналам.
5. При получении зашифрованных данных выполняется обратная операция.
6. Создается необходимый объект доступа к криптографической функциональности. Модуль криптографии, связанный с объектом, должен поддерживать работу с алгоритмами, которые использованы для шифрования данных.

7. Выполняется расшифровка полученных данных с помощью метода `Расшифровать()` созданного объекта доступа к криптографической функциональности.

Общая схема работы при подписи/проверке подписи выглядит следующим образом:

1. Создается необходимый объект доступа к криптографической функциональности (объект `МенеджерКриптографии`).
2. Выбирается необходимый сертификат, связанный с **закрытым** ключом подписывающей стороны (объект `СертификатКриптографии`). С помощью этого сертификата будет выполняться **формирование** ЭЦП.
3. Выполняется формирование ЭЦП необходимого файла или двоичных данных с помощью методов `Подписать()` созданного объекта типа `МенеджерКриптографии`.

При формировании подписи следует учитывать следующие особенности: вначале формируется хеш-сумма подписываемых данных и затем выполняется подпись полученной хеш-суммы. Это сделано для того, чтобы уменьшить время, необходимое для выполнения операции подписи данных.

В силу этого рекомендуется явным образом указывать алгоритмы, которые будут использоваться как для вычисления хеш-суммы (свойство `АлгоритмХеширования`), так и для выполнения подписи (свойство `АлгоритмПодписи`). Если принимающая сторона не будет поддерживать используемые алгоритмы (хеширования и подписи), то проверить подпись будет невозможно.

Для того чтобы явным образом указать используемые алгоритмы, следует воспользоваться методом `ПолучитьИнформацияМодуляКриптографии()` менеджера криптографии. В результате использования метода будет получен объект `ИнформацияМодуляКриптографии`, в котором свойства `АлгоритмыПодписи` и `АлгоритмыХеширования` будут содержать список алгоритмов, которые поддерживает используемый модуль криптографии. В свойства менеджера криптографии следует указывать те данные, которые содержатся в свойствах объекта `ИнформацияМодуляКриптографии`.

Информация об используемых алгоритмах будет содержаться в подписи, поэтому нет необходимости дополнительно передавать ее вместе с подписанными данными.

4. Подписываемые данные с электронной цифровой подписью передаются принимаемой стороне.
5. После получения подписанных данных и ЭЦП выполняется обратная операция.
6. Создается необходимый объект доступа к криптографической функциональности. Модуль криптографии, связанный с объектом, должен поддерживает работу с алгоритмами, которые использованы для формирования ЭЦП.
7. Выполняется проверка электронной подписи с помощью метода `ПроверитьПодпись()` созданного объекта доступа к криптографическому функционалу. Если модуль криптографии, используемый для проверки подписи, не поддерживает алгоритм, используемые для вычисления хеш-суммы и подписи данных, то проверить подпись будет невозможно и будет выдана ошибка.

Сертификаты, необходимые для выполнения криптографических операций, получают у соответствующего удостоверяющего центра.

При работе с объектом `МенеджерКриптографии` не рекомендуется получать несколько хранилищ сертификатов с одинаковыми характеристиками, поскольку модификация одного хранилища может привести к разному поведению другого, в зависимости от используемых модулей криптографии.

23.4. Работа с модулями криптографии

Для взаимодействия с модулями криптографии в ОС Windows используется Microsoft CryptoAPI.

Для взаимодействия с модулями криптографии в ОС Linux используется непосредственное взаимодействие с установленными компонентами.

Список поддерживаемых компонент:

- Средство криптографической защиты информации КристоПро. Для использования данного средства следует в качестве параметра `ТипМодуляКриптографии` конструктора объекта `МенеджерКриптографии` указать значение 75.

23.5. Примеры использования

В данном разделе приводятся примеры реализации некоторых типовых задач при использовании механизма криптографии.

Создание объекта доступа к криптографическому функционалу

Создание объекта доступа к криптографическому функционалу является базовой операцией, без выполнения которой невозможны дальнейшие операции с механизмом криптографии.

```
МенеджерКриптографии = Новый МенеджерКриптографии("", "", 75);
```

[Копировать в буфер обмена](#)

В данном примере создается модуль для работы с российскими модулями криптографии (значение параметра `ТипМодуляКриптографии` равен 75).

Получение списка сертификатов

Формируется список сертификатов из выбранных хранилищ сертификатов для дальнейшей работы с этим списком.

[Копировать в буфер обмена](#)

```
&НаКлиенте
Функция ПолучитьСписокСертификатов(ТипМенеджераКриптографии, МассивТипов, ПроверятьДатуОкончания)
// Список сертификатов
СписокСертификатов = Новый Массив;
МенеджерКриптографии = Новый МенеджерКриптографии("", "", ТипМенеджераКриптографии);
Для Каждого ТипХранилища Из МассивТипов Цикл
// Получаем сертификаты для каждого типа хранилища сертификатов
```

```

Хранилище = МенеджерКриптографии.ПолучитьХранилищеСертификатов(ТипХранилища);
// Выбираем все сертификаты
СертификатыХранилища = Хранилище.ПолучитьВсе();
ТекущаяДата = ТекущаяДата();
Для Каждого Сертификат Из СертификатыХранилища Цикл
    Если ПроверятьДатуОкончания И Сертификат.ДатаОкончания < ТекущаяДата Тогда
        // Пропускаем истекшие сертификаты, если нужно
        Продолжить;
    КонецЕсли;
    СписокСертификатов.Добавить(Сертификат);
КонецЦикла;
КонецЦикла;
Возврат СписокСертификатов;
КонецФункции
&НаКлиенте
Процедура ПолучениеСпискаСертификатов()
    ТипыХранилищ = Новый Массив;
    ТипыХранилищ.Добавить(ТипХранилищаСертификатовКриптографии.ПерсональныеСертификаты);
    ТипыХранилищ.Добавить(ТипХранилищаСертификатовКриптографии.СертификатыПолучателей);
    Список = ПолучитьСписокСертификатов(75, ТипыХранилищ, Истина);
    // ...
КонецПроцедуры

```

Шифрование файла

Выполняется шифрование интерактивно выбранного файла и интерактивная запись зашифрованного файла на диск клиентского компьютера.

В целях демонстрации для операции шифрования всегда используется первый по списку сертификат из всех сертификатов, установленных на компьютере.

Копировать в буфер обмена

```

&НаСервере
Функция ЗашифроватьНаСервере(АдресДанных, ДанныеСертификатов)
    // Создадим сертификаты на основании двоичных данных сертификатов с клиента
    Сертификаты = Новый Массив();
    Для Каждого ДанныеСертификата Из ДанныеСертификатов Цикл
        Сертификаты.Добавить(Новый СертификатКриптографии(ДанныеСертификата));
    КонецЦикла;
    МенеджерКриптографии = Новый МенеджерКриптографии("", "", 75);
    // Получим файл для шифрования из временного хранилища
    Данные = ПолучитьИзВременногоХранилища(АдресДанных);
    Если ТипЗнч(Данные) <> Тип("ДвоичныеДанные") Тогда
        Возврат Ложь;
    КонецЕсли;
    // Шифруем двоичные данные
    ЗашифованныеДвоичныеДанные = МенеджерКриптографии.Зашифровать(Данные, Сертификаты);
    // Сохраняем во временное хранилище
    АдресДанных = ПоместитьВоВременноеХранилище(ЗашифованныеДвоичныеДанные);
    Возврат Истина;
КонецФункции
&НаКлиенте
Процедура ШифрованиеФайла()
    Адрес = "";
    Результат = ПоместитьФайл(Адрес, , , Истина);
    Если Не Результат Тогда
        Возврат;
    КонецЕсли;
    ТипыСертификатов = Новый Массив;
    ТипыСертификатов.Добавить(ТипХранилищаСертификатовКриптографии.ПерсональныеСертификаты);
    Список = ПолучитьСписокСертификатов(75, ТипыСертификатов, Истина);
    // В примере всегда шифруем с помощью первого по порядку сертификата
    Сертификаты = Новый Массив;
    Сертификаты.Добавить(Список[0].Выгрузить());
    // Шифруем файл
    Результат = ЗашифроватьНаСервере(Адрес, Сертификаты);
    Если Не Результат Тогда
        Возврат;
    КонецЕсли;
    // Интерактивно сохраняем зашифрованный файл на диск
    ПолучитьФайл(Адрес, , Истина);
КонецПроцедуры

```

Расшифровка файла

Выполняется попытка расшифровки выбранного файла. Необходима реализация функции `ПолучитьПарольДоступа()`, которая возвращает значение пароля доступа к закрытому ключу.

ПРИМЕЧАНИЕ. При использовании метода `Расшифровать()` исключение будет вызвано только после того, как неудачно завершились попытки расшифровки с помощью всех доступных сертификатов, а не после первой ошибки.

Копировать в буфер обмена

```

&НаКлиенте
Процедура РасшифровкаФайла()
    // Выбрать зашифрованный файл
    Адрес = "";
    Результат = ПоместитьФайл(Адрес, , , Истина);
    Если Не Результат Тогда
        Возврат;
    КонецЕсли;
    Данные = ПолучитьИзВременногоХранилища(Адрес);
    // Расшифровать файл
    МенеджерКриптографии = Новый МенеджерКриптографии("", "", 75);
    МенеджерКриптографии.ПарольДоступаКЗакрытомуКлючу = ПолучитьПарольДоступа();
    РасшифованныеДанные = МенеджерКриптографии.Расшифровать(Данные);

```

```

Адрес = ПоместитьВоВременноеХранилище(РасшифрованныеДанные);
ПолучитьФайл(Адрес, , Истина);
КонiecПроцедуры

```

Формирование ЭЦП

Выполняется формирование файла с электронной цифровой подписью. В данном примере ЭЦП всегда сохраняется в файл `signature.sign`. Необходима реализация функции `ПолучитьПарольДоступа()`, которая возвращает значение пароля доступа к закрытому ключу.

Копировать в буфер обмена

```

&НаКлиенте
Процедура ПодписатьФайл()
    Адрес = "";
    Результат = ПоместитьФайл(Адрес, , , Истина);
    Если Не Результат Тогда
        Возврат;
    КонiecЕсли;
    ТипыСертификатов = Новый Массив;
    ТипыСертификатов.Добавить(ТипХранилищаСертификатовКриптографии.ПерсональныеСертификаты);
    Список = ПолучитьСписокСертификатов(75, ТипыСертификатов, Истина);
    Сертификат = Список[0];
    Данные = ПолучитьИзВременногоХранилища(Адрес);
    МенеджерКриптографии = Новый МенеджерКриптографии("", "", 75);
    МенеджерКриптографии.ПарольДоступаКЗакрытомуКлючу = ПолучитьПарольДоступа();
    МенеджерКриптографии.Подписать(Данные, "signature.sign", Сертификат);
КонiecПроцедуры

```

Проверка ЭЦП

Функция выполняет проверку действительности ЭЦП для пары **оригинальный файл – файл ЭЦП**.

Копировать в буфер обмена

```

&НаКлиенте
Функция ПроверитьПодписьФайла(ИмяПодписанногоФайла, ИмяФайлаПодписи)
    Сертификат = Неопределено;
    МенеджерКриптографии = Новый МенеджерКриптографии("", "", 75);
    Попытка
        МенеджерКриптографии.ПроверитьПодпись(ИмяПодписанногоФайла, ИмяФайлаПодписи, Сертификат);
    Исключение
        Возврат Ложь;
    КонiecПопытки
        Возврат Истина;
КонiecФункции

```

23.6. Асинхронная работа с механизмом криптографии

Использование асинхронной схемы работы с криптографией имеет ряд особенностей:

1. Она доступна только на стороне клиентского приложения. На стороне сервера следует использовать синхронный способ работы.
2. Инициализация объектов доступа к криптографической функциональности отличается от своих синхронных аналогов.

Рассмотрим пример работы с криптографической функциональностью в асинхронном режиме. В качестве примера будет рассмотрен процесс получения списка сертификатов из определенного типа хранилища сертификатов.

ПРИМЕЧАНИЕ. Пример, приведенный ниже, не является законченным. Он предназначен для демонстрации асинхронной работы со средствами криптографии.

Копировать в буфер обмена

```

&НаКлиенте
Процедура ПолучитьСписокСертификатов()
    ДополнительныеПараметры = Новый Структура;
    ДополнительныеПараметры.Вставить("ТипХранилищаСертификатов", ТипХранилищаСертификатовКриптографии.ПерсональныеСертификаты);
    ДополнительныеПараметры.Вставить("ПроверятьДатуОкончания", Истина);
    ОписаниеОповещения = Новый ОписаниеОповещения("ПолучитьСписокСертификатовЗавершение", ЭтотОбъект, ДополнительныеПараметры);
    ДополнительныеПараметры.Вставить("ОповещениеЗавершения", ОписаниеОповещения);
    ОбратныйВызов = Новый ОписаниеОповещения("ИнициализацияЗавершение", ЭтотОбъект, ДополнительныеПараметры);
    Криптография = Новый МенеджерКриптографии;
    Криптография.НачатьИнициализацию(ОбратныйВызов, "", "", 75);
КонiecПроцедуры
&НаКлиенте
Процедура ПолучитьСписокСертификатовЗавершение(СписокСертификатов, ДополнительныеПараметры) Экспорт
    // здесь используем полученные сертификаты
КонiecПроцедуры
&НаКлиенте
Процедура ИнициализацияЗавершение(Менеджер, ДополнительныеПараметры) Экспорт
    Оповещение = Новый ОписаниеОповещения("ПолучениеХранилищаСертификатовЗавершение", ЭтотОбъект, ДополнительныеПараметры);
    Менеджер.НачатьПолучениеХранилищаСертификатов(Оповещение, ДополнительныеПараметры.ТипХранилищаСертификатов);
КонiecПроцедуры
&НаКлиенте
Процедура ПолучениеХранилищаСертификатовЗавершение(Хранилище, ДополнительныеПараметры) Экспорт
    Оповещение = Новый ОписаниеОповещения("МассивСертификатовЗавершение", ЭтотОбъект, ДополнительныеПараметры);
    Хранилище.НачатьПолучениеВсех(Оповещение);
КонiecПроцедуры
&НаКлиенте
Процедура МассивСертификатовЗавершение(Сертификаты, ДополнительныеПараметры) Экспорт
    Результат = Новый Массив;
    ТекущаяДата = ТекущаяДата();
    Для Каждого Сертификат Из Сертификаты Цикл
        Если ДополнительныеПараметры.ПроверятьДатуОкончания И Сертификат.ДатаОкончания < ТекущаяДата Тогда
            Продолжить;
        КонiecЕсли;

```

[Результат.Добавить \(Сертификат\) ;](#)
[КонецЦикла ;](#)
[ВыполнитьОбработкуОповещения \(ДополнительныеПараметры.ОповещениеЗавершения, Результат\) ;](#)
[КонецПроцедуры](#)

Данный пример полностью иллюстрирует особенности работы с криптографией в асинхронном режиме. Получение списка сертификатов начинается с вызова метода `ПолучитьСписокСертификатов()` и заканчивается вызовом метода `ПолучитьСписокСертификатовЗавершение()`. Собственно получение состоит из инициализации модуля работы с криптографией и использования асинхронных аналогов следующих синхронных методов: `ПолучитьХранилищеСертификатов()` и `ПолучитьВсе()`. Также отдельным образом выполняется проверка на истекшие сертификаты. Оповещение `ИнициализацияЗавершение()` будет вызвано после того, как завершится инициализация менеджера криптографии. Затем работа строится аналогичным образом: в обработчике завершения предыдущего метода вызывается следующее действие, в обработчике завершения которого происходит очередное действие и т. д.

Получается четкая последовательность действий:

[Копировать в буфер обмена](#)

```

ПолучитьСписокСертификатов() ->
  ИнициализацияЗавершение() ->
    ПолучениеХранилищаСертификатовЗавершение() ->
      МассивСертификатовЗавершение() ->
        ПолучитьСписокСертификатовЗавершение()

```

Особого внимания требует только последний переход. В обработчике оповещения `МассивСертификатовЗавершение()` не вызывается никакого асинхронного метода, который требует указания обработчика оповещения. Нужно описание оповещения поступает в самый последний (по цепочке вызовов) обработчик оповещения с помощью задания дополнительных параметров. Нужные действия выполняются в методе `ПолучитьСписокСертификатов()`, а в самом последнем обработчике (`ПолучитьСписокСертификатовЗавершение()`) происходит явный вызов обработчика, «указатель» на который последовательно передавался по цепочки обработчиков с помощью дополнительных параметров. Фактический возврат происходит в результате выполнения метода `ВыполнитьОбработкуОповещения()`.

Пример разработан для использования в модуле формы, но переработка примера для работы из общих модулей не представляет существенной проблемы. В этом случае первые две процедуры примера будут в одном общем модуле, а остальные – в другом, с соответствующей корректировкой описания обработчиков оповещения и вызовов методов.

Аналогичным образом будут выглядеть и другие типичные приемы работы с криптографией.

Для работы с механизмом криптографии в веб-клиенте следует установить расширение работы с криптографией (см. [здесь](#)), при этом следует помнить, что во всех веб-браузерах, кроме Google Chrome, можно использовать как синхронные, так и асинхронные методы работы с криптографией. При этом используемый набор методов определяется свойством конфигурации [Режим использования синхронных вызовов расширений и внешних компонент](#) (см. [здесь](#)). При использовании веб-браузера Google Chrome доступны только асинхронные методы работы с криптографией. Соответствие методов см. [здесь](#).

Глава 24. Внешние источники данных

При эксплуатации информационной системы, построенной на базе «1С:Предприятия», могут возникать задачи, связанные с получением информации из внешних баз данных (как реляционных, так и аналитических) и использованием этой информации в «1С:Предприятии» различным образом, например, в виде отчетов или для каких-либо расчетов. Для решения такого рода задач в «1С:Предприятии» существует объект конфигурации [ВнешниеИсточникиДанных](#). В состав внешнего источника данных могут входить таблицы реляционного источника данных и кубы аналитического источника данных. В прикладном решении может существовать произвольное количество подключенных внешних источников данных. Далее будет более подробно рассмотрена работа с каждым видом внешнего источника данных.

24.1. Работа с реляционными внешними источниками данных

24.1.1. Общая информация

Объект, описывающий внешний источник данных, подключенный к реляционной базе данных, состоит из таблиц, а каждая таблица состоит из полей. Данные в таблицах могут ссылаться на другие таблицы – например, значение в поле таблицы может являться идентификатором записи в другой таблице. Объект, описывающий внешний источник данных, может быть использован следующим образом:

- в качестве источника данных для запросов;
- в качестве источника данных в системе компоновки данных;
- в качестве источника для динамических списков;
- входить в состав общих реквизитов (см. [здесь](#));
- записи таблиц могут отображаться в управляемых формах «1С:Предприятия» (не поддерживается использование обычных форм для объектов внешних источников данных);
- таблицы внешнего источника данных могут выступать в качестве типов реквизитов информационной базы;
- к таблицам (и полям) внешних источников данных можно применять права доступа и накладывать ограничения доступа к данным;
- доступ к таблицам и полям возможен из встроеного языка;
- таблица внешнего источника данных может входить в состав подсистем (см. [здесь](#));
- таблица внешнего источника данных может входить в состав функциональных опций (см. [здесь](#));
- для таблицы внешнего источника данных можно создавать характеристики (см. [здесь](#)).

Для получения доступа к внешним источникам данных используется механизм ODBC. Данные внешних источников данных доступны как для **чтения**, так и для **записи**.

ВНИМАНИЕ! Механизм внешних источников данных не должен использоваться для доступа к базам данных «1С:Предприятия», так как модель данных «1С:Предприятия» не рассчитана на работу с данными на уровне физических структур хранения в системах управления базами данных (СУБД). Возможность записи во внешние источники данных не должна использоваться для замены штатного механизма хранения данных прикладного решения.

Возможности системы «1С:Предприятие» максимально учитываются при использовании в качестве источника внешних данных следующих СУБД (свойство [Тип СУБД](#) параметров соединения с источником данных):

- Microsoft SQL Server,
- IBM DB2,
- PostgreSQL,
- Oracle Database,
- MySQL.

ПРИМЕЧАНИЕ. При работе с внешним источником данных могут возникать ошибки в том случае, если свойство [Тип СУБД](#) параметров соединения с внешним источником данных содержит значение, которое не соответствует реально используемой системе.

При использовании других СУБД возможности работы с внешними источниками данных зависят от самой СУБД. Для таких источников данных, следующие функции и выражения языка запросов: [ПОДСТРОКА](#), [ГОД](#), [КВАРТАЛ](#), [МЕСЯЦ](#), [ДЕНЬГОДА](#), [ДЕНЬ](#), [НЕДЕЛЯ](#), [ДЕНЬНЕДЕЛИ](#), [ЧАС](#), [МИНУТА](#), [СЕКUNДА](#), [РАЗНОСТЬДАТ](#), [ЕСТЬNULL](#), [ВЫРАЗИТЬ](#), [СПЕЦСИМВОЛ](#), [СОЕДИНЕНИЕ](#), [ЛЕВОЕ СОЕДИНЕНИЕ](#), [ПРАВОЕ СОЕДИНЕНИЕ](#), [ПОЛНОЕ СОЕДИНЕНИЕ](#), преобразуются в запрос к СУБД с использованием управляющих последовательностей ODBC (ODBC Escape Sequences, [http://msdn.microsoft.com/en-us/library/windows/desktop/ms715364\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms715364(v=vs.85).aspx), на английском языке).

В случае использования внешнего источника данных в качестве источника для динамического списка, следует помнить, что упорядочивание динамического списка выполняется с использованием порядка сортировки, который установлен для базы данных, выступающей в качестве внешнего источника данных. Порядок сортировки может не совпадать с тем порядком сортировки, который используется в том случае, если база данных используется собственно «1С:Предприятием». В связи с этим возможны ситуации, когда результат упорядочивания в информационной базе и во внешнем источнике данных будет отличаться (при одинаковых упорядочиваемых данных). Такое поведение является нормальным.

Для осуществления подключения к внешнему источнику данных следует сформировать строку подключения, которая может содержать в себе либо все параметры подключения, необходимые для выбранного драйвера ODBC, либо указание на сформированное описание источника данных DSN (Data Source Name). Подробнее о строке подключения см. [здесь](#).

Ниже приводятся примеры строк подключения.

Строка соединения с указанием логина и пароля доступа:

DRIVER={SQL Server};SERVER=(local);UID=user;PWD=password;DATABASE=AdventureWorksLT2008

Логин и пароль доступа требуется указывать дополнительно:

[Копировать в буфер обмена](#)

DRIVER={SQL Server};SERVER=CASH-SERVER;DATABASE=CashReceipts

Указание предварительно настроенного источника данных (DSN):

[Копировать в буфер обмена](#)

DSN=MyDB

Также следует помнить, что соединение с внешним источником данных следует указывать не только в конфигураторе (если используется механизм импорта структуры таблиц из внешнего источника данных), но и в режиме **1С:Предприятие** для получения собственно данных.

24.1.2. Общая схема использования

Для того чтобы использовать информацию из внешних источников данных в системе на базе «1С:Предприятия», следует придерживаться следующей общей схемы:

- Изучить структуру внешнего источника данных и понять, какая информация (таблицы и поля) необходима для работы алгоритмов или отчетов в «1С:Предприятии».
- Создать объект **Внешние источники данных** в конфигураторе и для него создать подчиненные таблицы и поля. Это можно сделать с помощью специального помощника.
- Реализовать использование созданных объектов в прикладном решении.
- Выполнить настройку параметров подключения к внешнему источнику данных в той сети, где будет использоваться прикладное решение. Эти параметры могут отличаться от тех, которые использовались при загрузке структуры внешнего источника данных.

ПРИМЕЧАНИЕ. Параметры доступа к внешнему источнику данных, которые были заданы в конфигураторе, не будут использованы системой в режиме 1С:Предприятие.

24.1.3. Редактирование структуры внешнего источника данных

Для работы с таблицами внешних источников данных предназначена ветвь с именем соответствующего внешнего источника данных ветви **Внешние источники данных – Таблицы** дерева конфигурации. Определение внешнего источника данных состоит из трех шагов:

- Определение собственно внешнего источника данных;
- Определение таблиц источника данных;
- Определение полей для каждой таблицы внешнего источника данных.

При этом в конфигурации может быть описано меньшее количество таблиц и полей в них, чем есть в реальной базе данных, но нельзя задать таблицу или поле, которое отсутствует в реальной базе данных.

Создание структуры внешнего источника данных возможно как в ручном режиме, так и при загрузке этой структуры с помощью специального конструктора.

24.1.3.1. Внешний источник данных

При создании внешнего источника данных следует указать его имя. Этот объект предназначен для идентификации группы таблиц при обращении к данным, которые в нем содержатся. Источник данных состоит из одной или нескольких таблиц, которые, в свою очередь, состоят из полей.

Если во внешний источник данных предполагается запись, то следует установить нужное значение в свойстве **Режим управления блокировкой данных**. Если для внешнего источника данных режим управления блокировок установлен в значение **Автоматический и управляемый**, то для каждой таблицы режим блокировок определяется значением соответствующего свойства таблицы. Если для внешнего источника данных выбран определенный конкретный режим управления блокировками, то одноименное свойство таблицы внешнего источника данных игнорируется.

24.1.3.2. Таблица внешнего источника данных

При создании новой таблицы открывается окно редактирования объекта (см. [здесь](#)).

Свойство таблицы **Имя** служит для идентификации таблицы внутри прикладного решения. Для того чтобы система знала, какой физической таблице внешнего источника данных соответствует объект конфигурации, следует заполнить свойство **Имя в источнике данных**. Значение в этом свойстве в точности должно соответствовать имени таблицы внешнего источника данных. Не поддерживается размещение в одной таблице прикладного решения данных из нескольких физических таблиц внешнего источника данных.

Свойство **Тип данных таблицы** определяет, какие сущности в этой таблице хранятся: объектные или неobjектные. Если в таблице можно выделить одно поле, которое однозначно определяет запись в таблице, значит, таблица может хранить объектные данные. Ближайшим аналогом таких таблиц может служить справочник. Если запись в таблице идентифицируется несколькими ключевыми полями, то такая таблица содержит неobjектные данные. Ближайшим аналогом таких таблиц служит регистр сведений. Ключевые поля таблицы необходимо указать в свойстве **Поле ключа**.

ПРИМЕЧАНИЕ. Не рекомендуется использовать в качестве ключевых полей таблицы внешнего источника данных поля, которые содержат значения **NULL**.

Для таблиц, содержащих объектные данные, можно указать поле, которое будет выступать в качестве представления объекта. Это можно сделать с помощью свойства **Поле представления**.

Если таблица хранит иерархические данные, то системе можно сообщить об этом с помощью свойств **Поле родителя**, **Признак незаполненного родителя** и **Значение незаполненного родителя**. Эти свойства доступны только для таблиц, в которых хранятся объектные данные. Свойство **Поле родителя** служит для указания поля таблицы базы данных, которое хранит ссылку на родителя для данной записи. Поля, которые могут быть выбраны в данном свойстве, должны иметь тип ссылки на используемую таблицу. Для таблицы **products** из примера (см. [здесь](#)) поле родителя

должно иметь тип `ВнешнийИсточникДанныхТаблицаСсылка.DBF.products`. Свойства `Признак незаполненного родителя` и `Значение незаполненного родителя` определяют, какое значение будет определять запись, у которой отсутствует родительская запись. После указания данных свойств:

- система компоновки данных использует их при работе с иерархическими группировками и при проверке условий иерархии;
- язык запросов использует их для работы конструкции `В ИЕРАРХИИ`;
- эти свойства используются при отображении таблиц, хранящих иерархические данные (возможности отображения аналогичны возможностям, которые используются при отображении иерархических справочников).

Свойство `Ввод по строке` (см. [здесь](#)) позволяет указать имена полей, по которым будет выполняться поиск в поле, тип данных которого указывает на таблицу внешнего источника данных.

При ручном создании таблиц свойства `Поле ключа`, `Поле представления` и `Ввод по строке` можно указать только после создания полей таблицы внешнего источника данных.

За возможность записи в таблицу внешнего источника данных отвечает флажок `Только чтение`. Если флажок сброшен – в указанную таблицу допускается запись информации. Флажок устанавливается в значение `Истина` автоматически для таблиц, основанных на представлениях (view) и функциях. Для таблицы, в которую допустима запись информации, следует обратить внимание на следующие свойства:

- `Уровень изоляции транзакций` – определяет уровень изоляции транзакций, которые будут устанавливаться при неявных транзакциях записи в эту таблицу (см. [здесь](#)).
- `Поле версии данных` – в данном свойстве указывается поле внешнего источника данных, которое увеличивает свое значение автоматически, при каждой записи в данную таблицу. Если поле указано, то система при интерактивном изменении объекта опирается на значение этого поля, При записи выполняется сравнение значения этого поля в базе данных и в форме: если значения различаются, значит, объект изменен.

Если поле не указано, то при интерактивном открытии формы происходит считывание объекта в память, а при записи происходит повторное чтение объекта и сравнение с копией в памяти. Если две копии различаются, значит, объект изменен.

- `Поля блокировки данных` – указывает поля, по которым можно выполнять блокировку данных. Подробнее см. [здесь](#).
- `Вводится на основании` – указываются объекты конфигурации, на основании которых могут создаваться записи в таблице.
- `Режим управления блокировкой данных` – указывает, какой режим управления блокировками будет применяться при записи в данную таблицу (см. [здесь](#)).

24.1.3.3. Поле таблицы внешнего источника данных

Поля таблицы описывают, какие данные физической таблицы будут доступны из прикладного решения. Свойство поля `Имя` служит для идентификации поля таблицы в прикладном решении. Для указания соответствия между полем прикладного решения и полем физической таблицы служит свойство `Имя в источнике данных`. Значение в этом свойстве должно в точности соответствовать имени колонки таблицы, указанной в свойстве `Имя в источнике данных` родительского объекта данного поля. Не поддерживается объединение в одном поле таблицы данных из нескольких колонок одной или нескольких таблиц внешнего источника данных.

Если значение, находящееся в свойстве `Имя в источнике данных` заключено в одинарные кавычки, то в SQL-запрос к базе данных это значение попадает без преобразований, вне зависимости от состава символов. Если значение в поле не заключено в одинарные кавычки, то в SQL-запрос к базе данных это значение попадет заключенное в двойные кавычки, если в имени содержатся спецсимволы.

Поле `Тип` позволяет указать тип данного поля. Для выбора доступен ограниченный набор типов. Типами поля могут быть:

- `Число`;
- `Строка`;
- `Дата`;
- `Булево`;
- `УникальныйИдентификатор`;
- `ДвоичныеДанные`;
- типы, которые определяются таблицами внешних источников данных.

Если необходимо указать для поля составной тип, то в таком составном типе могут участвовать только типы `Число`, `Строка`, `Дата`, `Булево`.

При получении данных из внешних источников система автоматически преобразует эти данные к тому типу, который указан у соответствующего поля в прикладном решении. Описание правил преобразования см. [здесь](#).

Свойство `Только чтение` указывает, что данное поле в таблице не может быть записано. Это свойство следует устанавливать в значение `Истина` для полей с автоматическим изменением, автоматически формируемые ключевые поля, вычисляемых полей и т. д.

Однако при работе с внешним источником данных возникает необходимость выполнять запись полей, которые в обычной работе предназначены только для чтения, или наоборот. При этом такая необходимость является временной, например, это необходимо для выполнения какой-либо однократной или регламентной операции. Для изменения списка записываемых полей следует использовать методы `ПолучитьИзменяемыеПоля()` и `УстановитьИзменяемыеПоля()`. Таким образом, состояние свойства `Только чтение` описывает поведение поля по умолчанию, а с помощью указанных методов можно в редких случаях изменять поведение поля.

Свойство `Разрешить Null` указывает, можно или нет в поле таблицы записать значение `NULL`. Если в поле таблицы записано значение `NULL`, то в форме это значение будет отображаться как `Не заполнено`.

24.1.3.4. Прочие свойства

Форма, используемая для отображения записи и списка таблицы внешнего источника данных, может быть создана автоматически системой или прикладным разработчиком. При этом следует помнить, что если тип таблицы – не объектная и не заданы ключевые поля, то:

- отображение формы записи невозможно;
- невозможен доступ к объекту `ВнешнийИсточникДанныхТаблицаНаборЗаписей`.

Это связано с тем, что невозможно однозначно идентифицировать необходимый набор записей в таблице.

24.1.3.5. Работа с транзакциями

При записи во внешние источники данных могут использоваться как автоматические, так и управляемые блокировки (см. [здесь](#)).

Для работы с транзакциями предназначены методы `НачатьТранзакцию()`, `ЗафиксироватьТранзакцию()` и `ОтменитьТранзакцию()` менеджера внешнего источника данных. В частности, это означает, что недоступны транзакции для нескольких источников данных (распределенные транзакции).

Управление блокировками осуществляется с помощью следующих свойств и объектов:

- Свойство `Режим управления блокировкой данных` внешнего источника данных;
- Свойство `Режим управления блокировкой данных` таблицы внешнего источника данных;
- Полями, указанными в свойстве `Поля блокировки данных` таблицы внешнего источника данных;
- Свойством `Уровень изоляции транзакций` таблицы внешнего источника данных;
- Параметрами метода `НачатьТранзакцию()` менеджера внешних источников данных.

Свойство `Режим управления блокировкой данных` внешнего источника данных предназначено для установки по умолчанию режима блокировок таблиц внешнего источника данных (аналогично свойству конфигурации `Режим управления блокировкой данных`, см. [здесь](#)). Данное свойство может принимать следующие значения:

- **Автоматический** – режим автоматических блокировок для всех таблиц внешнего источника данных. В качестве уровня изоляции транзакций используется `Упорядочиваемость`. Такой режим позволяет использовать транзакции к базе данных внешнего источника одновременно из платформы и из других клиентов. В данном режиме игнорируется одноименное свойство у таблицы внешнего источника данных.
- **Управляемый** – режим управляемых блокировок для всех таблиц внешнего источника данных. В качестве уровня изоляции транзакций используется `Чтение зафиксированных`. Такой режим можно использовать в том случае, когда к базе данных внешнего источника имеет только система на базе «1С:Предприятие». В данном режиме игнорируется одноименное свойство у таблицы внешнего источника данных.
- **Автоматический и управляемый** – режим блокировок выбирается для каждой таблицы в зависимости от одноименного свойства таблицы. Уровень изоляции транзакций выбирается в зависимости от свойства `Уровень изоляции транзакций` таблицы внешнего источника данных.

Если в свойстве `Уровень изоляции транзакций` таблицы внешнего источника данных установлено значение `Авто`, то фактический режим изоляции транзакций устанавливается в зависимости от свойства `Режим управления блокировкой данных` (описано выше). Если указано конкретное свойство – будет использоваться именно оно. Если СУБД внешнего источника данных не поддерживает указанный в этом свойстве уровень изоляции транзакций, то уровень будет автоматически изменяться по следующей цепочке (до первого поддерживаемого): `Чтение незафиксированных` " `Чтение зафиксированных` " `Повторяемое чтение` " `Упорядочиваемость`.

При работе в режиме управляемых блокировок, блокировки на уровне базы данных накладываются установкой соответствующего уровня изоляции транзакций и указанием выражения `ДЛЯ ИЗМЕНЕНИЯ` в запросах к базе данных.

24.1.3.6. Особенности поведения системы при записи во внешние источники данных

При не интерактивной работе с внешним источником данных следует учитывать следующие особенности поведения:

- При создании записи в таблице внешнего источника данных:
 - Изменить реквизиты записываемого объекта, у которых свойство `ТолькоЧтение` установлено в значение `Истина`, невозможно.
 - Если у реквизита, входящего в состав полей ключа, свойство `ТолькоЧтение` установлено в значение `Истина`, то такой реквизит может быть изменен только в результате вызова метода `УстановитьСсылкуНового()`. Если у такого реквизита свойство `ТолькоЧтение` установлено в значение `Ложь`, то его значение можно изменять явным образом (с помощью операции присваивания).
 - Система не занимается автоматическим заполнением значений реквизитов, участвующих в формировании ключа объекта (или набора записей). Это означает, что запись будет завершена с ошибкой, если для записываемого объекта не установлены значения ключевых реквизитов.
- При изменении записи в таблице внешнего источника данных:
 - Изменяются только те реквизиты, для которых свойство `ТолькоЧтение` установлено в значение `Ложь`.
 - При изменении ключевых полей изменяется значение ссылки.
 - Если заполнен реквизит `ВерсияДанных`, то это значение используется в выражении `WHERE` для запроса обновления данных. После успешного выполнения запроса значение реквизита повторно считывается из базы данных.
- При удалении записи из таблицы внешнего источника данных:
 - Выполняется только непосредственное удаление. Пометка удаления недоступна и не поддерживается.
 - При удалении не выполняется контроль ссылочной целостности.

24.1.3.7. Загрузка структуры таблиц из внешнего источника данных

Существует возможность загрузки структуры таблиц из внешнего источника данных, который доступен для разработчика (например, копии реальной базы данных).

Для выполнения этой операции следует при создании таблицы указать пункт **Выбрать из списка таблиц внешнего источника данных** в конструкторе таблиц внешнего источника данных. Затем нужно указать строку соединения с внешней базой данных, воспользовавшись окном **Подключение к источнику данных**. При формировании строки подключения можно воспользоваться конструктором строки соединения с внешним источником данных. Для этого следует нажать кнопку «...» справа от поля **Строка соединения**.

После того как подключение к внешней базе успешно выполнено, будет открыт список с перечнем таблиц и полей подключенного источника данных. Затем следует выбрать те таблицы и поля, которые будут использованы конфигуратором для создания структуры объектов, описывающих текущий источник данных.

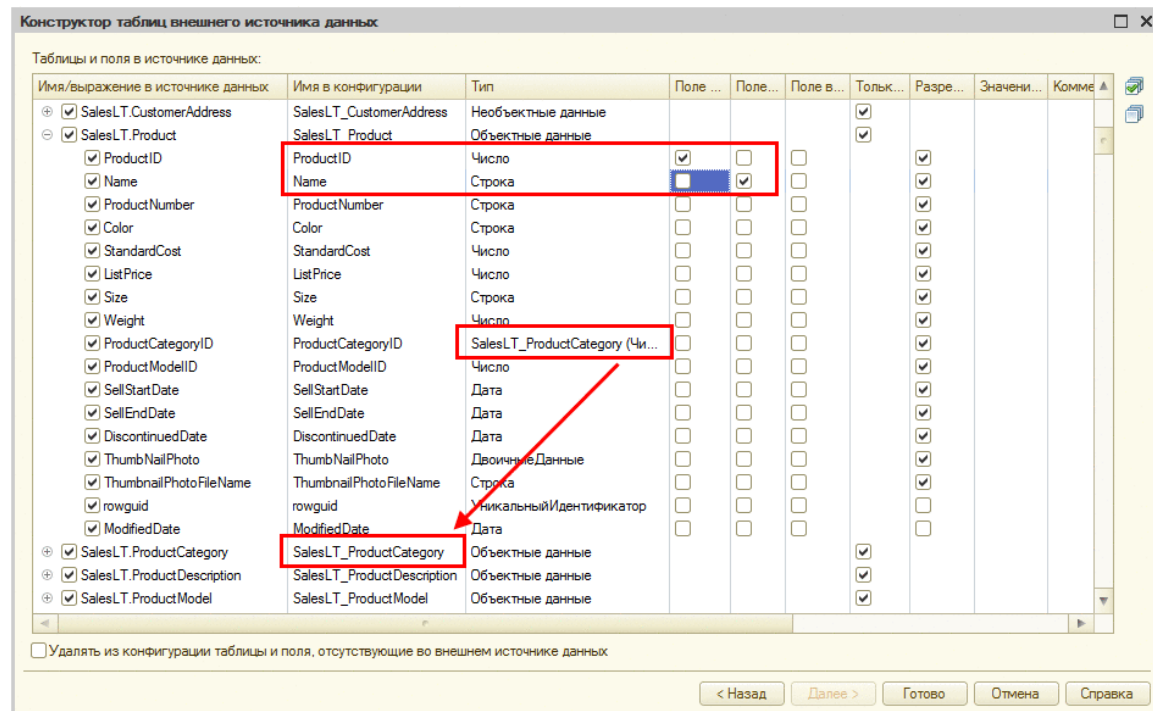


Рис. 342. Конструктор таблиц внешнего источника данных

При получении структуры внешнего источника данных система выполняет следующие действия:

- Выполняет попытку определить тип таблицы с данными: объектные или необъектные данные расположены в таблице. Таблица будет считаться объектной, если для нее указано только одно ключевое поле, и необъектной в противном случае. Если система ошиблась с указанием ключевых полей, можно вручную изменить состав полей, которые образуют ключ таблицы. Если таблица определена как объектная, у нее можно указать поле, которое формирует представление данных такого типа. Поле представления необходимо указать вручную.
- Преобразует типы колонок источника данных в типы «1С:Предприятия», которые будут использоваться для указания типов реквизитов таблиц. Все типы из внешнего источника данных преобразуются к следующим типам «1С:Предприятия»: число, строка, дата, булево, уникальный идентификатор, двоичные данные и типы, связанные с объектными таблицами источников данных.
- Выполняет попытку определить типы для полей таблицы. В этом случае система пытается определить, какого типа данные хранятся в колонке таблицы, и в том случае, если это можно считать ссылкой на данные другой таблицы, указывает в колонке соответствующий тип. Если система неправильно выбрала тип колонки таблицы, его можно изменить вручную.
- Во время настройки загрузки, имеется возможность настроить следующие параметры загружаемой структуры данных:
 - Имя объекта в конфигурации – колонка **Имя в конфигурации**;
 - Тип загружаемого реквизита – колонка **Тип**;
 - Указать поля, входящие в состав ключа (для объектных данных) – колонка **Поле ключа**;
 - Указать поле, хранящее представление объекта – колонка **Поле представления**;
 - Указать поле, хранящее версию объекта (для осуществления оптимистической блокировки данных) – колонка **Поле версии**;
 - Указать, что таблица или реквизит доступны только на чтение – колонка **Только чтение**;
 - Указать возможность указывать значение **NULL** в реквизите – колонка **Разрешить NULL**;
 - Задать значение заполнения для реквизита – колонка **Значение заполнения**.

Затем следует отметить флажками таблицы и поля, которые будут перенесены в метаданные прикладного решения.

Если флажок **Удалять из конфигурации таблицы и поля, отсутствующие во внешнем источнике данных** установлен, то при завершении работы конструктора таблиц из конфигурации будут удалены те таблицы и поля, которые отсутствуют во внешнем источнике (например, во внешнем источнике таблицы или поля были удалены) данных.

После нажатия кнопки **Готово** произойдет загрузка структуры внешнего источника данных.

24.1.4. Пример создания внешнего источника данных

Допустим, у нас существует база данных формата dbf, доступ к которой необходимо получить из прикладного решения.

База данных состоит из трех таблиц (файлов):

- Таблица `products` (хранится в файле `products.dbf`), которая содержит информацию о товарах и состоит из следующих полей:
 - `id` – идентификатор товара (ключевое поле). Тип: `Число`.
 - `code` – код товара. Тип: `Строка`.
 - `name` – наименование товара. Тип: `Строка`.
 - `article` – артикул товара. Тип: `Строка`.
- Таблица `price` (хранится в файле `price.dbf`), которая содержит текущие цены на товары и состоит из следующих полей:
 - `product` – идентификатор товара, для которого хранится цена. Тип: `Число`.
 - `price` – цена товара. Тип: `Число`.
- Таблица `sales` (хранится в файле `sales.dbf`), которая содержит данные о каждой продаже товара с указанием цены, количества и суммы. Таблица состоит из следующих полей:
 - `product` – идентификатор продаваемого товара. Тип: `Число`.
 - `price` – отпускная цена товара. Тип: `Число`.
 - `qty` – количество проданного товара. Тип: `Число`.
 - `summa` – отпускная стоимость проданного товара. Тип: `Число`.

Создадим необходимые объекты конфигурации, чтобы иметь доступ ко всем данным этих таблиц.

Вначале создадим собственно внешний источник данных. Назовем его `DBF`. Затем вручную создадим таблицы источника данных. При этом следует помнить, что реальное имя физической таблицы внешнего источника данных указывается в свойстве `Имя в источнике данных` создаваемого объекта. При создании таблиц назовем объекты конфигурации следующим образом:

- таблица `products` будет отображаться в объект `Товары`;
- таблица `price` будет отображаться в объект `Цены`;
- таблица `sales` будет отображаться в объект `Продажи`.

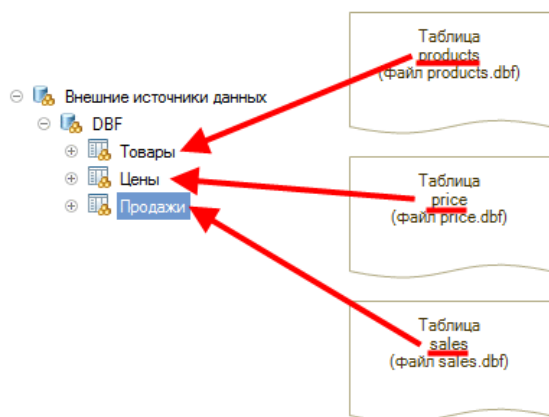


Рис. 343. Таблицы внешнего источника данных

Теперь следует создать поля для каждой таблицы, указать типы полей и задать поля ключей и представлений (если они есть).

В примере будет одна объектная таблица (таблица `Товары`). Остальные таблицы будут неobjектные. Соответствие полей физических таблиц и полей объекта конфигурации, а также типы полей можно посмотреть на [рис. 344](#).

Имя в источнике д...	Имя в конфигу...	Тип	Поле ...	По...	Комментар
price	Цены	Неobjектные данные			Цены товар
Product	Товар	ВнешнийИсточникДанныхТаблицаСсылка.DBF.Товар...	<input checked="" type="checkbox"/>		
Price	Цена	Число	<input type="checkbox"/>		
products	Товары	Objектные данные			Список тов
id	Идентификатор	Число	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Code	Код	Строка	<input type="checkbox"/>	<input type="checkbox"/>	
Name	Наименование	Строка	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Article	Артикул	Строка	<input type="checkbox"/>	<input type="checkbox"/>	
sales	Продажи	Неobjектные данные			Продажи тс
Product	Товар	ВнешнийИсточникДанныхТаблицаСсылка.DBF.Товар...	<input checked="" type="checkbox"/>		
Price	Цена	Число	<input type="checkbox"/>		
Qty	Количество	Число	<input type="checkbox"/>		
Summa	Сумма	Число	<input type="checkbox"/>		

Рис. 344. Структура таблиц в прикладном решении

Теперь следует указать, в какую подсистему входят созданные таблицы и другие параметры объектов конфигурации (при необходимости).

Если затем запустить систему в режиме 1С:Предприятие и корректно указать параметры соединения с внешним источником данных, то в панели навигации будут присутствовать созданные таблицы.

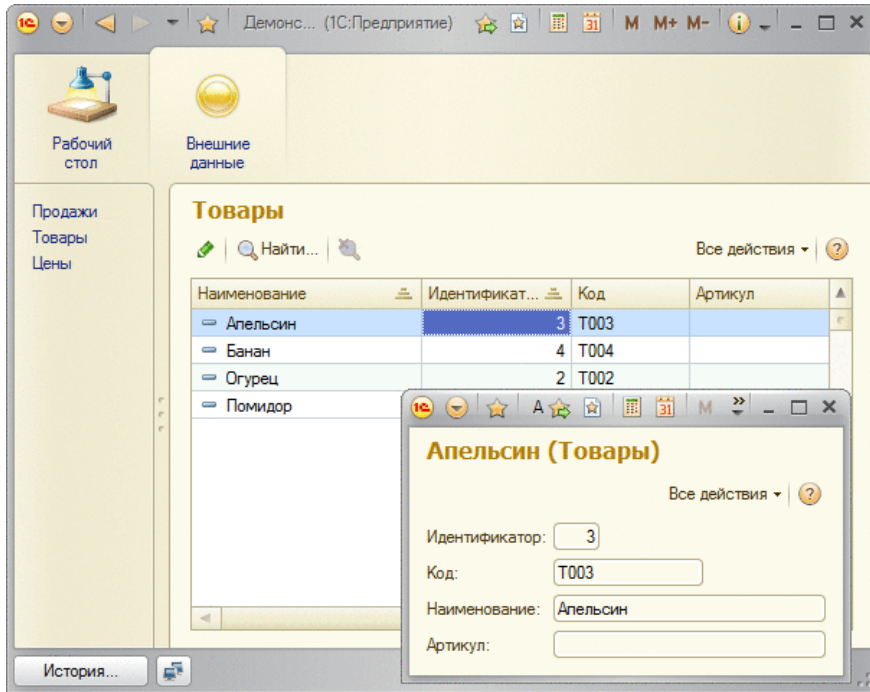


Рис. 345. Динамический список внешнего источника данных

24.1.5. Строка соединения ODBC

Для осуществления подключения к внешнему источнику данных следует сформировать строку соединения, которая может содержать в себе либо все параметры подключения, необходимые для выбранного драйвера ODBC, либо указание на сформированное описание источника данных DSN (Data Source Name).

24.1.5.1. Полная строка соединения

Описание строки соединения можно получить по следующему адресу (на английском языке): <http://msdn.microsoft.com/en-us/library/ms722656.aspx>. Из этого описания следует выделить следующее:

- Параметры строки соединения записываются парами **КлючевоеСлово=Значение** и разделяются символом ";".
- Если в строке соединения встречаются несколько параметров с одинаковым ключевым словом, то использоваться будет последнее указание параметра в строке соединения. Поэтому, если такие параметры как **Имя пользователя** или **Пароль** указаны и в строке соединения и в диалоге **Параметры соединения** (см. [здесь](#)), то будут использоваться значения из диалога **Параметры соединения**, т. к. эти параметры будут добавлены с правой стороны к указанной строке соединения.

Для упрощения формирования строки соединения можно воспользоваться сайтом <http://www.connectionstrings.com/>.

24.1.5.2. Описание источника данных

Вместо указания полной строки соединения, можно воспользоваться специальными утилитами, которые могут сформировать описание источника данных (DSN) и указывая в параметрах соединения специальную конструкцию вида **DSN=ИмяDSN**. Описание источника данных могут быть пользовательскими и системными. Пользовательские описания источников данных доступны на данном компьютере и только тому пользователю, который создавал этого описание. Системное описание создается системным администратором и доступно всем пользователям данного компьютера (при наличии соответствующих разрешений).

24.1.5.3. Утилита администрирования источников данных ODBC

Для создания описания источника данных следует воспользоваться утилитой администрирования источников данных ODBC. В ОС Windows эта утилита расположена следующим образом: **Панель управления – Администрирование – Источники данных (ODBC)**. При этом из панели управления Windows открывается утилита администрирования, соответствующая версии операционной системы. В ОС Linux эта утилита называется **ODBCConfig** и доступна при установке соответствующего пакета (зависит от версии ОС Linux).

Следует помнить, что в 64-разрядной версии ОС Windows драйвера и источники данных различаются для 32-разрядного и 64-разрядного ODBC. Поэтому для создания описания источника данных следует использовать версию утилиты администрирования, соответствующую версии системы «1С:Предприятия», которая будет исполнять запросы к внешнему источнику данных. Так, например, если обращение к ODBC выполняется из 64-разрядной версии сервера «1С:Предприятия», то надо использовать 64-разрядную версию утилиты администрирования. Если используется файловая версия «1С:Предприятия» на 64-разрядной ОС Windows, то следует использовать 32-разрядную версию утилиты администрирования, т. к. клиентское приложение является 32-разрядным.

На 64-разрядной версии ОС Windows утилиты администрирования расположены следующим образом:

- 64-разрядная версия: **%SYSTEMROOT%\System32\odbcad32.exe**.
- 32-разрядная версия: **%SYSTEMROOT%\SysWOW64\odbcad32.exe**.

24.2. Работа с внешним источником данных OLAP

24.2.1. Общая информация

OLAP – это технология обработки данных, заключающаяся в подготовке агрегированной информации на основе больших массивов данных, структурированных по многомерному принципу. Данные в аналитических базах данных (OLAP-системах) формируются на основании данных систем транзакционной обработки данных (также называемых OLTP-системами). Информация в OLAP-системах представлена в виде пространства (называемого **кубом**), оси которого представляют собой **измерения**, а в узлах этого пространства располагаются некоторые **меры** (measures). Каждое измерение куба характеризуется определенными **членами** (members) измерения.

Ближайшим (но не точным!) аналогом хранения данных в OLAP-системах можно считать регистр накопления, где сам регистр подобен кубу, измерение типа [СправочникСсылка](#) подобно измерению куба, сам справочник содержит члены измерения, а ресурсы регистра подобны мерам куба.

В системе «1С:Предприятие» куб OLAP-системы представлен в виде модели, где:

- Куб представлен кубом в конфигурации;
- Измерения и члены измерений представлены в виде двух объектов:
 - Измерения являются аналогом измерений,
 - Таблицы измерений служат для описания коллекции членов измерений.
- Меры представлены ресурсами.

Таким образом, в состав внешнего источника, который подключается к OLAP-системе, входят кубы, каждый куб состоит из таблиц измерений, собственно измерений и ресурсов. Таблицы измерений, в свою очередь, состоят из полей. Ресурсы кубов могут быть числовыми или строковыми.

Объект, описывающий внешний источник данных, может быть использован следующим образом:

- в качестве источника данных для запросов;
- в качестве источника данных в системе компоновки данных;
- в качестве источника для динамических списков;
- входить в состав общих реквизитов (см. [здесь](#));
- записи таблиц измерений и кубов могут отображаться в управляемых формах «1С:Предприятия» (не поддерживается использование обычных форм для объектов внешних источников данных);
- таблицы измерений внешнего источника данных могут выступать в качестве типов реквизитов информационной базы;
- к кубам, таблицам измерений, полям таблицы измерений, измерениям и ресурсам внешних источников данных можно применять права доступа;
- доступ к кубам, таблицам измерений, полям таблицы измерений, измерениям и ресурсам возможен из встроенного языка;
- кубы и таблицы измерений внешнего источника данных могут входить в состав подсистем (см. [здесь](#));
- кубы, таблицы измерений и измерения внешнего источника данных может входить в состав функциональных опций (см. [здесь](#));
- для кубов внешнего источника данных можно создавать характеристики (см. [здесь](#)).

Для работы с многомерными внешними источниками данных используется механизм XML для аналитики (XML for Analysis, XMLA). Платформа получает доступ к данным с помощью http-запросов к веб-серверу, который может быть внешним (по отношению к OLAP-системе) или встроенным в OLAP-сервер. Вопросы настройки доступа к данным OLAP-сервера следует искать в документации к используемой системе.

Строка соединения с аналитическим внешним источником данных представляет собой комбинацию URL поставщика XMLA (для данной OLAP-системы) и параметров, которые используются «1С:Предприятием» для работы с источником. Данные внешних источников данных доступны только для чтения.

Возможности системы «1С:Предприятие» максимально учитываются при использовании в качестве внешнего источника данных следующих OLAP-систем (свойство **Тип СУБД** параметров соединения с источником данных):

- IBM Infosphere Warehouse,
- Microsoft SQL Server Analysis Services,
- Oracle Essbase.

ПРИМЕЧАНИЕ. При работе с внешним источником данных могут возникать ошибки в том случае, если свойство **Тип СУБД** параметров соединения с внешним источником данных содержит значение, которое не соответствует реально используемой системе.

При использовании других СУБД возможности работы с внешними источниками данных зависят от самой СУБД. Для осуществления подключения к внешнему источнику данных следует сформировать строку подключения. Подробнее о строке подключения см. [здесь](#).

Также следует помнить, что соединение с внешним источником данных следует указывать не только в конфигураторе (если используется механизм импорта структуры таблиц из внешнего источника данных), но и в режиме 1С:Предприятие для получения собственно данных.

Документация по OLAP-системам:

- IBM Infosphere Warehouse:
 - http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.dwe.navigate.doc/welcome_db2warehouse.html (на английском языке).
- Microsoft SQL Server Analysis Services:

- Версия 2008: [http://technet.microsoft.com/ru-ru/library/bb522607\(v=sql.100\).aspx](http://technet.microsoft.com/ru-ru/library/bb522607(v=sql.100).aspx);
- Версия 2008 R2: [http://technet.microsoft.com/ru-ru/library/bb522607\(v=sql.105\).aspx](http://technet.microsoft.com/ru-ru/library/bb522607(v=sql.105).aspx);
- Версия 2012: [http://technet.microsoft.com/ru-ru/library/bb522607\(v=sql.110\).aspx](http://technet.microsoft.com/ru-ru/library/bb522607(v=sql.110).aspx).
- Oracle Essbase:
 - http://docs.oracle.com/cd/E17236_01/nav/portal_3.htm (на английском языке).

24.2.2. Общая схема использования

Для того чтобы использовать информацию из внешних источников данных в системе на базе «1С:Предприятия», следует придерживаться следующей общей схемы:

- Изучить структуру внешнего источника данных и понять, какая информация (кубы, таблицы измерений, измерения и ресурсы) необходима для работы алгоритмов или отчетов в «1С:Предприятии».
- Создать объект **Внешние источники данных** в конфигураторе и для него создать требуемую структуру. Это можно сделать с помощью специального помощника.
- Реализовать использование созданных объектов в прикладном решении.
- Выполнить настройку параметров подключения к внешнему источнику данных в той сети, где будет использоваться прикладное решение. Эти параметры могут отличаться от тех, которые использовались при загрузке структуры внешнего источника данных.

ПРИМЕЧАНИЕ. Параметры доступа к внешнему источнику данных, которые были заданы в конфигураторе, не будут использованы системой в режиме 1С:Предприятие.

24.2.3. Редактирование структуры внешнего источника данных

Для работы с таблицами внешних источников данных предназначена ветвь с именем соответствующего внешнего источника данных ветви **Внешние источники данных – Кубы** дерева конфигурации. Определение внешнего источника данных состоит из нескольких шагов:

- Определение собственно внешнего источника данных.
- Определение кубов;
- Определение таблиц измерений и полей для каждой таблицы измерений;
- Определение измерений для куба;
- Определение ресурсов для куба.

При этом в конфигурации может быть описано меньшее количество объектов, чем есть в реальной базе данных, но нельзя создать объект, которое отсутствует в реальной базе данных.

Создание структуры внешнего источника данных возможно как в ручном режиме, так и при загрузке этой структуры с помощью специального конструктора.

24.2.3.1. Внешний источник данных

При создании внешнего источника данных следует указать его имя. Этот объект предназначен для идентификации группы кубов при обращении к данным, которые в нем содержатся. Источник данных состоит из одного или нескольких кубов, которые, в свою очередь, состоят из других объектов (таблицы измерений, измерения, ресурсы).

24.2.3.2. Куб внешнего источника данных

При создании нового куба открывается окно редактирования объекта (см. [здесь](#)).

Свойство таблицы **Имя** служит для идентификации куба внутри прикладного решения. Для того чтобы система знала, какой физический куб внешнего источника данных соответствует объект конфигурации, следует заполнить свойство **Имя в источнике данных**. Значение в этом свойстве в точности должно соответствовать имени куба внешнего источника данных. Не поддерживается размещение в одном кубе прикладного решения данных из нескольких кубов внешнего источника данных.

24.2.3.3. Таблица измерения

Таблица измерения описывает члены (**members**) измерения куба OLAP-системы. Свойство **Имя в источнике данных** содержит точное имя измерения или уровня иерархии в источнике данных. Свойство **Поле представления** содержит имя свойства измерения или уровня иерархии, которое система «1С:Предприятие» использует для формирования представления элемента таблицы измерения.

Если измерение является иерархическим, то для такого измерения может быть использована иерархическая таблица измерения. Иерархическая таблица измерения описывается с помощью следующих свойств:

- **Иерархическая** – содержит признак того, что таблица измерения описывает иерархию в источнике данных.
- **Имя иерархии в источнике данных** – указывает имя иерархии, к которой относится таблица измерения, описывающая данный уровень.
- **Номер уровня** – в том случае, если таблица описывает какой-либо уровень иерархии, данное свойство содержит номер этого уровня. Для собственно иерархической таблицы значение этого свойства равно 0, а свойство **Имя иерархии в источнике данных** совпадает со значением свойства **Имя в источнике данных**.

Свойства **Признак незаполненного родителя** и **Значение незаполненного родителя** позволяют указать системе «1С:Предприятие», каким образом идентифицировать значения самого верхнего уровня иерархии.

24.2.3.4. Поле таблицы измерений

Поля таблицы измерений описывают, какие свойства измерений будут доступны из прикладного решения. Свойство поля **Имя** служит для идентификации свойства (поля таблицы измерения) в прикладном решении. Для указания соответствия между полем прикладного решения и атрибутом измерения служит свойство **Имя в источнике данных**. Значение в этом свойстве должно в точности соответствовать имени атрибута измерения, указанного в свойстве **Имя в источнике данных** родительского объекта данного поля. Не поддерживается объединение в одном поле таблицы измерений данных из нескольких атрибутов одного или нескольких измерений внешнего источника данных.

Поле **Тип** позволяет указать тип данного поля. Для выбора доступен ограниченный набор типов. Типами поля могут быть:

- Число;
- Строка;
- Дата;
- Булево;
- УникальныйИдентификатор;
- ДвоичныеДанные;
- типы, которые определяются таблицами измерений внешних источников данных.

Если необходимо указать для поля составной тип, то в таком составном типе могут участвовать только типы **Число**, **Строка**, **Дата**, **Булево**.

При получении данных из внешних источников система автоматически преобразует эти данные к тому типу, который указан у соответствующего поля в прикладном решении. Описание правил преобразования см. [здесь](#).

24.2.3.5. Измерения

Измерение внешнего источника данных описывает измерение куба OLAP-системы. Типом измерения может быть только ссылка на соответствующую таблицу измерения. Фактически, таблицы измерений и сами измерения, соотносятся один к одному: одно измерение описывается одной таблицей измерения.

Не поддерживается объединение в одном поле измерения данных из нескольких измерений внешнего источника данных.

24.2.3.6. Ресурсы

Ресурс внешнего источника данных описывает факт куба OLAP-системы. Ресурс куба может быть числовым или строковым. Именно значения ресурсов рассчитываются по значениям измерений куба. Свойство **Имя в источнике данных** должно совпадать с именем факта (ресурса) куба внешнего источника данных.

Не поддерживается объединение в одном ресурсе данных из нескольких фактов (ресурсов) внешнего источника данных.

24.2.3.7. Загрузка структуры куба из внешнего источника данных

Существует возможность загрузки структуры куба из внешнего источника данных, который доступен для разработчика (например, копии реальной базы данных).

Для выполнения этой операции следует при создании куба указать пункт **Выбрать из списка кубов внешнего источника данных** в конструкторе кубов внешнего источника данных. Затем нужно указать строку соединения с внешней базой данных, воспользовавшись окном **Подключение к источнику данных**. При формировании строки подключения можно воспользоваться конструктором строки соединения с внешним источником данных. Для этого следует нажать кнопку «...» справа от поля **Строка соединения**.

После того как подключение к внешней базе успешно выполнено, будет открыт список с перечнем кубов, таблиц измерений, полей таблиц измерений и ресурсов подключенного источника данных. Затем следует выбрать требуемый куб, а также таблицы измерений, поля и ресурсы, которые будут использованы конфигуратором для создания структуры объектов, описывающих текущий источник данных.

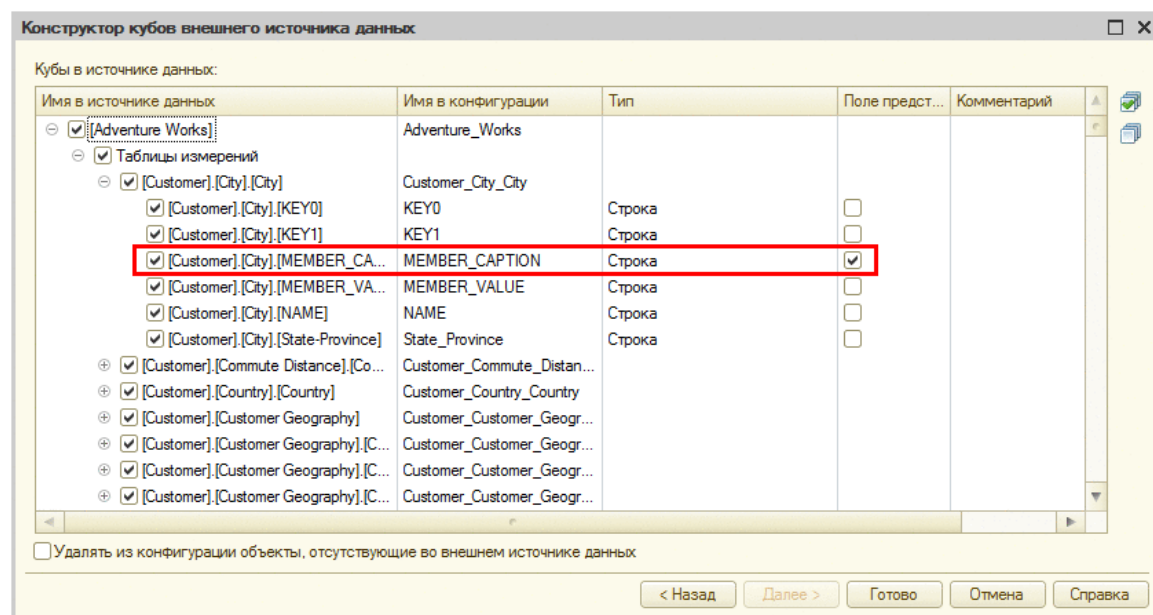


Рис. 346. Конструктор кубов внешнего источника данных

При получении структуры внешнего источника данных система выполняет следующие действия:

- Выполняет попытку определить поля, формирующие представления для таблиц измерений. Если система ошиблась, поле представления можно указать вручную.

- Преобразует типы колонок источника данных в типы «1С:Предприятия», которые будут использоваться для указания типов реквизитов объектов. Все типы из внешнего источника данных преобразуются к следующему типу «1С:Предприятия»: число, строка, дата, булево, уникальный идентификатор, двоичные данные и типы, связанные с таблицами измерений источника данных.

- Выполняет попытку определить типы для полей таблиц измерений. В этом случае система пытается определить, какого типа данные хранятся в колонке таблицы измерения, и в том случае, если это можно считать ссылкой на данные другой таблицы измерения, указывает в колонке соответствующий тип. Если система неправильно выбрала тип колонки таблицы, его можно изменить вручную.

Затем следует отметить флажками объекты, которые будут перенесены в метаданные прикладного решения.

Если флажок [Удалять из конфигурации объекты, отсутствующие во внешнем источнике данных](#) установлен, то при завершении работы конструктора кубов из конфигурации будут удалены те объекты (кубы, таблицы измерений, поля таблицы измерений, измерения, ресурсы), которые отсутствуют во внешнем источнике (например, во внешнем источнике куб или ресурс были удалены) данных.

После нажатия кнопки [Готово](#) произойдет загрузка структуры внешнего источника данных.

24.2.4. Ограничения языка запросов при использовании аналитического внешнего источника данных

При работе с кубами и таблицами измерений в языке запросов, следует учитывать следующие ограничения:

Выражение	Microsoft SQL Server Analysis Services	Oracle Essbase	IBM Infosphere Warehouse
РАЗРЕШЕННЫЕ	–	–	–
РАЗЛИЧНЫЕ	–	–	–
СТРУППИРОВАТЬ	–	–	–
ИМЕЮЩИЕ	–	–	–
СОЕДИНЕНИЕ	–	–	–
ОБЪЕДИНИТЬ	–	–	–
Подзапросы	–	–	–
ДЛЯ ИЗМЕНЕНИЯ	–	–	–
ПУСТАЯТАБЛИЦА	–	–	–
ГОД	Да	–	–
КВАРТАЛ	Да	–	–
МЕСЯЦ	Да	–	–
ДЕНЬГОДА	Да	–	–
ДЕНЬ	Да	–	–
НЕДЕЛЯ	Да	–	–
ДЕНЬНЕДЕЛИ	Да	–	–
ЧАС	Да	–	–
МИНУТА	Да	–	–
СЕКUNДА	Да	–	–
НАЧАЛОПЕРИОДА	Да	–	–
КОНЕЦПЕРИОДА	Да	–	–
ДОБАВИТЬКДАТЕ	Да	–	–
РАЗНОСТЬДАТ	Да	–	–

Где,

- – – указанное выражение не поддерживается при работе с соответствующей OLAP-системой;
- **Да** – указанное выражение поддерживается при работе с соответствующей OLAP-системой.

24.2.5. Строка соединения с OLAP-сервером

Строка соединения с OLAP-сервером имеет вид:

[Копировать в буфер обмена](#)

http://<Адрес хоста OLAP>:<порт>/<источник>?<параметры>

Где:

- **адрес хоста OLAP**, **порт**, **источник** – адрес доступа к OLAP-системе, сформированный по правилам, описанным в документации конкретной OLAP-системе.
- **параметры** – параметры, используемые «1С:Предприятием» для доступа к данным OLAP-системы. Параметры задаются в виде **Параметр=Значение**. Параметры разделяются символом **&**. Используются следующие параметры:
 - **ProviderName** – имя сервиса XMLA OLAP-источника;
 - **DataSourceName** – имя OLAP-источника;

- *Catalog* – имя каталога, или базы данных, OLAP-источника.

Примеры строк соединения приведены ниже:

Для *Microsoft SQL Server Analysis Services*:

[http://localhost:80/msolap/msmdpump.dll?ProviderName=Microsoft Analysis Services& DataSourceName=host&Catalog=Adventure Works DW](http://localhost:80/msolap/msmdpump.dll?ProviderName=Microsoft%20Analysis%20Services&DataSourceName=host&Catalog=Adventure%20Works%20DW) [Копировать в буфер обмена](#)

Для *Oracle Essbase*:

[http://localhost:13080/aps/XMLA?ProviderName=Essbase XML for Analysis&DataSourceName=host&Catalog=Sample](http://localhost:13080/aps/XMLA?ProviderName=Essbase%20XML%20for%20Analysis&DataSourceName=host&Catalog=Sample) [Копировать в буфер обмена](#)

24.3. Управление внешними источниками данных

Перед тем как начать использование данных из внешних источников в прикладном решении, следует настроить параметры подключения используемых внешних источников. Для этого служит стандартная функция [Управление внешними источниками данных](#), которая вызывается из окна [Все функции – Стандартные функции](#).

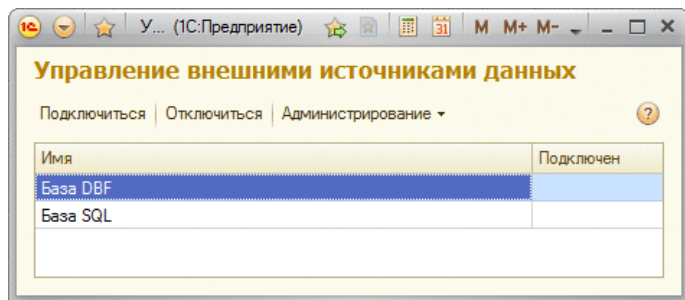


Рис. 347. Управление внешними источниками данных

В данной форме перечислены все внешние источники данных, созданные в конфигураторе и доступные пользователю, который открыл форму.

Колонка [Подключен](#) указывает состояние подключения к источнику данных в этом сеансе.

С помощью меню [Администрирование](#) можно указать общие параметры подключения (команда [Изменить общие параметры...](#)) и параметры подключения конкретных пользователей в том случае, если эти параметры отличаются от общих параметров (команда [Изменить параметры пользователей...](#)).

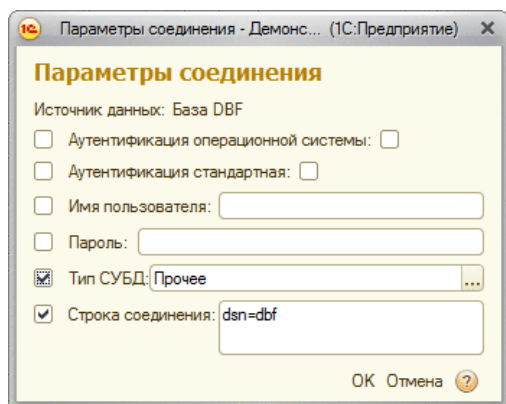


Рис. 348. Параметры соединения

Флажок в левой части формы (перед именами параметров) означает, что данный параметр используется в данном наборе параметров соединения. При этом параметры анализируются в следующем порядке:

- параметры, установленные для сеанса с помощью метода [УстановитьПараметрыСоединенияСеанса\(\)](#) менеджера внешнего источника данных;
- параметры, указанные для пользователя при интерактивной настройке или с помощью метода [УстановитьПараметрыСоединенияПользователя\(\)](#) менеджера внешнего источника данных;
- общие параметры, заданные интерактивно или с помощью вызова метода [УстановитьОбщиеПараметрыСоединения\(\)](#) менеджера внешнего источника данных.

Результирующий набор параметров будет использован для соединения с внешним источником данных. Если какой-то параметр указан в нескольких настройках, будет использовано то значение параметра, которое встретится в вышеописанной цепочке раньше.

Например, можно следующим образом комбинировать параметры подключения:

- в параметрах соединения сеанса указывается пользователь и его пароль;
- в общих параметрах указывается строка соединения с базой данных внешнего источника;
- в параметр конкретного пользователя указывается другая строка соединения, которая может использоваться для тестирования прикладного решения.

При указании строки соединения следует помнить, что доступ к внешним данным будет выполняться не с компьютера, где происходит настройка,

а с другого компьютера (описание того, с какого компьютера выполняется запрос к внешнему источнику данных, см. [здесь](#)). Подробнее о строке соединения см. [здесь](#).

Нажатие кнопки **Подключиться** открывает диалог, в котором можно задать (или уточнить) параметры подключения к выбранному внешнему источнику данных.

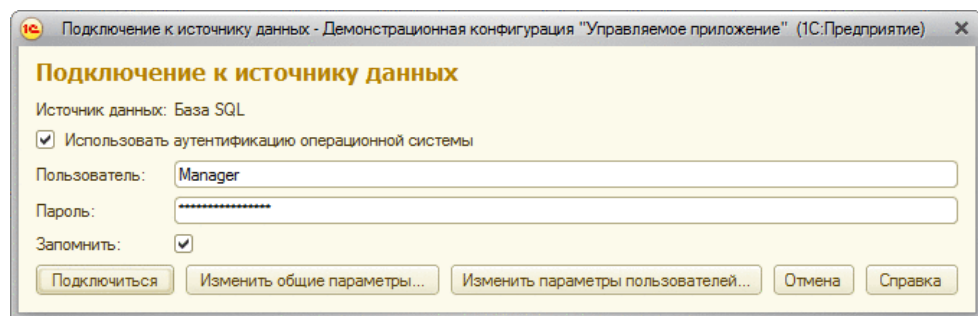


Рис. 349. Параметры подключения

Необходимость указания пользователя и пароля в данном диалоге зависит от драйвера ODBC и содержания строки соединения. Возможны ситуации, когда ввод логина и пароля не требуется. Если пользователь не является администратором внешнего источника данных, то кнопки **Общие параметры...** и **Параметры пользователя...** для него недоступны.

Флажок **Использовать аутентификацию операционной системы** доступен только в том случае, если у пользователя имеется право [ИзменениеАутентификацииОСсеанса](#).

После нажатия кнопки **Подключиться** происходит подключение к внешнему источнику данных. В случае успешного подключения изменяется содержимое колонки **Подключен** списка внешних источников данных.

Нажатие кнопки **Отключиться** приведет к отключению «1С:Предприятия» от выбранного внешнего источника данных.

24.4. Подключение к внешнему источнику данных в режиме «1С:Предприятие»

При выполнении любой операции, в ходе которой необходимо получать данные из внешнего источника данных, происходит попытка подключения, если подключение к этому источнику данных еще не было выполнено. Если подключение выполнено успешно, то выполняемые действия продолжают. Если подключение не выполнено, вызывается исключение.

В случае если в клиентском приложении возникает исключение, связанное с подключением к внешнему источнику данных, пользователю предлагается диалог, в котором можно уточнить параметры подключения (если пользователь имеет права на изменение параметров) и повторить подключение. Если подключение выполнено успешно, то пользователю будет предложено повторить действие, во время которого произошла ошибка подключения.

ПРИМЕЧАНИЕ. В режиме совместимости с версией 8.2.13, при работе с динамическим списком, в некоторых случаях диалог подключения к внешним данным может не показываться.

Также имеется возможность выполнить подключение вручную. Для этого предназначена стандартная функция [Управление внешними источниками данных](#).

24.5. Особенности использования внешних источников данных

24.5.1. Место исполнения запросов к внешним источникам данных

Исполнение запросов к внешним источникам данных выполняется:

- в файловом варианте – на компьютере с клиентским приложением;
- в файловом варианте с доступом через веб-сервер – на компьютере, где выполняется расширение веб-сервера;
- в клиент-серверном варианте – на компьютере, где выполняется:
 - сервис работы с внешними источниками данных через ODBC – для доступа к реляционным источникам данных;
 - сервис работы с внешними источниками данных через XMLA – для доступа к аналитическим источникам данных (OLAP-системы).

24.5.2. Использование внешних источников данных

В том случае, если один внешний источник данных содержит в своем составе и данные реляционных источников данных и данные аналитических источников данных, их одновременное использование не поддерживается. Это связано с тем, что один внешний источник данных может быть в один момент времени подключен или к реляционной или к аналитической СУБД.

24.5.2.1. В запросах

24.5.2.1.1. К реляционным внешним источникам

Таблицы внешних источников данных могут выступать в качестве источников запросов. При этом таблица внешнего источника данных описывается следующим образом:

ВнешнийИсточникДанных.<Имя источника>.Таблица.<Имя таблицы>

[Копировать в буфер обмена](#)

Пример:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    Товары.Код,
    Товары.Наименование,
    Товары.Артикул
ИЗ
    ВнешнийИсточникДанных.DBF.Таблица.Товары КАК Товары

```

В данном примере происходит выборка полей **Код**, **Наименование** и **Артикул** из таблицы **Товары** внешнего источника данных **DBF**.

В запросе к внешнему источнику данных возможно использование временных таблиц. При этом производится попытка создания временной таблицы непосредственно в базе данных, связанной с внешним источником данных. Если СУБД не поддерживает создание внешних источников данных – будет вызвано исключение. Имя временной таблице, при обращении к ней, формируется следующим образом:

```

ВнешнийИсточникДанных.<Имя внешнего источника данных>.ВременнаяТаблица.<Имя временной таблицы>

```

[Копировать в буфер обмена](#)

Пример:

```

ВЫБРАТЬ Name, ProductID
ПОМЕСТИТЬ ВнешнийИсточникДанных.AdventureWorks.ВременнаяТаблица.Остатки
ИЗ &ТаблицаЗначений
ВЫБРАТЬ Name, ProductID
ПОМЕСТИТЬ ВнешнийИсточникДанных.AdventureWorks.ВременнаяТаблица.Остатки
ИЗ ВнешнийИсточникДанных.AdventureWorks.Таблица.Production_Balance

```

[Копировать в буфер обмена](#)

При использовании в запросах внешних источников данных нужно помнить следующие ограничения:

- В одном запросе можно использовать один источник данных. Не поддерживается одновременное использование, например, внешнего источника данных и данных информационной базы «1С:Предприятия».

24.5.2.1.2. К аналитическим внешним источникам

Объекты аналитических внешних источников данных могут выступать в качестве источников запросов. При этом куб внешнего источника данных описывается следующим образом:

```

ВнешнийИсточникДанных.<Имя источника>.Куб.<Имя куба>

```

[Копировать в буфер обмена](#)

Пример:

```

ВЫБРАТЬ ПЕРВЫЕ 10
AdventureWorks.InternetSalesAmount
ИЗ
ВнешнийИсточникДанных.OLAP.Куб.AdventureWorks КАК AdventureWorks

```

[Копировать в буфер обмена](#)

В данном примере выполняется получение 10 первых ресурсов **InternetSalesAmount** из куба **AdventureWorks** внешнего источника данных **OLAP**.

ВНИМАНИЕ! Не рекомендуется получать **все** записи из куба внешнего источника данных. Количество записей очень велико и может привести к перегрузке системы.

Таблица измерений куба внешнего источника данных описывается следующим образом:

```

ВнешнийИсточникДанных.<Имя источника>.Куб.<Имя куба>.ТаблицаИзмерений.<Имя таблицы измерений>

```

[Копировать в буфер обмена](#)

Пример:

```

ВЫБРАТЬ
AdventureWorksТаблицаИзмеренияProduct.Ссылка
ИЗ
ВнешнийИсточникДанных.OLAP.Куб.AdventureWorks.ТаблицаИзмерения.Product КАК AdventureWorksТаблицаИзмеренияProduct

```

[Копировать в буфер обмена](#)

В данном примере происходит получения списка ссылок на членов таблицы измерения **Product** из куба **AdventureWorks** внешнего источника данных **OLAP**.

В запросе к внешнему источнику данных возможно использование временных таблиц. При этом производится попытка создания временной таблицы непосредственно в базе данных, связанной с внешним источником данных. Если СУБД не поддерживает создание внешних источников данных – будет вызвано исключение. Имя временной таблице, при обращении к ней, формируется следующим образом:

```

ВнешнийИсточникДанных.<Имя внешнего источника данных>.ВременнаяТаблица.<Имя временной таблицы>

```

[Копировать в буфер обмена](#)

При использовании в запросах внешних источников данных нужно помнить следующие ограничения:

- В одном запросе можно использовать один источник данных. Не поддерживается одновременное использование, например, внешнего источника данных и данных информационной базы «1С:Предприятия».

24.5.2.2. В системе компоновки данных

В системе компоновки данных возможно использование наборов данных, получающих данные из разных однотипных источников данных. Так, можно создать набор данных, описывающий список товаров. Данные в этом наборе будут получены из информационной базы «1С:Предприятия». Другой набор данных может описывать данные, полученные из внешнего источника данных, например таблица продаж товаров из внешней базы данных.

Затем можно настроить связи наборов данных в конструкторе схемы компоновки данных и получать в отчет информацию одновременно из «1С:Предприятия» и внешнего источника данных.

24.5.3. Правила преобразования значений

При получении данных из внешних источников система автоматически преобразует эти данные к тому типу, который указан у соответствующего поля в прикладном решении. Преобразование выполняется по следующим правилам:

- Тип **Строка**:
 - Строковое значение преобразуется в строковое значение.
 - Числовое значение преобразуется в строковое значение (для преобразования используются региональные настройки операционной системы на том компьютере, где происходит выполнение запроса к внешнему источнику данных).
 - Значение даты преобразуется в строковое значение (для преобразования используются региональные настройки операционной системы на том компьютере, где происходит выполнение запроса к внешнему источнику данных).
 - Значение уникального идентификатора преобразуется в строковое представление уникального идентификатора.
 - Двоичные данные (и данные других типов) преобразуются в строку, которая является текстовым представлением преобразуемых данных в шестнадцатеричном виде.
- Тип **Число**:
 - Числовое значение преобразуется в число.
 - Для строкового значения выполняется попытка преобразования строки в число в соответствии с региональными настройками операционной системы компьютера, на котором выполняется запрос к внешнему источнику данных.
 - Для остальных типов возникает исключение.
- Тип **Булево**:
 - Числовое значение преобразуется в значение **Ложь**, если его значение равно 0 и в значение **Истина** – для любого другого значения во внешнем источнике.
 - Двоичные данные преобразуются по значению первого байта данных. Если первый байт равен значению 0, то результатом преобразования является значение **Ложь** и значение **Истина** в случае любого другого значения.
 - Для остальных типов возникает исключение.
- Тип **ДвоичныеДанные**:
 - В двоичные данные преобразуются значения любых типов.
- Тип **Дата**:
 - Значение даты преобразуется в значение типа **Дата**.
 - Для остальных типов возникает исключение.

24.5.4. Внешний источник данных, входящий в состав разделителя

Если внешний источник данных входит в состав общего реквизита, то следует учитывать следующие особенности:

- Изменение структуры таблиц внешнего источника данных не производится, поэтому собственно данные не разделяются.
- Параметры соединения с внешним источником данных хранятся в разрезе значений разделителей, в состав которых входит внешний источник данных.
- Неиспользуемый разделитель является отдельным значением, в разрезе которого хранятся параметры соединения с внешним источником данных.

Таким образом, включая внешний источник данных в состав разделителя, появляется возможность реализовать один из следующих сценариев использования:

- Данные, доступные для пользователей различных областей данных, расположены в разных базах данных, которые совпадают по структуре таблиц. Т.к. параметры соединения хранятся в разрезе значений разделителя, имеется возможность для каждой области данных настроить работу с необходимой базой данных.
- Данные, доступные для пользователей различных областей расположены в одной базе данных, но доступ к данным регулируется средствами СУБД (имя и пароль пользователя). В этом случае имеется возможность для каждой области данных указать свои имя и пароль пользователя, от имени которого выполняется доступ к данным.

Приведенные сценарии не являются единственными. Они приведены как пример использования внешних источников данных, входящих в состав разделителей.

Глава 25. Механизм разделения данных

Механизм **разделения данных** – специальный механизм, позволяющий разделить все хранимые данные, а также работу прикладного решения на отдельные части.

25.1. Общая информация

Если у общего реквизита свойство **Разделение данных** установлено в значение **Разделять**, то включается режим разделения данных. При этом:

- изменяется поведение объектов конфигурации, входящих в состав общего реквизита (далее – **разделителя**);
- для каждого разделителя в информационной базе определяется текущее значение разделителя и признак использования данного разделителя;
- объекты конфигурации, входящие в состав такого общего реквизита, называются **разделенными объектами конфигурации**;
- конфигурация с такими общими реквизитами считается **разделенной конфигурацией**;
- информационная база с разделенной конфигурацией называется **разделенной информационной базой**;
- данные информационной базы, доступные для выбранных значений разделителей, и данные неразделенных объектов конфигурации называются **областью данных**.

Во всех операциях чтения записей базы данных «1С:Предприятие» автоматически отбирает только те записи, в которых значения используемых разделителей совпадают с текущими значениями разделителей. При записи данных платформа проверяет, что записываемые данные содержат значения используемых разделителей, равные текущим значениям разделителей или равные значениям по умолчанию для соответствующих типов. Если это не так, то вызывается исключение.

Например:

[Копировать в буфер обмена](#)

```
// Изначально объект записан со значением разделителя Абонент = 1
Объект = СсылкаНаОбъект.ПолучитьОбъект();
// Меняем значение разделителя в текущем сеансе
ПараметрыСеанса.АбонентЗначение = 2;
// Объект не будет записан, и будет вызвано исключение
Объект.Записать();
```

Если записываемые данные содержат значения разделителей, равные значениям по умолчанию для соответствующих типов, то в разделители объекта будут записаны значения используемых разделителей текущего сеанса. Для неиспользуемых в текущем сеансе разделителей:

- при чтении данных отбор по этим разделителям не выполняется;
- при модификации данных используются значения, содержащиеся в записываемых данных.

В состав разделителя могут входить следующие объекты конфигурации:

- константы,
- справочники,
- документы,
- последовательности,
- журналы документов,
- планы видов характеристик,
- планы счетов,
- планы видов расчета,

- бизнес-процессы,
- задачи,
- регистры сведений,
- регистры накопления,
- регистры бухгалтерского учета,
- регистры расчета,
- планы обмена,
- регламентные задания,
- пользователи информационной базы (см. [здесь](#)).

Следует помнить, что последовательности и журналы документов являются разделенными объектами только в том случае, если в их состав входят разделенные объекты конфигурации. Состав разделителей для объектов, входящих в состав последовательности или журнала документов, должен быть идентичным.

Разделитель в режиме **Независимо и совместно** может использоваться в ограничениях доступа к данным (см. [здесь](#)).

Тип разделителя **не может быть составным**.

Если общий реквизит выступает в роли разделителя, то у него появляется несколько дополнительных свойств, влияющих на поведение системы, а также изменяется поведение объектов конфигурации, входящих в состав разделителя. В следующих разделах эти свойства будут рассмотрены подробнее.

Свойства: Организация

▼ Основные:

Имя: Организация

Синоним: Организация

Комментарий:

Тип: СправочникСсылка.Организации

▼ Использование:

Состав: Открыть

Автоиспользование: Использовать

Разделение данных: Разделять

Использование разделяемых: Независимо

Значение разделения данных: ОрганизацияЗначение ... X

Использование разделения д: ОрганизацияИспользуйва ... X

Условное разделение: ... X

Разделение пользователей: Не использовать

Разделение аутентификации: Не использовать

Индексировать: Не индексировать

Используется платформой для разделения данных между ИБ

Рис. 350. Общий реквизит с разделением

ВНИМАНИЕ! Если использовать разделитель с типом **Строка**, не допускается изменение языка региональных настроек информационной базы при работе с Microsoft SQL Server.

ВНИМАНИЕ! Не поддерживается разделение данных на СУБД Microsoft SQL Server 2000, IBM DB2 9.1, а также на СУБД IBM DB2 других версий, если в СУБД установлено сравнение строк, чувствительное к регистру букв (параметр `COLLATION_SEQUENCE` не равен значению `UCA500R1_LROOT_AN_CX_EX_FX_HX_NX_S2`).

25.2. Свойства общего реквизита

25.2.1. Использование разделяемых данных

Свойство [Использование разделяемых данных](#) определяет возможность использования данных разделенных объектов конфигурации в том случае, если в режиме 1С:Предприятие разделитель не используется:

- Значение [Независимо](#) – запрещается любое обращение к разделенным объектам (кроме планов обмена, подробнее см. [здесь](#)), если в текущем сеансе работы с информационной базой разделитель не используется. При этом разделитель недоступен в объектной модели, в языке запросов, в формате выгрузки в XML и XDTO, а также ограничениях доступа к данным. При записи разделенных данных система автоматически заполняет значения общего реквизита значениями, установленными в текущем сеансе.

Данный режим может использоваться в тех случаях, когда прикладное решение в своей работе использует только данные одной области данных.

ВНИМАНИЕ! В этом режиме **ссылки объектов** базы данных **могут быть одинаковыми** для объектов, записанных в разных областях данных.

- Значение [Независимо и совместно](#) – разрешается работа с разделяемыми объектами независимо от использования разделителя в сеансе. Для такого режима будет доступна область данных, которую определяют значения используемых разделителей. При этом разделитель доступен в объектной модели, в языке запросов, в формате выгрузки в XML и XDTO, а также в ограничениях доступа к данным. В этом режиме невозможно создание предопределенных элементов для объектов, входящих в состав такого разделителя. Если в сеансе используются не все разделители, в состав которых входит объект, то работа с этим объектом будет менее эффективной.

Данный режим может использоваться в тех случаях, когда прикладное решение в большинстве случаев работает только с данными одной области данных (например, при вводе документов), а в некоторых режимах может использовать данные нескольких или всех областей, например, для получения консолидированной отчетности.

ВНИМАНИЕ! В этом режиме **ссылки объектов** базы данных **не могут быть одинаковыми** для разных объектов, записанных в разных областях данных.

25.2.2. Значение разделителя и признак использования

Свойство [Значение разделения данных](#) определяет параметр сеанса, который хранит установленное в текущем сеансе **значение разделителя**. Тип параметра сеанса в точности должен совпадать с типом разделителя.

Свойство [Использование разделения данных](#) определяет параметр сеанса (типа [Булево](#)), который управляет использованием разделителя. Если значение указанного параметра сеанса равно [Истина](#), то это означает, что в данном сеансе **разделитель используется**.

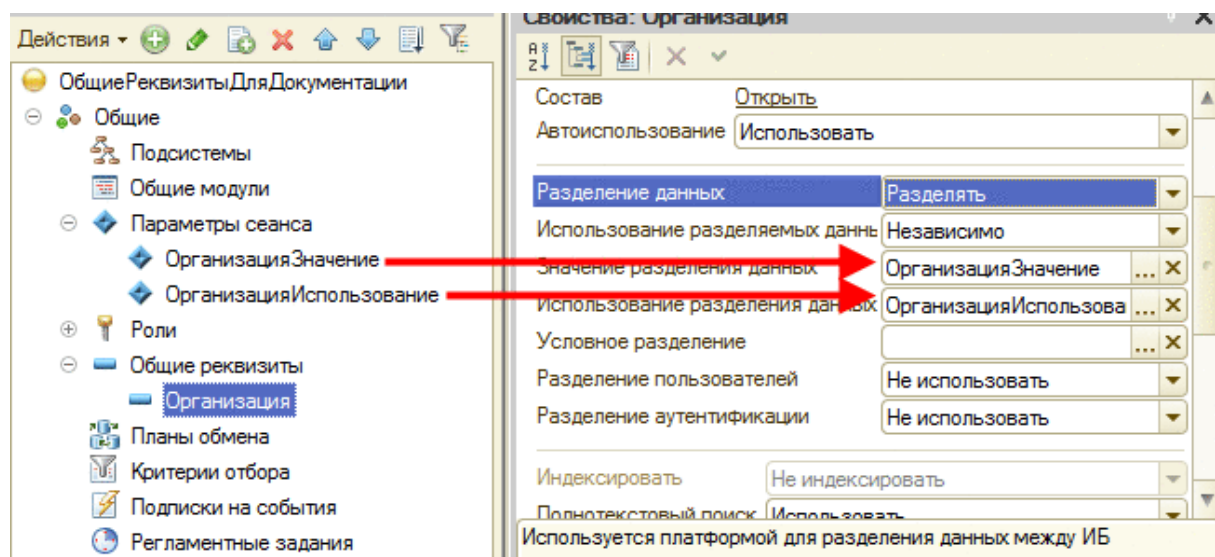


Рис. 351. Параметры сеанса и общий реквизит

Изменять значения параметров сеанса в процессе работы системы можно в том случае, если у пользователя, от лица которого работает сеанс, есть права на изменение необходимых параметров сеанса. «1С:Предприятие» не

гарантирует целостность данных приложения после изменения параметров сеанса.

Изменение из встроенного языка значения параметра сеанса, на который ссылается хотя бы один разделитель, приводит к следующим действиям:

- очищается кеш объектов;
- удаляются повторно используемые значения (см. [здесь](#)) и на стороне клиента, и на стороне сервера.

Значение разделителя можно указать в командной строке запуска «1С:Предприятия 8» с помощью параметра **Z** или в строке соединения с помощью параметра **Zn** (см. [здесь](#)).

25.2.3. Разделение пользователей

Свойство **Разделение пользователей** определяет, какая часть списка пользователей будет доступна в зависимости от того, используется данный разделитель или нет. Если свойство установлено в значение **Не использовать**, то используется единый список пользователей для любых значений разделителя. Разделение пользователей используется при программном доступе к списку пользователей, а также в конфигураторе при редактировании списка пользователей.

Для определения того, доступен пользователь в сеансе или нет, можно использовать следующую таблицу (по отношению к одному разделителю):

	Разделитель в сеансе используется	Разделитель в сеансе не используется
Значение для пользователя установлено	Доступен при равенстве значений разделителя	Доступен
Значение для пользователя не установлено	Недоступен	Доступен

При редактировании пользователя имеется возможность указания значений для всех разделителей, заданных в конфигурации, а не только тех, для которых свойство **Разделение пользователей** установлено в значение **Разделять**. Значения разделителей, указанные в свойствах пользователя информационной базы, будут использоваться для установки значений разделителей при начале сеанса этого пользователя. Значения разделителей в командной строке или строке соединения имеют более высокий приоритет, чем значения, указанные в свойствах пользователя, если пользователь имеет права на изменение соответствующих параметров сеанса.

25.2.4. Разделение аутентификации

Свойство **Разделение аутентификации** служит для управления возможностью создания одноименных пользователей для разных областей данных. Если свойство установлено в значение **Не использовать**, то уникальность имен пользователей отслеживается для всех областей данных. Если свойство установлено в значение **Разделять**, то существует возможность создавать одноименных пользователей в разных областях данных. Например, можно создать несколько пользователей **Администратор**, которые будут различаться только значениями разделителей (в том числе и для незаданного значения разделителя).

Если существует разделитель **Абонент** (типа **Число**) и для него значение свойства **Разделение аутентификации** установлено в значение **Разделять**, то имеется возможность создать следующий список пользователей:

Имя пользователя	Значение разделителя Абонент
Администратор	Не задано
Администратор	1634
Администратор	2245
Администратор	1245

Если значение свойства **Разделение аутентификации** для этого разделителя установлено в значение **Не использовать**, то такой список создать нельзя. Имена пользователей должны быть уникальны, например:

Имя пользователя	Значение разделителя Абонент
АдминистраторИБ	Не задано

Администратор1634	1634
Администратор2245	2245
Администратор1245	1245

25.2.5. Условное разделение

25.2.5.1. Общее описание

Условное разделение необходимо в тех случаях, когда прикладное решение предполагается использовать как в разделенном, так и в неразделенном вариантах. Например, прикладное решение может использоваться для автономной работы – в этом случае разделители не используются, но в самом решении они предусмотрены, а может использоваться для работы некоторого количества абонентов, никак не связанных друг с другом (с независимым набором данных в информационной базе). Такое поведение можно реализовать с помощью условного разделения: в случае автономного использования разделение будет выключено, а в разделенном режиме – включено.

Условное разделение устроено следующим образом:

- для разделителя (или объекта, входящего в состав разделителя) возможно задание объекта, который хранит состояние разделения по данному разделителю;
- изменяя значение этого объекта, можно включать или выключать разделение по данному разделителю для всех (если условное разделение установлено для самого разделителя) или выбранных объектов (если условное разделение задано для объекта, входящего в состав разделителя);
- если разделитель условно выключен, то в качестве значения разделителя будет использоваться значение по умолчанию для типа разделителя.

ПРИМЕЧАНИЕ. Для каждого разделителя значение условного разделения не может прямо или косвенно зависеть от значения самого разделителя.

Возможно рекурсивное управление условным разделением, когда для разделителя, используемого для определения условного разделения другого разделителя, настроено условное разделение и т. д. В этом случае определение состояния условного разделения будет начинаться с такого разделителя, значение которого условно не разделяется (но оказывает влияние на другие объекты, участвующие в условном разделении) и далее по связанным разделителям до тех пор, пока не будет определено состояние условного разделения для разделителя, от состояния которого не зависит более ни один разделитель. Создать зацикленное условное разделение невозможно.

Условное разделение может включать или отключать разделение следующими способами:

- Для всех объектов, входящих в состав разделителя. В этом случае следует использовать свойство разделителя [Условное разделение](#).
- Для отдельных объектов, входящих в состав разделителя. Для этого следует воспользоваться колонкой [Условное разделение](#) в окне редактирования состава разделителя.
- Комбинацией описанных выше свойств.

При организации условного разделения следует принимать во внимание следующие особенности:

- В качестве объекта, хранящего значение условного разделения, может выступать:
 - Константа типа [Булево](#), не входящая в состав разделителя, для которого настраивается условное разделение.
 - Реквизит типа [Булево](#) у ссылочного объекта конфигурации, который:
 - Не входит в состав разделителя, для которого настраивается условное разделение.
 - Должен выступать в качестве типа другого разделителя, и разделитель такого типа должен быть единственным.
- При запуске системы выполняется начальная установка значений параметров сеанса, хранящих признак использования и значений разделителей, в соответствии с параметрами командной строки запуска или

значениями, указанными для пользователя информационной базы. Подробнее об установке значений разделителей при запуске см. [здесь](#).

- Если разделение общего реквизита условно выключено, то:
 - При чтении и записи данных текущие значения параметров сеанса, отвечающих за значение и использование разделителя, игнорируются. При этом используются следующие значения:
 - Признак использования – *Истина* (разделитель используется).
 - Значение разделителя – значение по умолчанию для типа разделителя.
 - При чтении и записи пользователей информационной базы текущие значения параметров сеанса, отвечающих за значение и использование разделителя, игнорируются. При этом используются следующие значения:
 - Признак использования – *Ложь* (разделитель не используется).
 - Значение разделителя – не используется.
- Если значение, определяющее условное разделение, не прочитано или неоднозначно, будет выдано исключение.
- Если разделитель, который используется в условном разделении другого разделителя, не используется, то другой разделитель также считается неиспользуемым.
- Если в сеансе выключены все разделители (с помощью условного разделения), то становятся доступными действия с информационной базой, которые недоступны, если используется хотя бы один разделитель.

Если настроено условное разделение **и** в свойстве общего реквизита (свойство общего реквизита [Условное разделение](#)), **и** с помощью колонки [Условное разделение](#) состава общего реквизита, то в качестве значения разделителя будет использовано одно из следующих значений:

- Текущее значение разделителя (установленное в соответствующий параметр сеанса) в том случае, если разделение условно включено **и** объектом (реквизит или константа), указанным в соответствующей колонке состава общего реквизита, **и** объектом (реквизит или константа), указанным в свойстве самого разделителя.
- Значение по умолчанию для типа разделителя, если условное разделение выключено любым способом.

Примечание. Условное разделение не работает в конфигураторе. Считается, что в сеансе конфигуратора реквизиты всегда условно **включены**.

25.2.5.2. Условное разделение, задаваемое в свойствах общего реквизита

25.2.5.2.1. С помощью константы

Разделитель можно полностью отключать с помощью константы. Константа должна иметь тип *Булево* и не должна разделяться самим реквизитом.

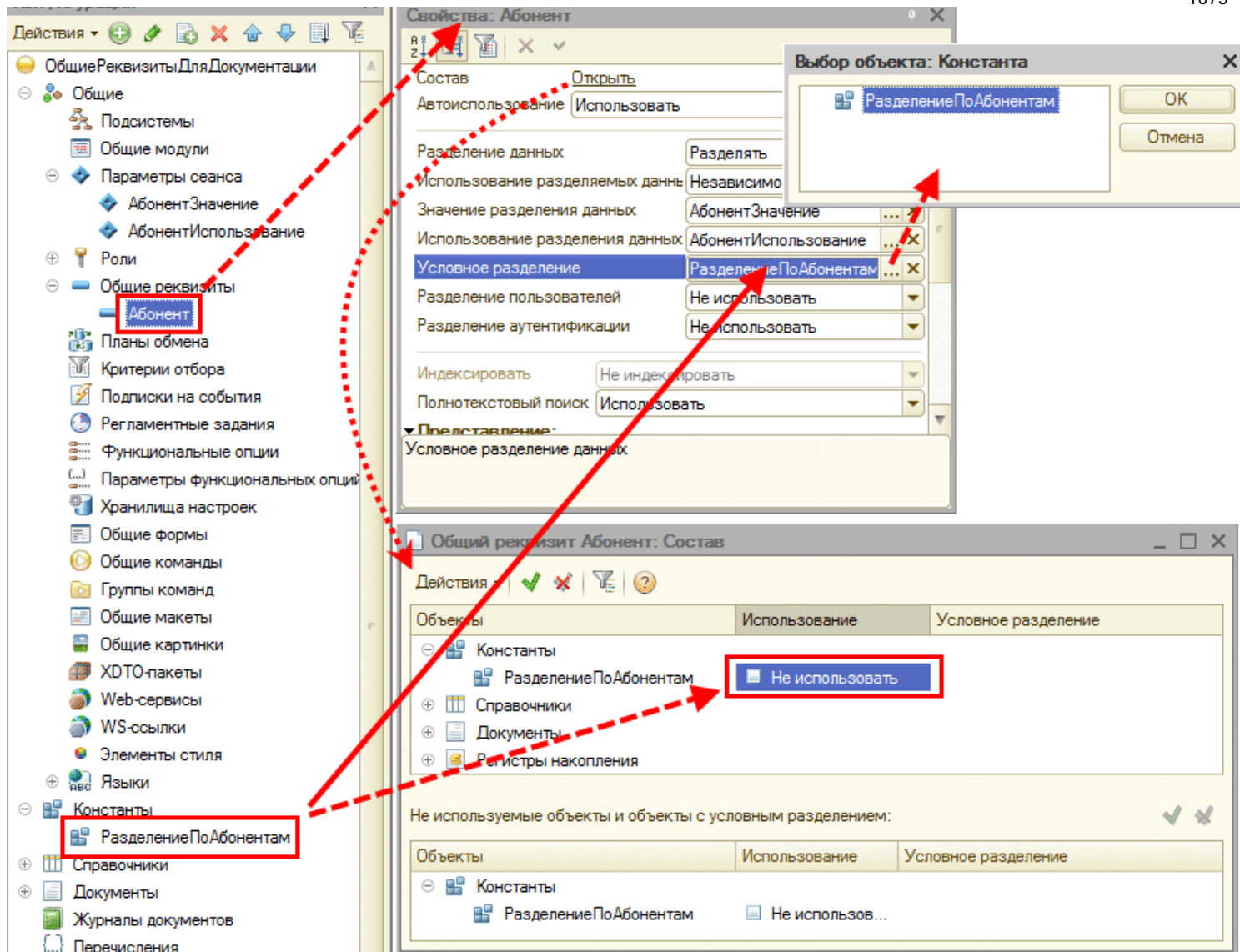


Рис. 352. Условное разделение с помощью константы

ВНИМАНИЕ! При первом запуске «1С:Предприятия» (после добавления условного разделения) значение константы будет равно *Ложь*, и разделение будет **выключено**.

Рассмотрим этот случай подробнее.

Имеется общий реквизит *Абонент* с типом *Число*. В состав этого реквизита входит справочник *Товары* со следующим содержанием:

Код	Наименование	Абонент
1	Босоножки	0
2	Ботинки	1
3	Валенки	2
4	Кроссовки	1
5	Сланцы	0

Для общего реквизита используется условное разделение, управляемое константой *РазделениеПоАбонентам*. Если значение константы равно *Ложь*, то это означает, что разделитель не работает (разделение **выключено**). Если значение константы равно *Истина*, то это означает, что разделитель работает (разделение **включено**).

Таким образом, можно выделить следующие ситуации:

- Значение разделителя *Абонент* равно *1*, разделитель используется, значение константы *РазделениеПоАбонентам* равно *Истина*. В данном сеансе **разделение** условно **включено** (для разделителя *Абонент*). В этой ситуации в справочнике доступны следующие элементы:

Код	Наименование	Абонент
2	Ботинки	1
4	Кроссовки	1

- Значение константы `РазделениеПоАбонентам` равно `Ложь`. В данном сеансе **разделение** условно **выключено** (для разделителя `Абонент`). В этой ситуации, независимо от значения параметров сеанса `АбонентИспользование` и `АбонентЗначение`, в справочнике доступны следующие элементы:

Код	Наименование	Абонент
1	Босоножки	0
5	Сланцы	0

25.2.5.2.2. С помощью реквизита объекта

Более сложный вариант управления условным разделением используется в том случае, если для условного разделения одного разделителя используются данные другого разделителя. Тогда в рамках разных областей данных возможно разное поведение разделителя, описывающего другую область данных.

Допустим, существует набор объектов конфигурации, описывающих нормативно-справочную информацию (далее НСИ): справочники `Банки`, `Валюты`, `Ставки НДС`. Необходимо реализовать следующую возможность:

- в информационной базе можно вести учет для разных групп компаний;
- каждая группа компаний состоит из организаций;
- содержимое объектов НСИ может быть как общим для всех организаций группы компаний, так и индивидуальным для каждой организации.

Реализация такой схемы потребует следующие объекты конфигурации:

- справочник `ГруппыКомпаний` с реквизитом `ЕдинаяНСИ` типа `Булево`;
- справочник `Организации`;
- справочник `Банки`;
- справочник `Валюты`;
- справочник `СтавкиНДС`;
- разделитель `ГруппаКомпаний` типа `СправочникСсылка.ГруппыКомпаний`;
- разделитель `Организация` типа `СправочникСсылка.Организации`. В состав этого разделителя входят только объекты, описывающие НСИ:
 - справочник `Банки`,
 - справочник `Валюты`,
 - справочник `СтавкиНДС`.

В состав разделителя `ГруппаКомпаний` входят все рассматриваемые справочники (кроме справочника `ГруппыКомпаний`), т. к. для каждой группы компаний нам нужен свой список организаций и содержимое НСИ.

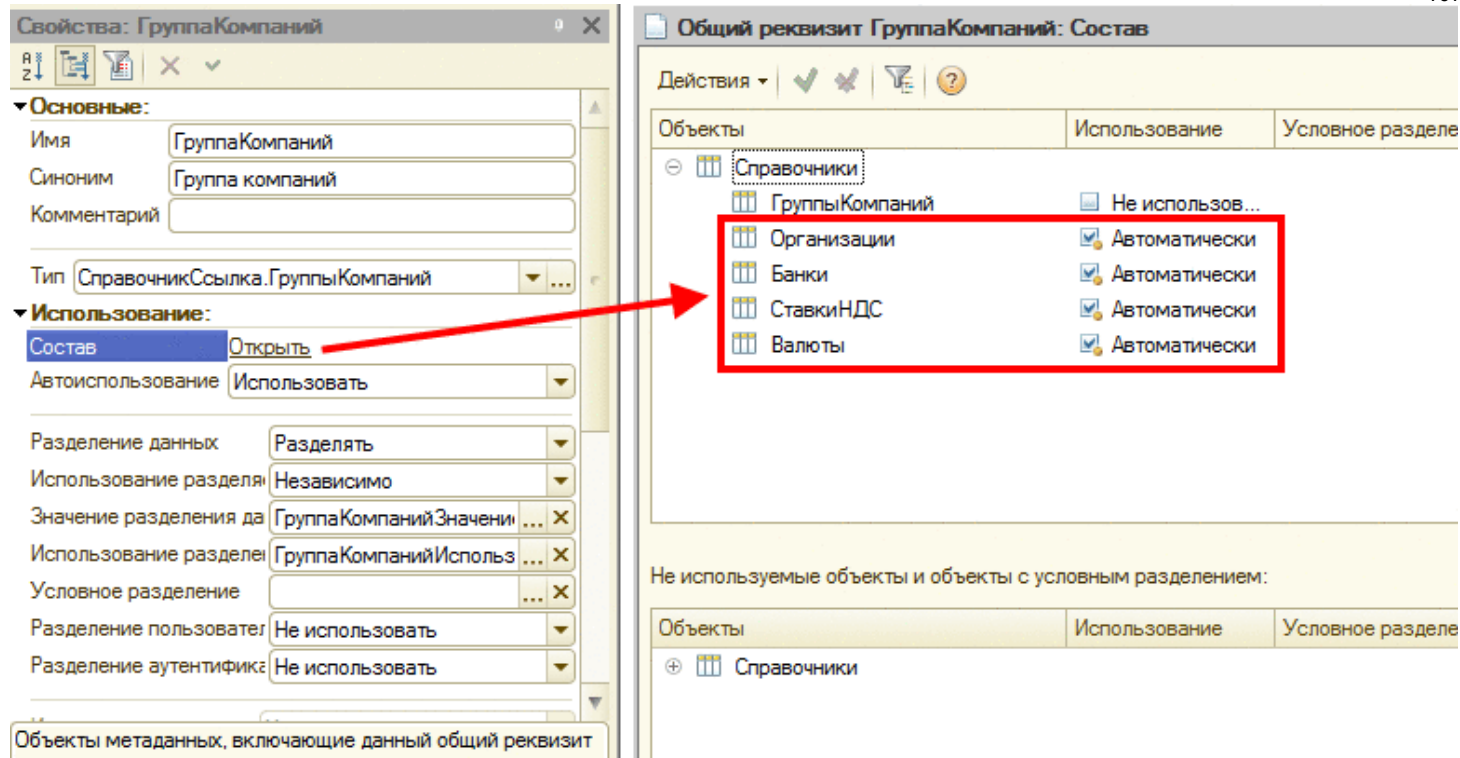


Рис. 353. Состав разделителя «ГруппаКомпаний»

В состав разделителя **Организация** входят только справочники **Банки**, **Валюты** и **СтавкиНДС**. Именно эти справочники являются НСИ, и именно для них нужно два варианта использования:

- для каждой организации выбранной группы компаний,
- для всех организаций выбранной группы компаний.

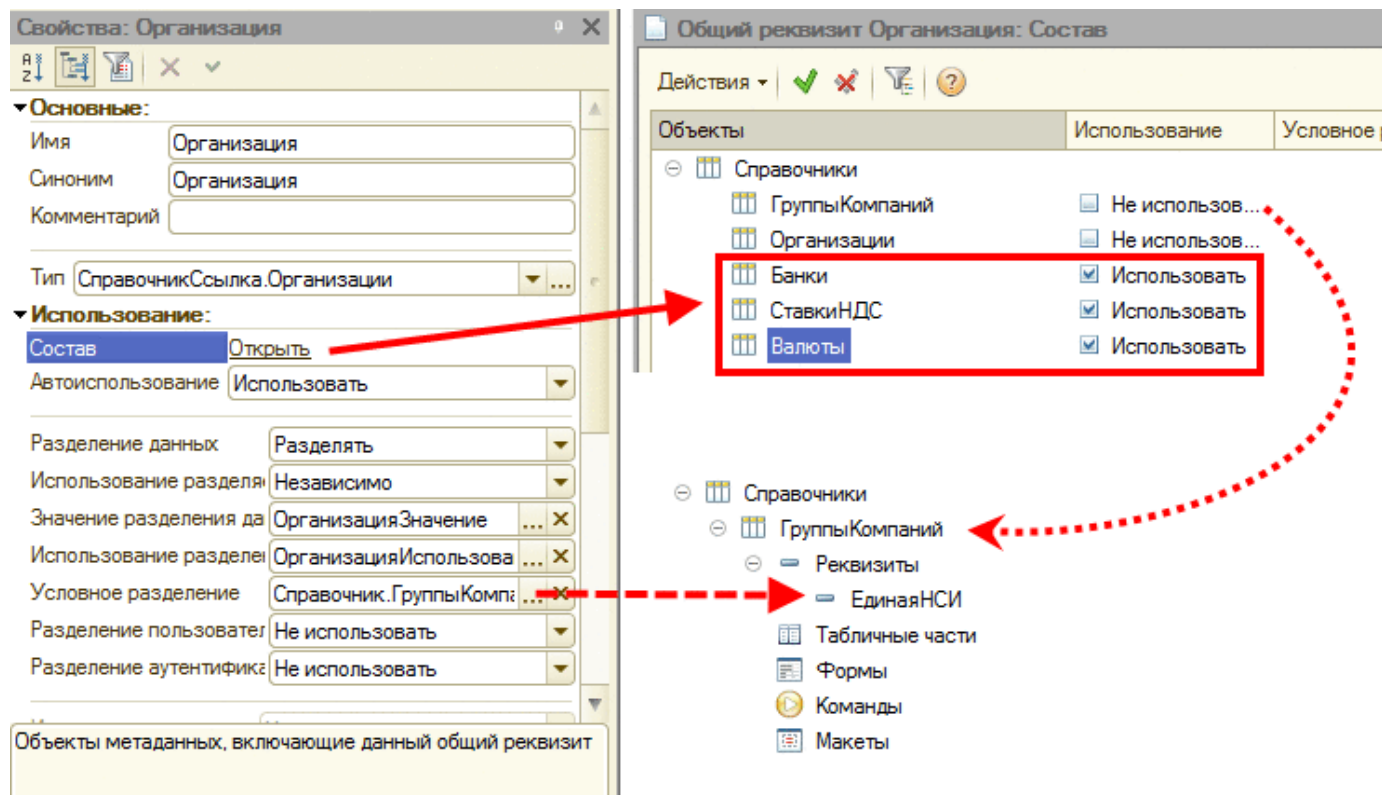


Рис. 354. Состав разделителя «Организация»

Справочник **ГруппыКомпаний** не входит в состав разделителя **Организация**, так как этого требуют правила организации условного разделения (см. [здесь](#)).

Теперь для каждого значения разделителя **ГруппаКомпаний** (элемент справочника **ГруппыКомпаний**) можно указать значение реквизита **ЕдинаяНСИ**:

- Реквизит установлен в значение *Ложь* – условное разделение **выключено**, и для любого значения разделителя *Организация* (в рамках одного значения разделителя *ГруппаКомпаний*) будет **единый** набор значений справочников *Банки*, *Валюты* и *СтавкиНДС*.

- Реквизит установлен в значение *Истина* – условное разделение **включено**, и для каждого значения разделителя *НСИ* (в рамках одного значения разделителя *ГруппаКомпаний*) будет **свой** набор значений справочников *Банки*, *Валюты* и *СтавкиНДС*.

Фактически, изменяя значение свойства *ЕдинаяНСИ*, происходит включение или выключение разделения по разделителю *Организация*.

Пусть в справочнике *ГруппыКомпаний* существуют следующие записи:

Наименование	ЕдинаяНСИ	Ссылка
Прогресс	<i>Ложь</i>	<i>ГК1</i>
Торговый дом	<i>Истина</i>	<i>ГК2</i>

В справочнике *Организации* существуют следующие записи (по две для каждого значения разделителя *ГруппаКомпаний*):

Наименование	ГруппаКомпаний	Ссылка
База комплектующих	<i>ГК1</i>	<i>ОР1</i>
Мир кондиционеров	<i>ГК1</i>	<i>ОР2</i>
СтройТоргВсе	<i>ГК2</i>	<i>ОР3</i>
Лайт	<i>ГК2</i>	<i>ОР4</i>

В справочнике *Банки* будет следующая информация:

Наименование	ГруппаКомпаний	Организация
АльфаБанк	<i>ГК1</i>	Пустая ссылка
СберБанк	<i>ГК1</i>	<i>ОР1</i>
ВТБ	<i>ГК1</i>	<i>ОР2</i>
Авангард	<i>ГК2</i>	<i>ОР1</i>
Уралсиб	<i>ГК2</i>	<i>ОР2</i>

Теперь рассмотрим на примере, какие данные будут доступны в каком случае.

Пример 1:

Реквизит *ЕдинаяНСИ* для элемента *Прогресс* установлен в значение *Ложь*.

Это значит, что разделение для справочника *Банки* **выключено**. Значит, содержимое этого справочника не зависит от значения разделителя *Организация* и в элемент справочника *Банки* (для разделителя *Организация*) в этом случае будет записываться значение, равное значению по умолчанию для типа разделителя. В нашем случае таким значением является пустая ссылка (так как тип разделителя *Организация* – ссылка на справочник). Поэтому если разделитель *ГруппаКомпаний* установлен в значение *ГК1*, то список справочника *Банки* будет показывать только одну запись: *АльфаБанк*.

Пример 2:

Реквизит *ЕдинаяНСИ* для элемента *Прогресс* установлен в значение *Истина*.

Это значит, что разделение для справочника *Банки* **включено**. Значит, содержимое этого справочника зависит **и** от значения разделителя *ГруппаКомпаний*, **и** от значения разделителя *Организация*.

Поэтому для группы компаний *Прогресс* и организации *Мир кондиционеров* в списке банков будет только банк *ВТБ*.

Меняя значение реквизита **ЕдинаяНСИ** (и значения разделителей) во время работы системы, можно получить тот или иной состав списка справочника **Банки** (в соответствии с правилами условного разделения).

25.2.5.3. Условное разделение, задаваемое для объекта (в свойстве «Состав»)

Условное разделение можно задавать для каждого объекта конфигурации, входящего в состав разделителя. В этом случае будет отключаться разделение не для всех объектов, входящих в состав разделителя, а только для тех объектов, для которых настроено условное разделение.

Условное разделение может управляться как константой, так и реквизитом объекта конфигурации (см. [здесь](#)).

Рассмотрим пример такого разделения.

Допустим, в одной информационной базе могут вести учет несколько абонентов, причем в рамках каждого абонента можно вести учет от лица нескольких дружественных организаций. При этом возможна ситуация, когда все дружественные организации оперируют единым списком складов, и возможна ситуация, когда для каждой организации можно сформировать свой список складов.

Реализация такой схемы потребует следующие объекты конфигурации:

- справочник **Абоненты** с реквизитом **РаздельныйСписокСкладов** типа **Булево**;
- справочник **Организации**;
- справочник **Склады**;
- разделитель **Абонент** типа **СправочникСсылка.Абоненты**;
- разделитель **Организация** типа **СправочникСсылка.Организации**.

Для организации рассмотренной схемы нам требуется выполнить следующее:

- из состава разделителя **Организация** нужно исключить справочник **Организации**;
- для справочника **Склады** (в составе разделителя **Организация**) нужно задать условное разделение через реквизит **РаздельныйСписокСкладов** справочника **Абоненты**.

Пусть в справочнике **Абоненты** есть две записи:

Наименование	РаздельныйСписокСкладов
Абонент 3453	Истина
Абонент 4617	Ложь

Если пользователь входит от имени абонента № 3453, то для каждой организации будет возможно вести индивидуальный список складов, т. к. реквизит **РаздельныйСписокСкладов** установлен в значение **Истина** (включено разделение справочника **Склады** в разделителе **Организация**).

В то же время при входе в систему от имени абонента № 4617 список складов будет един для всех организаций, реквизит **РаздельныйСписокСкладов** установлен в значение **Ложь** (выключено разделение справочника **Склады** в разделителе **Организация**).

25.2.5.4. Особенности условного разделения в распределенной системе

Использование условного разделения в распределенной системе имеет некоторые особенности. Эти особенности можно разделить на «логические» и «технические».

К «техническим» особенностям следует отнести невозможность произвольно изменять состояние объекта, управляющего условным разделением, в разделенном сеансе, если этот объект входит в состав разделенного плана обмена. Это связано с особенностями автоматической регистрации изменений объектов разделенными планами обмена (см. [здесь](#)). При автоматической регистрации изменений изменять объект, управляющий условным разделением, можно или в неразделенном сеансе, или в том случае, когда текущее состояние объекта равно **Ложь**. То есть можно только включить условное разделение, и нельзя его выключить (в автоматическом режиме).

«Логическая» особенность заключается в том, что изменение состояния условного разделения в главном узле распределенной информационной базы оказывает существенное влияние на состояние и поведение всей распределенной информационной системы. Такое изменение может затрагивать одну область данных, если условное разделение управляется реквизитом объекта, служащего типом другого разделителя (описание примера см. [здесь](#)). Также изменение состояния условного разделения может затрагивать всю информационную базу в том случае, если условное разделение служит для переключения между двумя режимами работы прикладного решения (описание примера см. [здесь](#)).

Так, в первом случае (пример с НСИ) при изменении условного разделения в периферийной базе у пользователей «пропадет» нормативно-справочная информация. Поэтому доступ к изменению реквизита, управляющего условным разделением, следует предоставлять только администратору области данных или человеку, обладающему достаточными полномочиями для принятия такого рода изменения.

Второй пример (изменение режима работы прикладного решения) будет оказывать влияние в целом на всю информационную базу. В периферийных базах пользователи могут «потерять» все свои данные (из-за смены значения разделителя) и т. д. Очевидно, что выполнять столь критическое изменение может только администратор всей информационной базы (или лицо, уполномоченное для таких действий), в то время как другие пользователи могут не иметь даже возможности чтения данного реквизита (или константы).

25.3. Определение возможности выполнения действия или набора данных

В том случае, когда в системе используется несколько разделителей, для определения доступности некоторого действия с тем или иным объектом, следует применять следующее правило:

- Для каждого разделителя, который разделяет требуемый объект, определяется возможность выполнения необходимого действия так, как будто разделитель в системе один (при этом, естественно, учитывается режим разделения – независимый или независимый и совместный).
- Результат по всем разделителям складывается «по И».

В дальнейших разделах будет рассматриваться выполнение того или иного действия с объектом для одного разделителя (если не будет явно описано обратное). Затем будет необходимо применить данное правило для получения итогового действия, если объект входит в состав нескольких разделителей.

Например, в системе есть два независимых разделителя: **Абонент** и **Организация**. Существуют два справочника: **Номенклатура** и **Контрагенты**. Каждый справочник разделен обоими разделителями.

Пример 1:

Необходимо определить возможность получения списка номенклатуры в случае, если разделитель **Абонент** используется, а разделитель **Организация** – нет.

Так как оба разделителя независимые, то для получения списка справочника необходимо, чтобы разделитель использовался. Следовательно:

- Для разделителя **Абонент** список номенклатуры получить можно, т. к. разделитель используется. Результат равен значению **Истина**.
- Для разделителя **Организация** список номенклатуры получить нельзя, т. к. разделитель не используется. Результат равен значению **Ложь**.
- Таким образом, в сеансе с такими установками разделителей, просмотр списка справочника будет (**Истина И Ложь = Ложь**) **невозможен**.

Пример 2:

Необходимо определить возможность получения списка контрагентов, если используются оба разделителя.

Так как оба разделителя независимые, то для получения списка справочника необходимо, чтобы разделитель использовался. Следовательно:

- Для разделителя **Абонент** список контрагентов получить можно, т. к. разделитель используется. Результат равен значению **Истина**.
- Для разделителя **Организация** список контрагентов получить можно, т. к. разделитель используется.

Результат равен значению *Истина*.

- Таким образом, в сеансе с такими установками разделителей просмотр списка справочника будет (*Истина И Истина = Истина*) **возможен**.

Пример 3:

Необходимо определить, какие значения для колонки *Пользователь* вернет метод *ПолучитьЗначенияОтбораЖурналаРегистрации()*. В системе создано два разделителя, при этом в текущем сеансе используется первый разделитель и не используется второй. Следовательно:

- для первого разделителя метод вернет только тех пользователей, которые зарегистрированы в журнале регистрации для значений разделителя, равному значению в текущем сеансе;
- для второго разделителя метод вернет всех пользователей (подробнее см. [здесь](#));
- таким образом, в сеансе с указанным состоянием разделителей метод вернет только тех пользователей, которые будут пересечением списков для первого и второго разделителей (операция **И**).

25.4. Установка значений разделителей при запуске

При запуске «1С:Предприятия» может возникнуть необходимость указать значение разделителя непосредственно в командной строке запуска (например, для проведения аутентификации, если выбрано разделение аутентификации).

Если в командной строке не указано одно или несколько значений разделителей, то недостающие разделители берутся из разделителей, указанных для текущего пользователя. Если один и тот же разделитель указан и у пользователя, и в командной строке и их значения не совпадают, то использование и значение разделителя будут установлены из командной строки, если пользователь информационной базы имеет право на установку параметров сеанса, хранящих использование и значение разделителя. Иначе аутентификация пользователя выполнена не будет.

При отладке клиента, запущенного из конфигуратора, можно задать разделители в диалоге настройки запуска в конфигураторе (см. [здесь](#)). При указании пользователя, от имени которого нужно выполнить запуск, доступен выбор пользователей. Список ограничивается установленными значениями разделителей, разделяющих аутентификацию.

В процессе аутентификации пользователя информационной базы значения параметров сеанса, на которые ссылаются свойства *Использование разделителя* и *Значение разделителя* объекта метаданных *Общий реквизит*, устанавливаются в значения, указанные для пользователя. К моменту исполнения любого кода на встроенном языке (в том числе и в модуле сеанса) значения этих параметров сеанса будут установлены, и при обращении к ним не будет вызываться обработчик события *УстановкаПараметровСеанса*.

Следует помнить, что определение значения разделителя по его переданному идентификатору выполняется без учета использования других разделителей. То есть определение значения разделителя не зависит от порядка установки значений (и признака использования) других разделителей. Если система не может однозначно определить значение разделителя по его идентификатору (например, существуют два элемента справочника с одинаковым кодом), то будет вызвано исключение.

Если использование и значение разделителя (см. далее), указанные в строке запуска, отличаются от значений, указанных для пользователя, то:

- если пользователь имеет права на изменение параметров сеанса, хранящих значение и признак использования разделителя, то выполняется установка этих значений из командной строки запуска;
- если такие права отсутствуют, аутентификация завершается с ошибкой.

25.4.1. Командная строка клиентского приложения

Для указания значений разделителей используется параметр командной строки **Z**.

Синтаксис параметра:

```
/Z "[<Признак>][<Значение>][, [<Признак>][<Значение>]]"
```

Описание:

Каждая пара <Признак><Значение> отвечает за установку одного разделителя. Пропуск разделителей невозможен. Порядок следования пар <Признак><Значение> соответствует порядку расположения разделителей в окне конфигурации. Если порядок общих реквизитов меняется (например, выполнена сортировка списка общих реквизитов), необходимо изменить расположение значений в параметре **Z**. Пары <Признак><Значение> разделяются запятыми (",").

Параметры:

<Признак>

Признак использования данного разделителя. Может принимать значения «+» (значение по умолчанию) и «-». Если перед значением разделителя указан признак «+» (или не указано ничего), то указанное значение будет записано в параметр сеанса, хранящий значение общего реквизита, и в параметр сеанса, хранящий использование общего реквизита, будет установлено значение **Истина**.

Если перед значением разделителя указан признак «-», то это означает, что разделитель не используется в данном сеансе (в параметр сеанса, отвечающий за использование разделителя, будет установлено значение **Ложь**). Но если значение указано, оно будет записано в параметр сеанса, хранящий значение общего реквизита.

<Значение>

Указывает значение, которое будет установлено в параметр сеанса, хранящий значение соответствующего разделителя. Значение разделителя задается в текстовом виде и имеет следующий формат (в зависимости от типа разделителя):

- Для типа **Булево** указываются значения **0** (**Ложь**) или **1** (**Истина**).
- Для типа **Число** указывается число в каноническом виде.
- Для типа **Дата** указывается дата в формате **ггггммддччиисс**, где:
 - **гггг** – год (4 цифры);
 - **мм** – номер месяца (2 цифры);
 - **дд** – номер дня (2 цифры);
 - **чч** – номер часа (2 цифры);
 - **ии** – номер минуты (2 цифры);
 - **сс** – секунды (2 цифры).

При задании параметра необходимо указывать все составляющие даты, т. е. параметр состоит из 14 цифр.

- Для типа **Строка** указывается собственно текст.
- Для ссылочных типов указывается текстовое представление стандартного реквизита **Код** или **Номер**. Коды и номера должны быть уникальны среди всех значений соответствующего объекта конфигурации. Не допускается использование справочников с длиной кода, равной 0, и документов с нумераторами.

Если в начале значения параметра должен быть символ «+» или «-», он удваивается. Если в текстовой строке, описывающей значение разделителя, должен быть символ «,», то этот символ удваивается.

Если значение, указанное в параметре командной строки, может соответствовать одновременно значениям разных типов (например, значение «1» может являться значением **Истина**, числом «1», кодом справочника «1» и номером документа «1»), правило конвертации определяется исходя из типа соответствующего разделителя.

Если в командной строке указан только признак («+» или «-»), но не указано значение, то в качестве значения разделителя используется значение по умолчанию соответствующего типа.

Пример:

```
/Z "-0,+001,+,---3"
```

В данном примере:

- первый разделитель не используется, но указано его значение: «0»;
- второй разделитель используется, и его значение равно «001»;
- третий разделитель используется, и его значением будет значение по умолчанию для типа, указанного в качестве типа общего реквизита;
- четвертый разделитель не используется, но его значением будет «-3».

25.4.2. Веб-клиент или тонкий клиент, подключенный через веб-сервер

Указание значений разделителей при запуске веб-клиента (или тонкого клиента, подключенного через веб-сервер) возможно двумя способами: в адресной (командной) строке запуска с помощью параметра **Z** и в файле `default.vrd`.

25.4.2.1. С помощью параметра Z

Значения разделителей могут задаваться в адресной строке с помощью параметра **Z**. Формат параметра для веб-клиента аналогичен командной строке запуска клиентского приложения.

Пример:

[Копировать в буфер обмена](#)

```
http://localhost/infobase?Z="-ПервыйРазделитель , +"
```

Символы, недопустимые в URL ([RFC 1738](http://tools.ietf.org/html/rfc1738), <http://tools.ietf.org/html/rfc1738.html>), преобразуются в кодировку UTF-8 и кодируются в соответствии с разделом [2.2. URL Character Encoding Issues](#) стандарта [RFC 1738](#) (с помощью символа «%» и двух шестнадцатеричных символов).

25.4.2.2. В файле default.vrd

Существует возможность настроить доступ в информационную базу таким образом, что конкретные значения разделителей будут задаваться адресом информационной базы, а не с помощью параметра командной строки **Z** или строки соединения с информационной базой. Для этого следует настроить соответствующим образом элемент `<zones>` файла `default.vrd`.

Также существует возможность запретить изменение значений объектов, связанных с механизмом разделения данных, в том случае, если доступ к информационной базе осуществляется с помощью веб-клиента или тонкого клиента, подключенного через веб-сервер (безопасный сеанс). Такую возможность нужно использовать, если нужна гарантия, что при доступе к информационной базе через Интернет будет невозможно получить доступ к другой области данных.

Подробнее о параметрах настройки файла `default.vrd` можно посмотреть в книге «1С:Предприятие 8.3. Руководство администратора».

25.4.3. Внешнее соединение и строка соединения

В строке соединения и при запуске внешнего соединения используется параметр **Zn**. Формат параметра аналогичен командной строке запуска клиентского приложения.

Пример:

[Копировать в буфер обмена](#)

```
Srvr=localhost ; Ref=Demo82Srv ; Zn="-ПервыйРазделитель , + , ---3"
```

Также этот параметр можно указывать в файле `default.vrd` (в строке соединения с информационной базой) при публикации на веб-сервере.

В файле `default.vrd` возможен еще один способ задания значений разделителей, подробнее о котором см. [здесь](#).

25.5. Безопасный режим разделения данных

При использовании различных регламентных операций, выполняемых в рамках сеансов с неиспользуемыми значениями разделителей, может исполняться программный код прикладного решения, «надежность» которого, в общем случае, проверить не представляется возможным. Под «надежностью» понимается работа строго в рамках поставленных ограничений, без злоупотреблений. Например, вызываемый код должен работать в рамках только тех значений разделителей, которые ему установила вызывающая сторона, без попыток перехода в

другие области данных. Такая попытка может привести к несанкционированному доступу к данным «чужой» области данных.

Для предотвращения попыток смены значений разделителей (и, как следствие, области данных, в которой работа программный код) существует возможность установить безопасный режим разделения данных. Для этого используется метод `УстановитьБезопасныйРежимРазделенияДанных()`. Использование данного метода для разделителя аналогично указанию для данного разделителя значения атрибута `safe="true"` для элемента `zone` файла `default.vrd`. Описание данного файла находится в книге «1С:Предприятие 8.3. Руководство администратора». Если безопасный режим разделения данных включен с помощью файла `default.vrd`, то отключить его с помощью метода `УстановитьБезопасныйРежимРазделенияДанных()` невозможно.

В безопасном режиме разделения данных запрещено:

- Выключать использование разделителя, если разделение не является условно выключенным.
- Изменять значение используемого разделителя, если разделение не является условно выключенным.
- Изменять объекты, которые управляют условным разделением:
 - Указанные для самого разделителя;
 - Указанные для объектов, входящих в состав разделителя.

Безопасный режим разделения данных можно устанавливать по каждому разделителю отдельно. Количество включений безопасного режима разделения данных должно совпадать с количеством выключений. Однако если внутри процедуры или функции происходило включение безопасного режима разделения данных (один раз или более), но не происходило его выключение, то система автоматически выполнит выключение столько раз, сколько незавершенных включений было в покидаемой процедуре или функции.

Если в процедуре или функции безопасный режим разделения данных для выбранного разделителя выключался большее количество раз, чем включался, то будет вызвано исключение.

Применение этого механизма может выглядеть следующим образом (в системе существует независимый разделитель `Абонент`):

[Копировать в буфер обмена](#)

```
Для Каждого ЗначениеРазделителя Из СписокЗначенийРазделителей Цикл
    // "переходим" в нужную область
    УстановитьЗначениеРазделителя(ЗначениеРазделителя);
    // установим безопасный режим разделения данных
    УстановитьБезопасныйРежимРазделенияДанных("Абонент", Истина);
    // вызвать нужный программный код
    ОбработатьДанныеОбласти();
    // отключить безопасный режим разделения данных
    УстановитьБезопасныйРежимРазделенияДанных("Абонент", Ложь);
КонецЦикла
```

Проверить состояние безопасного режима разделения данных можно с помощью метода `БезопасныйРежимРазделенияДанных()`.

25.6. Удаление области данных

В процессе эксплуатации системы может возникнуть ситуация, когда область данных становится невостребованной. Например, абонент перестает пользоваться услугами сервиса (каждый абонент работает в одной области данных), или происходит реорганизация группы компаний (каждая компания в группе работает в своей области).

Также возможна ситуация, когда из системы удаляются разделители. Но если в состав разделителей входят объекты конфигурации с предопределенными данными, то удалить разделитель не получится до тех пор, пока не будут удалены предопределенные данные (см. [здесь](#)).

Удаление области данных осуществляется с помощью метода глобального контекста `УдалитьДанныеИнформационнойБазы()` или с помощью параметра командной строки `/EraseData` пакетного режима запуска Конфигуратора. Для метода встроенного языка удаляемая область определяется значениями параметров сеанса, определяющих значения разделителей в сеансе, откуда происходит вызов метода:

- Если в текущем сеансе не используется ни один разделитель, то удаляются **все данные информационной базы**. После удаления состояние информационной становится таким же, как после создания новой, пустой,

информационной базы и загрузки в нее конфигурации.

- Если в текущем сеансе используется хотя бы один разделитель, то удаляются **полностью разделенные данные** текущей области данных. Иначе говоря, выполняется удаление только тех данных текущей области, которые разделены всеми разделителями, которые используются в текущем сеансе. Удаление данных области приводит область в такое же состояние, как после первой установки текущего набора значений используемых разделителей до начала выполнения операций с данными.

СОВЕТ. Сразу после выполнения метода рекомендуется сменить значения разделителей, определяющих область данных.

Для вызова метода требуется наличие у пользователя права [Администрирование](#) и возможности получения монопольного доступа к информационной базе. Монопольный доступ необходимо установить с помощью явного вызова метода глобального контекста [УстановитьМонопольныйРежим\(\)](#).

При удалении области данных с помощью пакетного режима запуска Конфигуратора, удаляемая область определяется с помощью параметра `/z` командной строки запуска (см. [здесь](#)). Если заданы не все разделители, то область определяется аналогично вызову метода [УдалитьДанныеИнформационнойБазы\(\)](#) (см. выше). Система пытается автоматически получить монопольный доступ к информационной базе. В случае ошибки получения монопольного доступа, выполнение пакетного режима запуска Конфигуратора завершается с ошибкой.

Во время удаления будут удалены следующие данные:

- Из таблиц, связанных с объектами конфигурации, входящих в состав разделителей (включая предопределенные данные);
- Хранилищ настроек;
- Истории работы пользователя;
- Списка пользователей информационной базы;
- Часовой пояс информационной базы – если в конфигурации не определено независимых разделителей, то значение это параметра становится неопределенным, в противном случае – значение удаляется.
- Если в сеансе не используется ни один разделитель, то следующие настройки информационной базы заполняются значениями по умолчанию:
 - Время ожидания блокировки данных;
 - Минимальная длина паролей пользователей;
 - Проверка сложности паролей пользователей;
 - Разрешение полнотекстового поиска.
- При установленном режиме совместимости с версией 8.3.2 или младше, предопределенные данные получают значения, определенные в конфигураторе или значения по умолчанию. При отключенном режиме совместимости предопределенные данные будут созданы при первом обращении к ним;
- Значения реквизитов элемента [ЭтотУзел](#) планов обмена будут установлены в значения по умолчанию;
- Значения неразделенных констант получают значения по умолчанию, а значения разделенных констант удаляются.

25.7. Особенности поведения объектов системы

Особенности поведения объектов системы в случае использования разделителей см. [здесь](#).

Глава 26. Разработка решений для мобильной платформы

26.1. Общая информация

Система программ «1С:Предприятие» позволяет разрабатывать приложения, функционирующие на мобильных устройствах (смартфоны, планшетные компьютеры). **Мобильное приложение** – прикладное решение, предназначенное для выполнения на мобильной платформе «1С:Предприятия». **Мобильная платформа** – это специальная версия «1С:Предприятия», предназначенная для исполнения мобильных приложений на мобильных устройствах, которые функционируют под управлением операционных систем iOS (<http://www.apple.com/ios/>) и Android (<http://www.android.com/>). Мобильная платформа реализована в архитектуре тонкого клиента, работающего с файловым вариантом информационной базы, расположенной на мобильном устройстве. Для функционирования мобильного приложения не требуется наличие постоянного канала связи с каким-либо компонентом внешней сетевой инфраструктуры. В случае необходимости, можно реализовать внешнее взаимодействие с помощью различных механизмов мобильной платформы.

Разработку мобильного приложения можно условно разбить на следующие этапы:

- Принимается решение, каким будет мобильное приложение:
 - Независимо от каких-либо прикладных решений на удаленной системе. В этом случае структура данных мобильного приложения определяется только областью применения приложения. Скорее всего, никакой обмен данными не потребуется.
 - Автономное рабочее место некоторого прикладного решения, расположенного на удаленной системе. В этом случае структура данных в основном определяется прикладным решением, но будут изменения, определяемые как областью применения, так и особенностями мобильного приложения. Протокол обмена данными, скорее всего, будет представлять собой специализированный протокол обмена конкретного прикладного решения с конкретным мобильным приложением. Протокол обмена не ориентирован на обмен с другими решениями.
 - Универсальное автономное рабочее место. В этом случае структура данных должна учитывать специфику всех прикладных решений, с которыми должен взаимодействовать мобильное приложение. Также необходимо учитывать особенности мобильного приложения. Протокол обмена должен быть достаточно универсальным и стандартизированным.
- Уточняется, мобильное решение должно работать только на мобильном устройстве или оно также может работать и на персональном компьютере (например, на ноутбуке).
- Выполняется реализация мобильного приложения и его отладка на персональном компьютере (включая механизмы обмена данными). При этом необходимо понимать, что мобильная платформа имеет ряд отличий и особенностей, в силу чего невозможно полностью проверить работоспособность мобильного приложения на персональном компьютере. Однако код на встроенном языке можно отлаживать только на персональном компьютере.
- На мобильном устройстве (одном или нескольких) выполняется окончательная проверка и отладка интерфейса и функциональности приложения.

СОВЕТ. Рекомендуется выбирать для разработки вариант автономного рабочего места, которое может функционировать и на мобильном устройстве и на персональном компьютере.

При разработке механизма обмена данными необходимо учитывать, что обмениваться данными должны приложения с разными версиями интерфейса обмена. Это не должно вызывать проблем при обмене. Связано это с тем, что обновление приложения сразу для всей инфраструктуры практически невозможно. Также следует отметить, что синхронизация данных по устаревшей версии протокола обмена может использоваться для выполнения операций резервного копирования перед обновлением конфигурации мобильного приложения.

При разработке мобильного приложения необходимо учитывать ограничения, которые накладывает мобильная платформа по сравнению с платформой «1С:Предприятие» для персонального компьютера:

- Ограниченный перечень доступных объектов конфигурации и механизмов;
- Ограниченный набор свойств реквизитов;
- Ограниченный набор элементов управляемых форм;
- Упрощенная реализация некоторых механизмов (например, права доступа или начальная страница);
- Отсутствие обычных форм и обычного режима запуска;
- Отсутствие средств отладки мобильного приложения;
- Полный список особенностей см. [здесь](#).

Также следует учитывать, что мобильная платформа и платформа для персонального компьютера, при совпадении версий различаются по предоставляемой функциональности. Ниже приведено функциональное соответствие версий платформы для мобильного устройства и персонального компьютера.

Мобильная платформа	Платформа для персонального компьютера
8.3.5	8.3.4
8.3.4	8.3.2
8.3.3	8.3.2

В документации, которая разрабатывается для мобильного приложения, прикладной разработчик может не описывать поведение стандартных механизмов системы, а ссылаться на «Руководство пользователя мобильного приложения», которое расположено по адресу http://v8.1c.ru/mobile_user_guide/.

26.2. Интерфейс приложения

Интерфейс приложения является аналогом интерфейса **Такси**, с возможностью переключения между окнами с помощью главного меню приложения. Навигация в приложении зависит от того, какая форма активна в данный момент:

- Форма, для которой не установлен владелец и не сохраняющая данные.
 - Например: форма списка, форма обработки.
- Форма открывается командой из меню функций – в левой части заголовка окна находится кнопка открытия главного меню приложения. Главное меню приложения доступно с помощью жеста с левой стороны экрана устройства или специальной кнопки.
- Форма открывается другими командами или из встроенного языка – в левой части заголовка окна находится кнопка возврата в предыдущее окно (<). При закрытии такой формы будет активизирована та форма, которая была активной в тот момент, когда была вызвана команда открытия закрываемой формы. Главное меню приложения доступно с помощью жеста с левой стороны экрана устройства.
- Форма с установленным владельцем.
 - Например: форма выбора, форма объекта, открытая из списка.
 - При последовательно открытии таких окон образуется цепочка открытых окон, в которой пользователю доступна только самая последняя открытая форма. При этом закрытие текущей формы приводит к активизации предыдущей формы цепочки, т. е. той формы, откуда была открыта закрываемая форма. Следует понимать, что возврат может произойти не в ту форму, в которой пользователь находился до того, как переключился на закрываемую форму.
 - В левой части заголовка окна находится кнопка возврата в предыдущее окно (<). При этом текущее окно будет закрыто.
 - Главное меню приложения доступно с помощью жеста с левой стороны экрана устройства.
- Форма, для которой не установлен владелец и сохраняющая данные.
 - Например: форма создания объекта, вызванная из главного меню приложения или с помощью встроенного языка.
 - При закрытии такой формы будет активизирована та форма, которая была активной в тот момент, когда была вызвана команда открытия закрываемой формы.
 - В левой части заголовка окна находится кнопка возврата в предыдущее окно (<). При этом текущее окно будет закрыто.
 - Главное меню приложения доступно с помощью жеста с левой стороны экрана устройства.
- Формы, блокирующие интерфейс.
 - Например: форма, для которой свойство **Режим открытия окна** установлено в значение **Блокировать окно владельца** или **Блокировать весь интерфейс**.
 - В заголовке окна находится кнопка, связанная с командой формы:
 - При наличии сохраняемых данных (признак реквизита **Сохраняемые данные**) – стандартная кнопка **Отмена**.
 - При отсутствии сохраняемых данных – стандартная кнопка **Закреть**.
 - Если форма содержит команду **Отмена**, **Закреть**, **Нет** – отображается эта команда.
 - Главное меню приложения недоступно.

При запуске мобильного приложения отображается полноэкранный картинка, заданная разработчиком приложения. В определенный момент времени в нижней части экрана появляется логотип и копирайт фирмы «1С», перекрывающая часть заставки:

- В портретной ориентации – примерно 10% площади экрана.
- В ландшафтной ориентации – примерно 15% площади экрана.

26.3. Особенности использования

26.3.1. Глобальный командный интерфейс

Интерфейс мобильной платформы ориентирован на то, что в один момент времени на экране отображается одна форма. Под нее освобождается максимум свободного места на экране. Поэтому, команды, которые на персональном компьютере формируют командный интерфейс основного раздела (панели навигации и действий), собраны в главное меню приложения. Также это меню содержит команду возврата на начальную страницу, список разделов (команды каждого раздела отображаются в виде подменю), а также команды отображения информации о программе и перехода к списку приложений. Вызов главного меню приложения осуществляется жестом с левой стороны экрана (если доступно) или с помощью специальной кнопки **≡**, расположенной в левой части заголовка окна.

СОВЕТ. При разработке командного интерфейса рекомендуется ориентироваться на возможности мобильной платформы.

Существует возможность настраивать ролевую видимость командного интерфейса (см. [здесь](#)).

26.3.2. Начальная страница

На начальной странице поддерживается отображение только одной формы. Выбор этой формы выполняется по следующим правилам:

- Из списка форм, указанных в настройках рабочей области начальной страницы исключаются формы, которые недоступны в соответствии с правами доступа;
- Затем из оставшегося списка исключаются формы, которые отключены функциональными опциями;
- Из оставшихся форм выбирается первая видимая форма, при этом обход форм рабочего стола идет сверху вниз, вначале левая колонка, затем правая (если задано расположение форм начальной страницы в две колонки).

Кроме формы, на начальной странице может располагаться командный интерфейс основного раздела. Состав командного интерфейса основного раздела на мобильной платформе аналогичен таковому на платформе для персонального компьютера (кроме группы **См. также**). Он отображается в виде матрицы кнопок (каждая кнопка связана с одной командой), сгруппированной по страницам. Картинка на кнопке получается из связанной

команды. Если команда не обладает настраиваемой картинкой, используется стандартная картинка для данной группы команд.

Команды выводятся так чтобы максимально заполнить рабочую область окна. Группы команд размещаются на страницах так, чтобы избежать переноса между страницами, т. е. если после одной группы команд вторая не помещается полностью в оставшееся место, то она начинается с новой страницы. Если все команды не помещаются на экран, то они размещаются на нескольких страницах, которые можно пролистывать.

26.3.3. Сообщения пользователю

Отображение сообщений пользователю зависит от того, сколько не просмотренных сообщений имеется в настоящий момент. Первое не просмотренное сообщение отображается в виде диалога с текстом сообщения и кнопкой **OK**. После прихода второго сообщения в диалоге дополнительно появляется кнопка **Все сообщения**. В кнопке **Все сообщения** имеется подпись с полным количеством пришедших сообщений.

По нажатию на кнопку **Все сообщения** открывается панель сообщений, которая содержит все непрочитанные сообщения. Если в диалоге нажата кнопка **OK** или закрывается панель сообщений, то список сообщений очищается. Нажатие на сообщение приводит к активации элемента формы, если сообщение связано с таковым. Для повторного открытия списка сообщений следует повторно вызвать действие, приводящее к открытию списка.

26.3.4. Форма

26.3.4.1. Общая информация

Для тех механизмов или объектов конфигурации, которые не поддерживаются мобильной платформой, будут недоступны свойства элементов формы, связанные с этим механизмом или объектом. Например, свойство **Сочетание клавиш** или события, связанные с перетаскиванием.

Принципы и правила формирования формы для мобильной платформы в основном совпадают с таковыми для тонкого клиента, однако, есть некоторые особенности, которые следует учитывать при разработке форм мобильного приложения:

- Экраны смартфонов, в основном, ориентированы на вертикальное расположение (портретная ориентация), поэтому форма существенно ограничена по своей ширине и слабо ограничена по высоте.
- Для мобильных устройств не принята прокрутка формы по горизонтали, т. к. полосы прокрутки на мобильных устройствах не отображаются постоянно, поэтому пользователь может не обратить внимания на то, что форма (отчет или таблица) показывают только фрагмент информации.

26.3.4.2. Доступные элементы формы

В форме мобильного приложения можно использовать следующие элементы формы:

- Поля следующих видов:
 - Поле ввода;
 - Поле надписи;
 - Поле переключателя;
 - Поле картинки;
 - Поле флажка;
 - Поле индикатора;
 - Поле полосы регулирования;
 - Поле диаграммы;
 - Поле табличного документа.
- Кнопка;
- Таблица;
- Декорация: надпись и картинка;
- Группы всех видов.

26.3.4.3. Расположение элементов формы

Группы, включая саму форму, и большая часть элементов формы, отображаются в виде списков, как это принято в мобильных интерфейсах. Элементы формы условно делятся на три типа:

1. Обязательно располагаются в строке списка. В эту группу входят следующие элементы:
 1. Поле ввода;
 2. Поле надписи;
 3. Поле переключателя;
 4. Кнопка;
 5. Поле диаграммы.
2. Могут располагаться в строке списка. В группу входит декорация.
3. Не могут располагаться в строке списка. В эту группу входят все остальные элементы.

Под фразой «отображение списком» понимается следующее:

- Группа имеет фон, отличающийся от цвета формы.
- Строкой списка выступает либо элемент формы, либо группа с горизонтальной группировкой элементов.
- Элементы, расположенные в группе с горизонтальной группировкой, располагаются в одной строке.
- Элементы, расположенные в группе с вертикальной группировкой, располагаются каждый на своей строке.
- Если в группе с вертикальной группировкой находятся элементы, которые не могут располагаться в строке списка, то в списке отображается разрыв.
- Если группа обладает отображаемым заголовком, то перед группой формируется отступ.
- Визуальными разделителями строк являются горизонтальные линии, отображением которых нельзя управлять.

Особенностью поведения элементов 2 типа является то, что они располагаются в строке списка, только если сгруппированы с элементами 1 типа. Сами по себе элементы 2 типа в список не оформляются.

Многие элементы формы не имеют рамок (поля ввода и кнопки, у которых цвет рамки установлен в значение [Авто](#)). В связи с этим, выравнивание полей ввода на форме (как на платформе для персонального компьютера) не имеет практического смысла. Вместо него используется алгоритм выравнивания значений в полях ввода. В зависимости от наличия и расположения заголовка у поля ввода, информация в нем выравнивается по-разному:

- Заголовок поля отсутствует или отображается не слева, то числовые значения выравниваются вправо, а остальные – влево.
- Заголовок поля находится слева:
 - В полях с редактированием текста – текст выравнивается влево.
 - Числовые значения выравниваются вправо.
 - Выравнивание полей остальных типов зависит от того, есть в группе с этими полями поле для редактирования текста или нет. Если такое поле есть, то выравнивание остальных полей осуществляется аналогично выравниванию этого поля. В противном случае – вправо.

Заголовок формы отображается только в заголовке окна и не дублируется на самой форме.

26.3.4.4. Особенности отображения элементов формы

26.3.4.4.1. Общие особенности

На форме не отображается отметка незаполненного как для элементов, расположенных на форме, так и для элементов, расположенных в таблице.

Заголовки полей выводятся без символов ":" в конце текста.

26.3.4.4.2. Форма

Свойство [ТекущийЭлемент](#) позволяет получить и установить текущий элемент формы. Однако установка текущего элемента формы не приводит к переходу этого элемента в режим редактирования и никак не отображается визуально.

26.3.4.4.3. Поле ввода

Для облегчения работы с полем ввода с помощью сенсорных экранов мобильных устройств, во всех полях, кроме полей ввода текста и полей, где используется автоподбор, редактирование значения непосредственно в поле ввода не предполагается. Поле ввода ведет себя как кнопка, заголовок которой отображает представление текущего значения, а нажатие этой кнопки открывает специализированный редактор. В том случае, если в поле ввода используется автоподбор, имеется возможность очистить значение с помощью кнопки [Очистить](#), расположенной в левой части заголовка окна.

Длинное нажатие в поле ввода приводит к появлению контекстного меню, которое состоит из стандартных команд системы и команд, которые добавлены в это меню во время разработки.

Для отображения кнопок очистки, открытия и регулирования, следует установить в значение [Да](#) соответствующее свойство поле ввода. Кнопка выбора всегда не отображается, при этом событие [НачалоВыбора](#) срабатывает в момент нажатия на само поле ввода. Если для редактирования значения поля используется отдельная форма, то рядом с полем размещается специальный маркер (>), нажатие на который приведет к открытию этой формы. Расположение и внешний вид редакторов отличается на планшете и телефоне.

Многострочное поле ввода всегда работает в режиме ввода текста. Такое поле ввода не поддерживает выбор значений и автоподбор. Если поле ввода связано с данными, отличными от типа [Строка](#), то для такого поля ввода игнорируется значение флажка [МногострочныйРежим](#). Многострочное поле ввода может отображаться тремя различными способами:

1. Автоматическая подстройка высоты поля под высоту текст (чтобы избежать прокрутки). Происходит когда поле занимает всю ширину формы, не указана высота поля (свойство [Высота](#)) и поле может растягиваться по вертикали (свойство [РастягиватьПоВертикали](#)).
2. Поле имеет заданную высоту и допустима вертикальная прокрутка внутри поля. Происходит когда указана высота поля (свойство [Высота](#)) и поле не может растягиваться по вертикали (свойство [РастягиватьПоВертикали](#)).
3. Поле занимает максимально возможное место на форме. Происходит когда указана высота поля (свойство [Высота](#)) и поле может растягиваться по вертикали (свойство [РастягиватьПоВертикали](#)).

Значение свойства [Подсказка ввода](#) не отображается в многострочном поле ввода.

26.3.4.4.4. Поле флажка

Флажок может отображать только два состояния. При работе под управлением ОС iOS флажок всегда отображается в виде тумблера.

26.3.4.4.5. Поле переключателя

Поле переключателя вида [Переключатель](#) всегда выводится в отдельной вертикальной группе (количество колонок не настраивается). Поле

переключателя вида [Переключатель](#) не комбинируется с другими элементами в одной строке.

Поле переключателя вида [Тумбер](#) всегда выводится в одной строке (количество колонок не настраивается).

26.3.4.4.6. Кнопка

В том случае, когда кнопка расположена в вертикальной группе, она растягивается до ширины родительской группы. Текст внутри кнопки выравнивается по центру кнопки, кроме случая, когда цвет рамки кнопки отличается от значения по умолчанию – в этом случае текст выравнивается по левому краю кнопки.

26.3.4.4.7. Группа

При отображении групп управляемой формы поддерживается только отступ для групп с режимом отображения [Обычное выделение](#) и [Сильное выделение](#).

26.3.4.4.8. Страницы

Группа вида [Страницы](#) позволяет реализовать несколько вариантов панели с закладками. Вид панели (и ее поведение) зависит от значения свойства [Отображение закладок](#):

- [Закладки сверху](#).
 - Закладки отображаются в виде тумблера. Если текст заголовка не помещается в отведенное место, то он сокращается так, что в конце текст отображается символ Картинки в заголовках не отображаются.
 - Страницы расположены внутри группы под списком закладок.
 - Активная закладка отображается инверсией заголовка.
- [Закладки снизу](#).
 - Закладки отображаются в виде панели кнопок. Если текст заголовка не помещается в отведенное место, то он сокращается так, что в конце текст отображается символ Картинки в заголовках отображаются, при этом текст всегда отображается под картинкой.
 - Страницы расположены внутри группы над списком закладок.
 - Если на экране не хватает места для отображения заголовков всех страниц группы, то в конец списка добавляется кнопка [Еще](#). Нажатие на эту кнопку открывает список заголовков всех страниц группы.
- [Закладки слева](#).
 - Закладки выводятся в виде списка. Нажатие на закладку приводит к открытию нового окна с содержимым закладки, которое замещает текущую форму (со списком закладок). Для возврата к списку закладок служит специальная кнопка (<), расположенная в левой верхней части заголовка страницы.
- [Пролистывание](#).
 - Закладки выводятся в виде точек под страницами. Текст и картинки не отображаются. Переключение между страницами выполняется жестом слева-направо или справа-налево в области страниц.
 - Страницы расположены внутри группы над точками, описывающими количество закладок.
 - В платформе для персонального компьютера данное значение интерпретируется как значение [Закладки слева](#).
- [Закладки справа](#).
 - Не поддерживается мобильной платформой.

26.3.4.4.9. Таблица

Главным отличием таблицы в мобильном приложении от таблицы в платформе для персонального компьютера является работа с текущей строкой. В таблице на платформе для персонального компьютера всегда существует текущая строка. На мобильной платформе текущая строка появляется только на время обработки интерактивного действия пользователя, которое выполняется с этой таблицей. Это отличие обуславливает большинство различий в поведении мобильной платформы, которые описаны далее.

Свойства [ТекущаяСтрока](#), [ТекущийЭлемент](#), [ТекущиеДанные](#) определены только во время выполнения:

- обработчика контекстной команды;
- событий активизации строки или ячейки;
- событий редактирования строки.

В остальное время эти свойства таблицы имеют значение [Неопределено](#).

Свойство [ТекущийРодитель](#) определено в следующих случаях:

- в иерархических списках – всегда;
- при отображении дерева – только в тех случаях, когда определено свойство [ТекущаяСтрока](#).

При выборе строки одиночным нажатием на нее, вызываются события активизации (строки, ячейки, поля). Если при этом свойство таблицы [РежимВыбора](#) установлено в значение [Истина](#), то после события активизации строки также вызывается событие [ВыборЗначения](#).

Таблица в мобильной платформе никогда не имеет горизонтальной прокрутки. В связи с этим:

- игнорируется фиксация колонок в таблице;

- значение свойства [ГоризонтальнаяПолосаПрокрутки](#) игнорируется;
- колонки, которые не помещаются в видимой части таблицы, не отображаются.

Таблица в мобильном приложении может отображаться в трех режимах:

- основной режим;
- режим выбора нескольких строк;
- режим сортировки строк.

Основной режим аналогичен режиму отображения таблицы на платформе для персонального компьютера, за исключением отображения текущей и выбранных строк. В мобильной платформе эти строки не отображаются. Исключением служит кратковременная подсветка текущей строки таблицы при открытии формы. Таблица формы автоматически подстраивается под размер содержимого в том случае, если одновременно выполняются следующие условия:

- таблица занимает всю ширину формы;
- не задана фиксированная высота таблицы;
- таблица растягивается по вертикали;
- в объекте, который отображается таблица, содержится менее 100 строк данных. Если строк более 100 – будет доступна прокрутка.

Режим выбора нескольких строк характеризуется специальной колонкой с флажками, с помощью которых и выполняется выбор. Колонка всегда отображается самой первой. В этом режиме контекстные команды таблицы доступны и действуют на все выделение сразу.

Режим сортировки строк характеризуется наличием специальной колонки с маркером, за который можно таскать строку. Эта колонка всегда отображается самой последней. В этом режиме недоступны контекстные команды таблицы.

Режим множественного выделения в таблице (свойство [РежимВыделения](#) установлено в значение [Множественный](#)) на мобильной платформе визуально никак не отображается. Признаком его существования служит специальная команда [Выбор нескольких](#), которая доступна в командной панели, связанной с таблицей. При выборе этой команды таблица переходит в режим выбора нескольких строк.

Если установлен режим множественного выбора (свойство [МножественныйВыбор](#) установлено в значение [Истина](#)), то таблица сразу находится в режиме выбора нескольких строк, при этом команда перехода в этот режим в командной панели отсутствует.

Если в таблице доступно изменение порядка строк (например, табличная часть документа), то для активизации этого режима необходимо выполнить специальную команду [Перестановка строк](#), которая доступна в командной панели, связанной с таблицей. В этом случае таблица переходит в режим сортировки строк, и сбрасываются выделенные строки (если они были).

Для таблицы поддерживаются только два режима ввода строк (свойство [РежимВводаСтрок](#)): в конце списка и в конце окна. Всегда используется режим выделения строки – [Строка](#) (свойство [РежимВыделенияСтроки](#)). Фактическое значение свойства игнорируется.

Таблица формы поддерживает использование контекстного меню, которое открывается жестом от правой границы таблицы или длительным нажатием на строку.

Заполнение командной панели и контекстного меню имеет важные отличия от платформы для персонального компьютера. На платформе для персонального компьютера вопрос распределения команд между командной панелью и контекстным меню – это вопрос дизайна формы. На мобильной платформе (в силу отсутствия текущей строки) это вопрос функциональной модели. Если команда использует свойства таблицы, связанные с текущей строкой, то такая команда должна располагаться в контекстном меню. Если такая команда размещена разработчиком в другом месте, то система может принудительно переместить ее в контекстное меню. Исключением из этого правила выступает случай, когда команда может выполнять свою работу как в связи с текущей строкой, так и без таковой. В качестве примера можно привести команду [Создать](#) динамического списка, отображающего данные иерархического справочника. Внеконтекстное использование команды предполагает создание элемента в корне справочника, а контекстное – в текущей группе.

Для того чтобы указать системе, что команда оперируется данными текущей строки, предназначено свойство команды формы [ИспользованиеТекущейСтроки](#). Для того чтобы связать команду формы с таблицей, чьи данные она должна обрабатывать, служит свойство команды [ИспользуемаяТаблица](#). Использование данного свойства означает, что на мобильной платформе невозможно реализовать команду, оперирующую одновременно данными более одной таблицы. Для управления размещением команды существует свойство команды [ОтображениеВКонтекстномМеню](#):

- [Авто](#) – команда, которая использует данные текущей строки, всегда располагается в контекстном меню. Команда, которая не использует данные текущей строки – всегда располагается в командной панели таблицы.
- [ТолькоВКонтекстномМеню](#) – команда всегда расположена в контекстном меню.
- [ДополнительноВКонтекстномМеню](#) – команда располагается в контекстном меню и в командной панели таблицы.
- [Нет](#) – команда располагается только в командной панели таблицы.

Управление расположением строки поиска также отличается от платформы для персонального компьютера. В зависимости от значения свойства таблицы [ПоложениеСтрокиПоиска](#), возможно следующее поведение:

- [Авто](#) – строка поиска доступна по нажатию кнопки командной панели;
- [Верх, Командная панель](#) – строка поиска расположена над таблицей;
- [Нет](#) – строка поиска отсутствует.

Обновление данных, отображаемых таблицей, можно осуществлять в стиле, принятом на мобильных устройствах: путем попытки потянуть список вниз или вверх при достижении первой или последней (соответственно) строки отображаемых данных ([pull to refresh](#)). Для управления данной возможностью предназначено свойство таблицы [ЗапросОбновления](#):

- [Нет](#) – обновление можно выполнить только через команду командной панели таблицы;
- Любое другое значение означает, что обновление можно инициировать попыткой потянуть список в соответствующую сторону. При этом в

области таблицы за пределами данных отображается анимация вращающейся звезды.

При выполнении обновления списка, таким образом, будет вызван обработчик таблицы [ОбработкаЗапросаОбновления](#). Если динамический список, связанной с таблицей, не содержит отображаемых данных, то внутри таблицы отображается надпись [Нет данных для отображения](#). Таблица, не связанная с динамическим списком, не отображает никакого текста при отсутствии данных для отображения.

При прокрутке таблицы, связанной с динамическим списком, в ее верхней и нижней частях появляются полупрозрачные кнопки, нажатие на которые приводит к переходу в начало или конец списка.

26.3.4.4.10. Отображение строк таблиц

Вместо редактирования данных в строке таблицы, связанной с табличной частью или таблицей значений, мобильная платформа предоставляет специальную системную форму. Эта форма содержит все отображаемые на родительской форме колонки текущей строки таблицы. При использовании этой формы необходимо помнить о следующих особенностях:

- Колонки, для которых установлено свойство [Только просмотр](#), отображаются в системной форме также без возможности редактирования;
- В системной форме не отображаются поля типа картинка и надпись;
- На системную форму действует условное оформление формы;
- Вызываются все обработчики событий;
- Системная форма не редактируется в конфигураторе и ее нельзя модифицировать из встроенного языка.

После возврата из режима редактирования, редактируемая строка кратковременно подсвечивается.

26.3.4.5. Динамический список

Динамический список, используемый в мобильной платформе, имеет следующие особенности:

- Поддерживается управление настройками динамического списка с помощью встроенного языка. Для интерактивной настройки необходимо самостоятельно реализовать соответствующие формы настройки.
- Не поддерживается интерактивный вывод списка;
- Имеется возможность выполнять сортировку по нажатию на заголовок. По нескольким колонкам таким образом сортировать нельзя.

26.3.4.6. Панели формы

Панель навигации формы реализована с помощью специального меню, которое отображается в нижней части формы в виде переключателя. Нажатие кнопки переключателя приводит к открытию соответствующей формы.

Командная панель формы:

- В режиме отображения сверху:
 - Командная панель находится в заголовке формы, правее текста.
 - Если командная панель содержит стандартные пары команд [ОК-Отмена](#) и [Да-Нет](#), то кнопка с отрицательным действием переносится в левую часть заголовка формы, а положительная отображается в командной панели формы. В меню [Еще](#) присутствуют все команды, кроме команды с отрицательным действием, которая была перенесена в заголовок формы.
 - Если командная панель не содержит вышеуказанных пар команд, но содержит стандартную команду формы [Отмена](#), [Нет](#) или [Закреть](#), то кнопка с такой командой переносится в панель в заголовке слева.
 - Команда, которая размещена в левой части заголовка формы, будет отображаться специальным символом (<), но будет выполнять свою изначальную функцию.
 - Если командная панель содержит кнопку по умолчанию, то она всегда отображается в заголовке первой.
 - Если свободного места в правой части заголовка достаточно для отображения оставшихся команд – они отображаются там. В противном случае отображается кнопка [Еще](#) и система пытается сделать так, чтобы кнопка [Еще](#) не была единственной: если команда, отображаемая в правой части заголовка еще не определена, то ей станет первая команда из списка оставшихся команд, а остальные будут расположены в меню [Еще](#).
- В режиме отображения снизу:
 - Фиксируется при прокрутке формы.
 - При появлении клавиатуры не отображается.

В список стандартных команд не включаются команды непосредственного удаления (для объектов ссылочного типа), вызова справки, вывода списка, изменения формы. Стандартные команды [Записать и закрыть](#) и [Провести и закрыть](#) отображаются в командных панелях с заголовком [Готово](#).

Для прикладных решений, работающих под управлением ОС iOS, необходимо помнить о том, что системные механизмы закрытия формы не задают пользователю вопрос о необходимости сохранения данных в том случае, когда закрывается форма с модифицированными данными.

Командная панель, связанная с таблицей, располагается сверху или снизу таблицы и по ширине не превышает ширины самой таблицы. При прокрутке таблицы командная панель всегда остается в видимой области.

26.3.5. Работа с файлами

Все приложения на мобильных устройствах работают в рамках специального механизма, обеспечивающего безопасное исполнение программ («песочница», [sandbox](#)). В рамках этого механизма, приложениям предоставляется ограниченный доступ к файловой системе мобильного устройства. При разработке мобильного приложения может требоваться доступ к файловой системе, например для промежуточного хранения данных или записи документов, сформированных в системе, для последующего обмена с другими компьютерами.

Для получения доступа к таким местам файловой системы предназначены специальные функции:

- `КаталогВременныхФайлов()` – возвращает каталог временных файлов данного приложения. При необходимости может быть очищен операционной системой;
- `РабочийКаталогДанныхПользователя()` – возвращает каталог, в котором рекомендуется хранить «долговременные» данные пользователя.
- `КаталогДокументов()` – каталог документов, которые предназначены для обмена с внешним (по отношению к мобильному приложению) окружением.

Вне зависимости от реальных возможностей мобильной ОС, не рекомендуется использовать для работы любые другие каталоги файловой системы мобильного устройства из соображений безопасности, а также требований разработчиков мобильных ОС к прикладным приложениям.

26.3.6. Специальные возможности

26.3.6.1. Работа с телефонией

Мобильная платформа предоставляет набор возможностей, которые позволяют выполнять звонки из мобильных приложений. Звонки можно выполнять как с подтверждением пользователя, так и без такового. Также система предоставляет доступ к журналу звонков. Прикладное решение может определить возможности по работе с телефонией. Для этого предназначены методы `ПоддерживаетсяНаборНомера()` и `Поддерживается журналЗвонков()` объекта `СредстваТелефонии`. При работе мобильного приложения на ОС iOS, невозможно получить доступ к журналу звонков.

Набор номера выполняется следующим образом:

```
Если СредстваТелефонии.ПоддерживаетсяНаборНомера() Тогда
    СредстваТелефонии.НабратьНомер(НомерТелефона, ВызватьСразу);
КонецЕсли;
```

[Копировать в буфер обмена](#)

Параметр `ВызватьСразу` отвечает за то, как будет выполняться вызов абонента:

- Параметр равен значению `Истина` – в этом случае при выполнении метода номер начинает набираться автоматически. На экране появляется стандартный интерфейс программы работы со звонками.
- Параметр равен значению `Ложь` – в этом случае при выполнении метода будет открыто приложение работы со звонками, в котором будет установлен заданный номер. Пользователю останется только нажать кнопку начала вызова.

Мобильные устройства, функционирующие под управление ОС Android, предоставляют возможность доступа к журналу звонков. Имеется возможность перебрать записи журнала с установкой необходимого отбора. Для работы с журналом необходимо использовать метод `ПолучитьЖурналЗвонков()` объекта `СредстваТелефонии`.

Следующий пример демонстрирует получение списка всех пропущенных звонков на мобильном устройстве.

```
Если СредстваТелефонии.ПоддерживаетсяЖурналЗвонков() Тогда
    Журнал = СредстваТелефонии.ПолучитьЖурналЗвонков();
    Отбор = Новый ОтборКомпоновкиДанных;
    НовыйЭлемент = Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));
    НовыйЭлемент.ЛевоеЗначение = Новый ПолеКомпоновкиДанных("ТипЗвонка");
    НовыйЭлемент.ВидСравнения = ВидСравненияКомпоновкиДанных.Равно;
    НовыйЭлемент.ПравоеЗначение = ТипЗвонкаЖурналаЗвонков.Пропущенный;
    НовыйЭлемент.Использование = Истина;
    ЗаписиЖурнала = Журнал.НайтиЗаписи(Отбор);
    Для каждого Запись Из ЗаписиЖурнала Цикл
        // обработка списка звонков
    КонецЦикла;
КонецЕсли;
```

[Копировать в буфер обмена](#)

При работе с журналом звонков следует помнить, что отбор журнала выполняется с помощью объекта `ОтборКомпоновкиДанных`.

26.3.6.2. Работа с сообщениями (SMS и MMS)

Мобильная платформа предоставляет набор инструментов, позволяющих работать с сообщениями (SMS и MMS). Предоставляется возможность отправлять и получать сообщения. Доступ к списку сообщений не предоставляется. Отправка сообщений возможна как в интерактивном, так и в полностью программном режиме (не интерактивном режиме).

Платформа предоставляет средства для определения возможностей используемой мобильной ОС. Для этого у объекта глобального контекста `СредстваТелефонии` существуют методы `ПоддерживаетсяОтправкаSMS()`, `ПоддерживаетсяПриемSMS()` и `ПоддерживаетсяОтправкаMMS()`. В зависимости от используемой мобильной операционной системы, существуют следующие ограничения на работу с сообщениями:

- Для ОС iOS:
 - Недоступна подписка на получение SMS/MMS;
 - Недоступна не интерактивная отправка SMS/MMS.
- Для ОС Android:
 - Недоступна не интерактивная отправка MMS.

Сообщение в «1С:Предприятии» представлено объектом `SMSCoобщение`. Отправка сообщения выглядит следующим образом:

```
&НаКлиенте
Процедура ОтправитьСМС(Кому, Текст, ПослатьИнтерактивно)
    СМС = Новый SMSCoобщение();
    СМС.Текст = Текст;
    СтрокаПолучатели = СтрЗаменить(Кому, ",", "Символы.ПС");
    Для Счетчик=1 По СтрЧислоСтрок(СтрокаПолучатели) Цикл
```

[Копировать в буфер обмена](#)

```

СМС.Получатели.Добавить(СокрЛП(СтрПолучитьСтроку(СтрокаПолучатели, Счетчик)));
КонецЦикла;
СредстваТелефонии.ПослатьSMS(СМС, ПослатьИнтерактивно);
КонецПроцедуры

```

В качестве значения параметра *Кому* могут быть переданы несколько номеров телефонов, разделенных символом «,». Параметр *ПослатьИнтерактивно* определяет, как будет отправлено сообщение:

- Параметр имеет значение *Истина* – сообщение будет отправлено сразу;
- Параметр имеет значение *Ложь* – будет открыто приложение для работы с сообщениями (установленное по умолчанию для используемой мобильной ОС). В этом приложении будет открыта форма отправки сообщения с заполненными полями. В этом случае пользователю необходимо самому нажать кнопку отправки сообщения.

Система не предоставляет доступа к журналу сообщений, однако имеется возможность подписаться (если это позволяет используемая мобильная ОС) на оповещения о получении сообщений.

[Копировать в буфер обмена](#)

```

&НаКлиенте
Процедура ПодключитьОбработчикПолученияСообщений()
    ПолучательСообщений = Новый ОписаниеОповещения("ПолучениеСообщения", ЭтотОбъект);
    СредстваТелефонии.ПодключитьОбработчикSMSСообщений(ПолучательСообщений);
КонецПроцедуры
&НаКлиенте
Процедура ПолучениеСообщения(Сообщение, ДополнительныеПараметры) Экспорт
    Сообщить(Сообщение.Отправитель + " - " + Сообщение.Текст);
КонецПроцедуры

```

При получении входящего сообщения, будет вызываться обработчик оповещения для обработки полученного сообщения. Сообщения, полученные с помощью установленного обработчика оповещения, также будут доступны и из системного приложения по работе с сообщениями.

Для того чтобы SMS-сообщение было преобразовано в MMS-сообщение, необходимо добавить вложения в сообщения. Это делается с помощью свойства *Вложения* объекта *SMSСообщение*. При этом в качестве значения свойства *ТипСодержимого* объекта *MMSВложение* следует указывать MIME-тип передаваемых данных, например, для jpeg-файла это будет *image/jpeg*.

26.3.6.3. Средства геопозиционирования

Большое количество мобильных устройств поддерживают определение координат, по которым располагается устройство. Также имеется возможно выполнять преобразование полученных координат в адрес и наоборот. Для доступа к этим функциям предназначен **провайдер** геопозиционирования. Каждый провайдер описывается именем и определенным набором параметров (возможностей). Конкретный провайдер геопозиционирования может поддерживать не все возможности.

Общая схема работы со средствами геопозиционирования выглядит следующим образом:

1. Выбирается необходимый провайдер геопозиционирования. В каждом случае следует выбирать тот провайдер, который максимально подходит для реализации поставленной задачи. В общем случае стоит выбирать провайдера, который обеспечивает минимальное энергопотребление и максимальную точность определения координат.

Пример:

[Копировать в буфер обмена](#)

```

// Определим провайдера, который не приводит к расходованию средств и
// использует для определения координат сеть сотовой связи
ИскомыйПровайдер = Неопределено;
Провайдеры = СредстваГеопозиционирования.ПолучитьПровайдеров();
Для каждого Провайдер Из Провайдеры Цикл
    Если НЕ Провайдер.Платный и Провайдер.ИспользуетСотовуюСеть Тогда
        ИскомыйПровайдер = Провайдер;
        Прервать;
    КонецЕсли;
КонецЦикла;
Если ИскомыйПровайдер = Неопределено Тогда
    Сообщить("Требуемый провайдер не обнаружен");
Иначе
    Сообщить("Найден провайдер: " + ИскомыйПровайдер.Имя);
КонецЕсли;

```

В случае необходимости получить самый энергоэффективный провайдер или самый точный провайдер, можно воспользоваться методами *ПолучитьСамогоЭнергоЭкономичногоПровайдера()* или *ПолучитьСамогоТочногоПровайдера()* (соответственно).

2. Выполняется определение координат с помощью выбранного провайдера геопозиционирования. Для получения координат используются два метода. Метод *ПолучитьПоследнееМестоположение()* возвращает последние данные о местоположении, полученные выбранным провайдером. Среди свойств объекта *ДанныеМестоположения* (который возвращает метод *ПолучитьПоследнееМестоположение()*), есть свойство *Дата*, которое описывает, когда были получены данные о местоположении. Если эти данные были получены очень давно, то необходимо вызвать метод *ОбновитьМестоположение()* для того, чтобы провайдер геопозиционирования заново определил данные о местоположении. Один из параметров метода определяет тайм-аут на определение местоположения. В том случае, если метод завершен по тайм-ауту, фактическое определение координат в данном случае зависит от особенностей реализации конкретной версии мобильной операционной системы.

Пример:

[Копировать в буфер обмена](#)

```

// Если местоположение на данном устройстве ни разу не выполнялось
// или выполнено более часа назад - обновить местоположение и
// получить определенные координаты
Координаты = СредстваГеопозиционирования.ПолучитьПоследнееМестоположение(ИмяПровайдера);
Если Координаты = Неопределено ИЛИ ТекущаяДата()-Координаты.Дата > 3600 Тогда
    СредстваГеопозиционирования.ОбновитьМестоположение(ИмяПровайдера, 60);
    Координаты = СредстваГеопозиционирования.ПолучитьПоследнееМестоположение(ИмяПровайдера);
КонецЕсли;

```

3. При необходимости, выполняется подписка на изменение координат. После того, как потребность в подписке отпала – можно отключить оповещение об изменении координат.

После получения данных о местоположении, имеется возможность выполнить несколько вспомогательных операций, связанных с полученным местоположением:

- Получить адрес по координатам. Для этого предназначен метод `ПолучитьАдресПоМестоположению()`.
- Получить координаты места по адресу. Для этого предназначен метод `ПолучитьМестоположениеПоАдресу()`.
- Показать карту, на которой отображены координаты одной (или нескольких) точки. Для этого предназначен метод `ПоказатьНаКарте()`. При этом на iOS всегда будет использоваться приложение `Карты` фирмы Apple, а на ОС Android – приложение `Google Карты`.

ПРИМЕЧАНИЕ. Для работы методов преобразования координат в адрес и обратно, на мобильном устройстве должен быть доступ в Интернет.

26.3.6.4. Работа с мультимедиа

Мобильные устройства обладают различными мультимедийными возможностями: сделать фотоснимок, сделать аудио– или видеозапись. Для различных мобильных устройств эти возможности могут быть доступны или нет. Для определения возможности выполнения той или иной операции предоставляются специальные функции: `ПоддерживаетсяФотоснимок()`, `ПоддерживаетсяВидеозапись()`, `ПоддерживаетсяАудиозапись()`. В том случае, если мобильное устройство поддерживает ту или иную возможность, можно воспользоваться методом глобального контекста, который активирует необходимое системное приложение для выполнения нужного действия: `СделатьФотоснимок()`, `СделатьВидеозапись()`, `СделатьАудиозапись()`.

Выполнение программы на встроеном языке будет ожидать выполнения требуемого действия.

В связи с тем, что мобильные устройства могут обладать двумя камерами (передней и задней), платформа предоставляет мобильному приложению возможность работать с камерами по выбору. Для этого следует использовать системное перечисление `ТипКамерыУстройства` при определении возможностей мобильного устройства и при выполнении того или иного действия.

[Копировать в буфер обмена](#)

`Передняя` = `СредстваМультимедиа.ПоддерживаетсяФотоснимок(ТипКамерыУстройства.Передняя);`
`Задняя` = `СредстваМультимедиа.ПоддерживаетсяФотоснимок(ТипКамерыУстройства.Задняя);`

Вышеприведенный программный код позволяет определить, какая из двух камер устройства поддерживает фотосъемку. С помощью этих методов можно определять наличие той или иной камеры на устройстве. При выполнении фото– или видеосъемки, также имеется возможность указать используемую камеру. Если тип камеры указан как `ТипКамерыУстройства.Авто`, то на устройствах это будет обрабатываться следующим образом:

- На устройстве имеется две камеры – будет использована задняя камера;
- На устройстве имеется одна камера – будет использована эта камера (в зависимости от наличия).

Кроме того, при попытке сделать фотоснимок или выполнить видеосъемку, имеется возможность указать некоторые дополнительные параметры, влияющие на результат действия.

При попытке выполнить фотосъемку имеется возможность (кроме типа камеры) дополнительно указать:

- С каким разрешением необходимо выполнить съемку. Если указано разрешение, не поддерживаемое устройством, будет выбрано ближайшее подходящее разрешение. Перечень поддерживаемых разрешений можно получить с помощью метода `ПолучитьПоддерживаемыеРазрешенияКамеры()` объекта `СредстваМультимедиа`. Для устройств, работающих под управлением ОС iOS, данный параметр игнорируется.
- С каким качеством будет получен результирующий снимок. Фактически определяет степень сжатия jpeg-файла. 100 – означает максимальное качество.
- Какой снимок должен быть получен в результате: цветной или черно-белый. Для устройств, работающих под управлением ОС iOS, указание на необходимость получить черно-белый снимок, не влияет на интерфейс камеры (но сам снимок будет черно-белым).

При попытке выполнить видеосъемку, кроме указания типа камеры, имеется возможность указать качество получаемого ролика. Устройства, работающие под управлением ОС iOS, игнорируют этот параметр.

26.3.6.5. Сканирование штрихкодов

Мобильное устройство, обладающее камерой, позволяет выполнять функции сканирования штрихкодов. Для доступа к этой возможности платформа предоставляет специальный интерфейс.

ПРИМЕЧАНИЕ. Для эффективного сканирования штрихкодов желательно, чтобы камера мобильного устройства обладала возможностью автоматической наводки объектива камеры на резкость (автофокусом). В противном случае сканирование штрихкодов будет затруднено.

Общая схема работы со сканером штрихкодов выглядит следующим образом:

1. Определяется возможность работы с механизмом сканирования. Для этого необходимо использовать метод `ПоддерживаетсяСканированиеШтрихКодов()` объекта глобального контекста `СредстваМультимедиа`.
2. В том случае, если сканирование поддерживается, метод `ПоказатьСканированиеШтрихКодов()` объекта `СредстваМультимедиа` открывает специальный интерфейс по сканированию штрихкодов. При выполнении операции сканирования вызывается специальный обработчик, содержащий информацию о результате сканирования.
3. Закрыть интерфейс сканирования можно или вручную, с помощью стандартной кнопки перехода к предыдущей форме интерфейса ОС Android или программно, с помощью метода `ЗакрытьСканированиеШтрихКодов()` объекта `СредстваМультимедиа`.

Рассмотрим пример работы с данным механизмом. Для демонстрации будет использоваться простая форма, на которой расположена кнопка открытия интерфейса сканирования и признак закрытия этого интерфейса после однократного сканирования. Сканированный штрихкод будет выводиться в виде сообщения.

ПРИМЕЧАНИЕ. Данный пример не является законченным инструментом. Он предназначен для демонстрации работы с механизмом сканирования штрихкодов.

В форме необходимо создать реквизит `ЗакрытьИнтерфейс` типа `Булево` и разместить его на форме. Также следует создать команду формы `ОткрытьИнтерфейсСканирования`, которую также следует разместить на форме. Модуль формы будет иметь следующий вид:

```
// Обработчик команды формы ОткрытьИнтерфейсСканирования
&НаКлиенте
Процедура ОткрытьИнтерфейсСканирования(Команда)
    ОбработчикСканирования = Новый ОписаниеОповещения("ОбработкаСканирования", ЭтотОбъект);
    ОбработчикЗакрытия = Новый ОписаниеОповещения("ОбработкаЗакрытияИнтерфейса", ЭтотОбъект);
    СредстваМультимедиа.ПоказатьСканированиеШтрихКодов("Наведите камеру на штрихкод", ОбработчикСканирования, ОбработчикЗакрытия);
КонецПроцедуры
&НаКлиенте
Процедура ОбработкаСканирования(Штрихкод, Результат, Сообщение, ДополнительныеПараметры) Экспорт
    Если Результат Тогда
        Текст = "" + Штрихкод;
    Иначе
        Текст = "Ошибка: " + Сообщение;
    КонецЕсли;
    Если ЗакрытьИнтерфейс Тогда
        СредстваМультимедиа.ЗакрытьСканированиеШтрихКодов();
    КонецЕсли;
    Сообщить(Текст);
КонецПроцедуры
&НаКлиенте
Процедура ОбработкаЗакрытияИнтерфейса(ДополнительныеПараметры) Экспорт
    Сообщить("Закрывается интерфейс сканирования");
КонецПроцедуры
&НаКлиенте
Процедура ПриОткрытии(Отказ)
    Если НЕ СредстваМультимедиа.ПоддерживаетсяСканированиеШтрихКодов() Тогда
        ВызватьИсключение "Сканирование штрихкодов не поддерживается";
    КонецЕсли;
    ЗакрытьИнтерфейс = Ложь;
КонецПроцедуры
```

При открытии интерфейса сканирования в метод `ПоказатьСканированиеШтрихКодов()` передается два описания оповещения. Первое оповещение срабатывает (параметр `<ОбработчикСканирования>`) в том случае, когда система успешно распознала штрихкод, второе оповещение (параметр `<ОбработчикЗакрытия>`) срабатывает в том случае, когда закрывается окно, предназначенное для сканирования штрихкода. Закрытие может быть как программное, так и интерактивное. Пример программного закрытия приведен в обработчике оповещения `ОбработкаСканирования()`, а интерактивное закрытие можно выполнить, нажав кнопку [Назад](#) в интерфейсе сканирования.

При открытии формы сканирования можно определить, на какие штрихкоды будет «реагировать» система: на линейные (например, EAN), на двухмерные (например, QR-код) или на все. Задается это с помощью параметра `<ТипШтрихКода>` метода `ПоказатьСканированиеШтрихКодов()`. При этом надо помнить, что указание на необходимость распознавать все типы штрихкодов снижает «производительность» сканирования, т. к. система будет тратить больше времени на поиск в изображении различных штрихкодов.

26.3.6.6. Работа с электронной почтой

Мобильная платформа предоставляет прикладному разработчику несколько способов работы с электронной почтой. Один способ позволяет реализовать полноценный почтовый клиент на платформе «1С:Предприятие» (объект `ИнтернетПочта` и связанные объекты), а другой способ позволяет реализовать только отправку почтовых сообщений с помощью приложения по работе с электронной почтой, установленное в системе по умолчанию (объект `СредстваПочты`).

При отправке почтовых сообщений типа `ТипТекстаПочтовогоСообщения.HTML` с помощью объекта `СредстваПочты` (под управлением ОС Android) выполняется конвертация текста сообщения в формат `Spannable`. Отправляемое письмо отображается большинством почтовым клиентам.

При использовании текстов почтовых сообщений с типом `ТипТекстаПочтовогоСообщения.HTML` (под управлением ОС Android) не рекомендуется использовать следующие возможности:

- каскадные таблицы стилей;
- встроенные таблицы;
- встроенные изображения;
- шрифты разных размеров.

Рекомендуется проверять отображение электронных писем на максимальном количестве устройств, чтобы минимизировать зависимость от различий реализации на разных ОС и в разных версиях почтовых программ.

Также следует помнить, что тип текстов почтовых сообщений `ТипТекстаПочтовогоСообщения.РазмеченныйТекст` не поддерживается объектом `СредстваПочты`.

26.3.6.7. Работа с контактами

Список контактов – это база данных, которая хранит информацию о субъектах, с которыми общается владелец мобильного устройства. Описание контакта содержит идентификационные данные контакта, адреса, телефоны и другую информацию, необходимую для связи с субъектом. Также, на мобильном устройстве могут быть зарегистрированы учетные записи различных сервисов. Под учетной записью понимается набор информации, которые пользователь использует для идентификации себя в каком-либо сервисе. Например, это имя пользователя и пароль. На мобильном устройстве всегда существует учетная запись, описывающая само устройство (локальная учетная запись).

Каждая запись в списке контактов связана с какой-либо одной учетной записью. Мобильная ОС может выполнять синхронизацию списка контактов с данными учетной записи (зависит от настроек ОС). Если контакт связан с локальной учетной записью, то синхронизация информации не выполняется и все данные расположены только на мобильном устройстве.

Основная информация о контакте хранится в объектах типа `ДанныеКонтакта`. Объекты данного типа содержат всю информацию о контакте, но не содержат связь контакта с учетной записью. В описании контакта есть несколько типов данных, которые могут содержать более одного значения. Такими данными являются номера телефонов, адреса, адреса электронной почты и другие. Для хранения таких данных предназначены массивы объектов типа `ЭлементДанныхКонтакта`. Для хранения списка идентификаторов в программах мгновенных сообщений предназначен массив объектов `ЭлементДанныхКонтактаМгновенныеСообщения`.

Учетные записи, зарегистрированные на данном мобильном устройстве для синхронизации контактов, описываются с помощью объекта `УчетнаяЗаписьКонтактов`.

Связь между учетной записью контактов и собственно элементов списка контактов образует объект `ДанныеКонтактаУчетнойЗаписи`. Кроме информации о контакте и учетной записи, данный объект содержит также идентификатор, по которому выполняется синхронизация между списком контактов на мобильном устройстве и учетной записью. Можно сказать, что свойство `ГлобальныйКлючКонтакта` объекта `ДанныеКонтактаУчетнойЗаписи` является аналогом свойства `Ссылка` для прикладных объектов «1С:Предприятие».

Для идентификации контакта на мобильном устройстве существует объект `ЛокальныйКлючКонтакта`. Все операции с контактами (создание, изменение, удаление, поиск) выполняются с использованием данного объекта. Для выполнения операций с контактами следует использовать менеджер контактов (одноименный объект).

Следующий пример демонстрирует создание элемента списка контактов в локальной учетной записи:

[Копировать в буфер обмена](#)

```
&НаКлиенте
Процедура СоздатьЭлементКонтакта(Имя, Отчество, Фамилия, ЭлектроннаяПочта, Телефон)
    ДанныеКонтакта = Новый ДанныеКонтакта;
    ДанныеКонтакта.Имя = Имя;
    ДанныеКонтакта.Отчество = Отчество;
    ДанныеКонтакта.Фамилия = Фамилия;
    ЭлементПочта = Новый ЭлементДанныхКонтакта(ТипАдресаЭлектроннойПочтыДанныхКонтакта.Рабочий, ЭлектроннаяПочта);
    ДанныеКонтакта.АдресаЭлектроннойПочты.Добавить(ЭлементПочта);
    ЭлементТелефон = Новый ЭлементДанныхКонтакта(ТипНомераТелефонаДанныхКонтакта.Рабочий, Телефон);
    ДанныеКонтакта.НомераТелефонов.Добавить(ЭлементТелефон);
    МК = Новый МенеджерКонтактов;
    ЛокальныеКонтакты = МК.ПолучитьЛокальнуюУчетнуюЗаписьКонтактов();
    Контакт = Новый ДанныеКонтактаУчетнойЗаписи(ДанныеКонтакта, ЛокальныеКонтакты);
    МК.ДобавитьКонтакт(Контакт);
КонiecПроцедуры
```

26.3.6.8. Работа с календарем

На мобильных устройствах имеется возможность просмотра календаря и заведения в нем различных событий. События характеризуются некоторым набором характеристик, таких как дата и время начала и завершения, описание, занятые лица и т. д. Также, на мобильном устройстве могут быть зарегистрированы учетные записи различных сервисов. Под учетной записью понимается набор информации, которые пользователь использует для идентификации себя в каком-либо сервисе. Например, это имя пользователя и пароль. На мобильном устройстве всегда существует учетная запись, описывающая само устройство (локальная учетная запись).

На мобильном устройстве может быть несколько календарей (с разными именами). Каждый календарь связан с какой-либо одной учетной записью. Мобильная ОС может выполнять синхронизацию событий календарей с данными учетной записи (зависит от настроек ОС). Если события календаря связаны с локальной учетной записью, то синхронизация информации не выполняется и все данные расположены только на мобильном устройстве.

Основная информация о календаре хранится в объекте `ДанныеКалендаря`. События календаря описываются объектом `ДанныеСобытияКалендаря`. Данные объекты описывают характеристики календаря или события календаря, но не описывают, с какой учетной записью связан объект.

Учетные записи, зарегистрированные на данном мобильном устройстве для синхронизации контактов, описываются с помощью объекта `УчетнаяЗаписьКалендарей`.

Связь между данными календарей и событий и учетными записями образуются объектами `ДанныеКалендаряУчетнойЗаписи` и `ДанныеСобытияКалендаряУчетнойЗаписи`. Эти объекты, кроме информации о собственно данных и информации об учетной записи, хранят информацию значения ключей, по которым выполняется синхронизация между мобильным устройством и сервисом, с которым выполняется синхронизация. Для синхронизации данных о календарях используется `ГлобальныйКлючКалендаря`, для синхронизации данных о событиях календарей используется `ГлобальныйКлючСобытияКалендаря`.

Для идентификации данных календарей и событий на мобильном устройстве существуют объекты `ЛокальныйКлючКалендаря` и `ЛокальныйКлючСобытияКалендаря`. Все операции с календарями и событиями (создание, изменение, удаление, поиск) выполняются с использованием данного объекта. Для выполнения операций с календарями и событиями следует использовать менеджер календарей (одноименный объект).

26.3.6.9. Ориентация экрана

Платформа предоставляет информацию о некоторых характеристиках экрана устройства, на котором выполняется клиентское приложение (персональный компьютер или мобильное устройство). Для этого служит метод `ПолучитьИнформациюЭкрановКлиента()`. В результате возвращается массив структур, описывающих экраны, подключенные к устройству.

Каждый элемент массива описывает один экран, подключенный к устройству. Параметры `Высота` и `Ширина` описывают, соответственно, высоту и ширину экрана в точках. При этом для мобильного устройства эти параметры зависят от ориентации устройства (если это разрешено в настройках), а платформы для персонального компьютера всегда возвращает канонические параметры подключенных экранов. Таким образом, на мобильном устройстве можно анализировать, в каком положении находится устройство и на основании этого принимать решения о трансформации интерфейса приложения. Для облегчения работы с ориентацией существует событие `ПриИзмененииПараметровЭкрана`, которое можно обработать в модуле управляемого приложения и в модуле управляемой формы.

В качестве примера рассмотрим изменение отображения формы, отображаемой на рабочем столе. В указанной форме есть два списка, которые отображают некоторую информацию. При портретной ориентации экрана (узкая сторона устройства – нижняя) списки будут отображаться друг под другом, а при ландшафтной (широкая сторона устройства – нижняя) ориентации – списки расположены рядом. Для реализации такого поведения в модуле формы, расположенной на рабочем столе, следует разместить обработчик события `ПриИзмененииПараметровЭкрана` следующего вида:

[Копировать в буфер обмена](#)

```
&НаКлиенте
Процедура ПриИзмененииПараметровЭкрана()
    МассивИнформаций = ПолучитьИнформациюЭкрановКлиента();
    Если МассивИнформаций[0].Ширина > МассивИнформаций[0].Высота Тогда
        ЭтаФорма.Группировка = ГруппировкаПодчиненныхЭлементовФормы.Горизонтальная;
    Иначе
        ЭтаФорма.Группировка = ГруппировкаПодчиненныхЭлементовФормы.Вертикальная;
    КонiecЕсли;
КонiecПроцедуры
```

26.3.6.10. Работа с уведомлениями

26.3.6.10.1. Общая информация

При работе информационной системы регулярно возникает необходимость оповестить пользователя о каком-либо событии. Событие может

возникать как по расписанию, так и в случае возникновения другого события. Само событие может наступить как на мобильном устройстве, которым владеет пользователь, так и в каком-то другом месте информационной системы.

Для извещения пользователя о наступлении событий служат уведомления. Они могут создаваться локально или рассылаться с помощью push-сервисов. Локальные уведомления служат для информирования о событиях, которые происходят на устройстве пользователя. Обычно источником push-уведомлений выступает центральный узел информационной системы. С помощью push-уведомлений он информирует удаленных пользователей о том, что изменились данные, важные для пользователей. Однако это не единственный сценарий использования push-уведомлений. Push-уведомления могут рассылать специализированные сервисы, информирующие о курсах валюты, результатах спортивных состязаний, прогнозах погоды и т. д.

26.3.6.10.2. Локальные уведомления

Локальные уведомления служат для извещения о событиях, которые происходят непосредственно на мобильном устройстве. Локальные уведомления бывают моментальными и запланированными. Они используются для информирования о разных событиях и, кроме того, их отличает поведение на мобильном устройстве. Моментальные уведомления используются в том случае, когда необходимо «здесь и сейчас» оповестить пользователя о наступлении события, например завершения какого-либо процесса. Таким процессом может быть операция синхронизации с внешней системой. Запланированные уведомления позволяют в назначенное время напомнить пользователю о необходимости выполнить какую-либо операцию. Запланированное уведомление может быть однократным или периодическим. Однократное запланированное уведомление можно использовать, например, если в системе помечено, что какому-то контрагенту необходимо перезвонить в определенное время. Повторяемое запланированное уведомление можно использовать, например, в том случае, когда какая-то операция повторяется каждый день, вне зависимости от внешних событий.

Чтобы использовать локальные уведомления, следует установить разрешение [Локальные уведомления](#) (см. [здесь](#)). Сборщик мобильных приложений добавляет соответствующее разрешение для собираемого мобильного приложения. Следует учитывать, что мобильная платформа разработчика обладает всеми разрешениями для работы с локальными уведомлениями. Поэтому рекомендуется после окончания разработки механизма проверить его функционирование на обычной мобильной платформе. Для этого следует воспользоваться сборщиком мобильных приложений и проверить работоспособность получившегося мобильного приложения на мобильном устройстве.

Для работы с локальными уведомлениями используется менеджер доставляемых уведомлений (свойство глобального контекста `ДоставляемыеУведомления`) и объект `ДоставляемоеУведомление`. Объект описывает собственно уведомление (включая параметры его отображения), а с помощью менеджера выполняется управление механизмом уведомлений.

Рассмотрим создание локальных уведомлений более подробно.

[Копировать в буфер обмена](#)

```
&НаКлиенте
Функция СоздатьУведомление(Заголовок, Текст, Данные = 0, ДатаСрабатывания, ИнтервалПовторения = 0)
    Уведомление = Новый ДоставляемоеУведомление;
    Уведомление.Заголовок = Заголовок;
    Уведомление.Текст = Текст;
    Уведомление.Данные = Данные;
    Уведомление.ДатаПоявленияУниверсальноеВремя = ДатаСрабатывания;
    Уведомление.ИнтервалПовтора = ИнтервалПовторения;
    Уведомление.ЗвуковоеОповещение = ЗвуковоеОповещение.ПоУмолчанию;
    Возврат Уведомление;
КонiecФункции
&НаКлиенте
Процедура ЗадатьУведомление(Команда)
    Уведомление = СоздатьУведомление("Заголовок", "Текст", 2014, '00010101');
    ДоставляемыеУведомления.ДобавитьЛокальноеУведомление(Уведомление);
КонiecПроцедуры
```

При создании уведомления следует помнить о следующих особенностях:

- Дата и время первого отображения уведомления задается в формате универсального времени (UTC). Для перевода из локального времени можно использовать функцию глобального контекста `УниверсальноеВремя()`.
- Если свойство `ДатаПоявленияУниверсальноеВремя` задано пустым значением (или не задано вовсе), то уведомление будет отображено сразу. Таким уведомлением можно извещать пользователя о том, что мобильное приложение завершило какую-то длительную операцию, например, завершило операцию синхронизации данных.
- Если необходимо, чтобы уведомление появлялось регулярно, следует использовать свойство `ИнтервалПовтора` объекта `ДоставляемоеУведомление`. В результате можно реализовать простой вариант безусловного таймера, который по истечении заданного интервала времени будет формировать уведомление. Значение, описывающее интервал повторения уведомления, зависит от используемой мобильной операционной системы. Особенности указания интервала описаны в синтакс-помощнике. Система не предоставляет возможности указать дату и время окончания срабатывания уведомления.
- При необходимости, имеется возможность установить наклейку – число, отображаемое в шторке уведомлений (OC Android) или поверх иконки приложения (OC iOS). Для этого служит свойство `Наклейка`. При работе под управлением OC iOS имеется возможность устанавливать и получать значение наклейки с помощью методов `УстановитьНаклейку()` и `ПолучитьНаклейку()` менеджера доставляемых уведомлений.
- При отображении уведомления имеется возможность воспроизвести некоторую мелодию. Для указания воспроизводимой мелодии служит свойство `ЗвуковоеОповещение` объекта `ДоставляемоеУведомление`. Если в качестве значения этого свойства установлено значение системного перечисления `ЗвуковоеОповещение.ПоУмолчанию`, то будет воспроизводиться мелодия, установленная для уведомлений в используемой операционной системе. Если в качестве значения свойства `ЗвуковоеОповещение` указано имя файла, то будет воспроизведен именно этот файл. Однако на мобильной платформе разработчика отсутствует возможность подключать ресурсы (к которым относятся воспроизводимые файлы). Для того, чтобы указанный файл был прикреплен к мобильному приложению, следует воспользоваться возможностями сборщика мобильных приложений (см. [здесь](#)). В силу этого, проверить использование звуковых оповещений из файлов можно только на обычной мобильной платформе.

В ряде случаев может возникнуть необходимость удалять установленные (но еще не наступившие) уведомления. Такое может потребоваться в тех случаях, когда действие, для напоминания о котором установлено уведомление, уже выполнено. Для удаления уведомления можно использовать метод `ОтменитьЛокальныеУведомления()` менеджера доставляемых уведомлений. При этом следует помнить, что отменяются сразу все локальные уведомления, которые были установлены данным мобильным приложением, и время срабатывания которых еще не наступило.

После появления уведомления, нажатие на него приведет к запуску (или активизации) приложения. Однако этого может оказаться недостаточно для полноценной обработки уведомления. В этом случае можно воспользоваться обработчиком уведомлений. Подключается этот обработчик с помощью метода `ПодключитьОбработчикУведомлений()` менеджера доставляемых уведомлений.

[Копировать в буфер обмена](#)

```

Процедура ПриОткрытии(Отказ)
    Обработчик = Новый ОписаниеОповещения("ОбработчикУведомлений", ЭтотОбъект);
    ДоставляемыеУведомления.ПодключитьОбработчикУведомлений(Обработчик);
КонецПроцедуры
&НаКлиенте
Процедура ОбработчикУведомлений(Уведомление, Локальное, Показано, ДополнительныеПараметры) Экспорт
    // Важные и нужные действия
КонецПроцедуры

```

В результате, после того, как пользователь мобильного приложения нажмет на уведомление, будет активизировано мобильное приложение, создавшее уведомление, и в этом приложении будет вызван обработчик уведомления. В рамках этого обработчика можно реализовать необходимую обработку уведомления. Следует помнить, что объект, который передается в параметре *Уведомление*, содержит реальные данные только в свойствах *Текст* и *Данные*. Остальные свойства заполнены значениями по умолчанию.

Также следует отметить, что уведомление может отображаться в шторке уведомлений, а может не отображаться. Для оповещения мобильного приложения об этом, в обработчик уведомлений передается параметр *Показано*. Так, если мобильное приложение запущено и активно, то локальное уведомление не будет отображено на экране, но обработчик уведомлений будет вызван, а параметр *Показано* будет установлен в значение *Ложь*. В этом случае задача привлечения внимания пользователя лежит на прикладном разработчике.

26.3.6.10.3. PUSH-уведомления

Общая информация

Push-уведомления (далее в этом разделе термины «уведомление» и «push-уведомление» используются равнозначно) предназначены для оповещения о событиях, которые произошли где-то в другом месте. Например, в корпоративной информационной системе изменились какие-то данные, важные для мобильных пользователей системы.

При работе с push-уведомлениями используются несколько различных систем:

1. Собственно мобильное приложение. Получает уведомления и реализует реакцию на них со стороны мобильного устройства.
2. Прикладное решение, которое занимается рассылкой уведомлений (отправитель). В роли отправителя выступает прикладное решение, созданное на платформе «1С:Предприятие».
3. Сервис доставки уведомлений. Этот сервис обеспечивает передачу уведомления от отправителя получателю (мобильному приложению). В качестве такого сервиса выступает Apple Push Notification Service (APNs, <https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html>, на английском языке) и Google Cloud Messaging (GCM, <http://developer.android.com/intl/ru/google/gcm/index.html>, на английском языке).
4. Фирмой «1С» разработан специальный вспомогательный сервис для отправки уведомлений. Этот сервис предназначен для облегчения реализации отправки уведомлений из тиражных прикладных решений. При работе под управлением iOS сервис позволяет изолировать получателей уведомлений одного отправителя, от других отправителей, которые работают с тем же мобильным приложением. Кроме того, сервис позволяет избежать публикации сертификата мобильного приложения (что необходимо при непосредственной работе с сервисом APNs). Особенности работы с этим сервисом будут описаны далее, в соответствующих разделах. Следует понимать, что сервис фирмы «1С» не может работать без использования APNs и GCM. Сервис расположен по адресу <https://pushnotifications.1c.com>.

В общем случае отправка уведомления выглядит следующим образом:

- Отправитель формирует уведомление;
- Отправитель определяет список получателей уведомления;
- Отправитель подключается к сервису уведомлений;
- Отправитель передает сервису уведомление и список получателей;
- Сервис обеспечивает доставку уведомлений на мобильные устройства;
- Мобильное приложение на устройстве обеспечивает обработку уведомлений.

Сервис доставки push-уведомлений не гарантирует доставки уведомления до мобильного устройства.

Чтобы использовать уведомления, следует установить разрешение *Push-уведомления*. Сборщик мобильных приложений добавляет соответствующее разрешение для собираемого мобильного приложения. Работа с мобильной платформой разработчика имеет некоторые особенности, которые будут рассмотрены ниже. В любом случае, рекомендуется после окончания разработки механизма проверить его функционирование на обычной мобильной платформе. Для этого следует воспользоваться сборщиком мобильных приложений и проверить работоспособность получившегося мобильного приложения с использованием всех сервисов доставки уведомлений.

Обработка push-уведомления мало отличается от обработки локального уведомления (см. [здесь](#)). Очевидно, что параметр *Локальное* будет иметь значение *Ложь*, если обработчик будет вызван для обработки push-уведомления.

Схема работы с сервисами рассылки уведомлений

Для понимания работы с push-уведомлениями, следует учитывать, что в процессе разработки и внедрения системы участвуют несколько различных ролей. В ряде случаев исполнители ролей могут объединяться, но само количество ролей остается неизменным:

- **Разработчик мобильного приложения** – занимается разработкой мобильного приложения (получателя уведомлений). Реализует всю необходимую функциональность приложения для получения уведомлений.
- **Разработчик отправителя уведомлений** – занимается разработкой информационной системы, которая занимается рассылкой уведомлений и механизма получения и хранения идентификаторов получателей уведомлений.
- **Пользователь мобильного приложения** – использует мобильное приложение. Настройка мобильного приложения также выполняется этим пользователем, поэтому настройки мобильного приложения должны быть максимально просты и понятны даже неподготовленному пользователю.
- **Внедренец отправителя** – занимается установкой и настройкой системы-отправителя уведомлений в сети пользователя системы.

При рассмотрении схем работы с тем или иным сервисом, описание действий будет происходить от лица носителя той или иной роли.

Схема работы содержит описание действий, которые надлежит выполнить представителю той или иной роли в момент выполнения своих действий. Порядок описания действий всегда будет одинаковым (разработчик мобильного приложения, разработчик отправителя, внедренец отправителя, пользователь мобильного приложения), однако это не означает, что физический порядок действий будет ровно таким же.

Кроме действий, описанных далее в разделе, реализация следующих этапов считается обязательной, и далее про них отдельно упоминаться не будет:

- Реализация реакции на уведомление – выполняет разработчик мобильного приложения;
- Реализация отправки уведомлений – выполняет разработчик отправителя.

Также считается, что мобильное приложение и отправитель «знают» о том, что они могут взаимодействовать друг с другом:

- Отправитель имеет некоторый публичный программный интерфейс (**сервис обмена**) для того, чтобы:
 - Предоставить мобильному приложению номер проекта GCM (если таковой необходим для функционирования системы).
 - Получить идентификатор получателя уведомления от мобильного приложения.
- Получатель (мобильное приложение) умеет использовать сервис обмена, который публикует отправитель для выполнения обеих действий.

Далее будут рассмотрены действия (по ролям), которые необходимо выполнить для работы с тем или иным сервисом отправки уведомлений. Пошаговые инструкции для выполнения конкретных шагов будут приведены далее в данном разделе.

APNs

Разработчик мобильного приложения должен:

- Зарегистрировать для мобильного приложения возможность получения push-уведомлений на сайте компании Apple. При регистрации следует получить специальный сертификат.
- Предоставить полученный сертификат отправителю уведомлений.
- Реализовать получение идентификатора получателя уведомлений и доставку получившегося объекта отправителю уведомлений.

Разработчик отправителя должен:

- Реализовать механизм ввода и хранения сертификатов получателей уведомлений.
- Реализовать механизм получения и хранения идентификаторов получателей уведомлений.

Внедренец отправителя должен:

- Загрузить в систему-отправителя сертификат мобильного приложения, которое будет получать уведомления.
- Сертификат должен быть получен у разработчика мобильного приложения.

Пользователь мобильного приложения должен:

- Указать мобильному приложению параметры сервиса обмена, который предоставляет приложение-отправитель.
- Инициировать команду создания идентификатора получателя уведомлений и отправку его отправителю уведомлений.

GCM

Разработчик мобильного приложения должен:

- Реализовать ввод и хранение номеров проектов тех экземпляров системы отправителя, уведомления от которых может получать мобильное приложение.
- Реализовать получение идентификатора получателя уведомлений и доставку получившегося объекта отправителю уведомлений.

Разработчик отправителя должен:

- Реализовать механизм ввода и хранения ключа программного интерфейса (**API KEY**) и номера проекта для каждого экземпляра системы-отправителя уведомлений. Регистрацию конкретного экземпляра отправителя на сайте Google выполняет внедренец отправителя.
- Реализовать механизм получения и хранения идентификаторов получателей уведомлений.

Внедренец отправителя должен:

- Выполнить регистрацию конкретного экземпляра отправителя на сайте Google. Получившийся ключ программного интерфейса (**API KEY**) и номер проекта указать в соответствующих настройках отправителя.

Пользователь мобильного приложения должен:

- Указать мобильному приложению параметры сервиса обмена, который предоставляет приложение-отправитель.
- Инициировать выполнения команды (или команд) для получения номера проекта (который сохранен в информационной базе отправителя), создания идентификатора получателя уведомлений и отправку его отправителю уведомлений.

Сервис фирмы «1С»

В силу того, что прямое общение с сервисами APNs и GCM существенно различается, фирмой «1С» был реализован специальный сервис, который несколько упрощает процесс отправки и получения уведомления для тиражных решений.

Разработчик мобильного приложения должен:

- Выбрать, какие сервисы будут использоваться для доставки уведомлений на мобильное устройство:
 - При необходимости указать в сервисе фирмы «1С», что мобильное приложение может получать уведомления с помощью сервиса APNs. Для этого следует выполнить регистрацию мобильного приложения на сайте Apple и предоставить полученный сертификат мобильного приложения сервису фирмы «1С».
 - При необходимости указать в сервисе фирмы «1С», что мобильное приложение может получать уведомления с помощью сервиса GCM. Использовать специальный номер проекта **797785657074**, если получать уведомления планируется только через сервис фирмы «1С». Если планируется получать уведомления от множества приложений, зарегистрированных на сайте Google, необходимо реализовать ввод и хранение номеров проектов тех экземпляров системы отправителя (например, с помощью сервиса обмена), уведомления от которых может получать мобильное приложение. Регистрацию конкретного экземпляра отправителя на сайте Google выполняет внедренец отправителя.
- Реализовать получение идентификатора получателя уведомлений и доставку получившегося объекта отправителю уведомлений.

Разработчик отправителя должен:

- Реализовать механизм ввода и хранения ключа программного интерфейса (**API KEY Отправителя**) для текущего экземпляра системы-отправителя уведомлений. Регистрацию конкретного экземпляра отправителя на сайте сервиса фирмы «1С» выполняет внедренец отправителя.
- Реализовать механизм получения и хранения идентификаторов получателей уведомлений.

Внедренец отправителя должен:

- Выполнить регистрацию конкретного экземпляра отправителя на сайте сервиса фирмы «1С». Получившийся ключ программного интерфейса (**API KEY Отправителя**) ввести в соответствующие настройки отправителя.

Пользователь мобильного приложения должен:

- Указать мобильному приложению параметры сервиса обмена, который предоставляет приложение-отправитель.
- Инициировать команду создания идентификатора получателя уведомлений и отправку его отправителю уведомлений.

Работа с сервисом APNs

Для проверки работы механизма отправки уведомлений на ОС iOS, необходимо самостоятельно собрать собственную мобильную платформу разработчика. При сборке следует использовать специальный профиль провизии (**provision profile**), для которого разрешена возможность работать с push-уведомлениями. Кроме того, для «собственной» мобильной платформы разработчика необходимо указать новый идентификатор мобильного приложения, при этом идентификаторы **com.elc.mobile** и **com.elc.mobile.ios** использовать не смогут, т. к. они зарезервированы для использования фирмой «1С».

Для установки соединения с APNs, на отправляющем компьютере должен быть доступен удостоверяющий сертификат мобильного приложения получателя уведомлений. Сертификаты для APNs бывают двух видов **Apple Development iOS Push Services** (сертификат разработчика) и **Apple Production iOS Push Services** (сертификат публичного мобильного приложения). Сертификат разработчика предназначен для разработки мобильного приложения и его отладки. Его следует использовать, если приложение установлено на устройство с помощью средств разработки. Сертификат публичного мобильного приложения следует использовать, если мобильное приложение распространяется через магазин приложений.

В связи с этим, мобильная платформа предъявляет следующие требования к сертификату:

- Сертификат приложения должен быть предоставлен в формате **.PEM**.
- Сертификат должен содержать секцию **Bag Attributes**, в которой должен присутствовать атрибут **friendlyName**. По значению атрибута **friendlyName** платформа различает сертификат разработчика и сертификат публичного мобильного приложения. Сертификат без атрибута **friendlyName** будет отвергнут платформой «1С:Предприятие».

ВНИМАНИЕ! Описание консоли разработчика приводится по состоянию на момент написания документации.

Для подготовки к рассылке уведомлений с использованием APNs, разработчик мобильного приложения должен выполнить следующие действия:

1. Необходимо войти в консоль разработчика (iOS Dev Center, <https://developer.apple.com/devcenter/ios/index.action>). Пользователь, от имени которого выполняется вход в консоль, должен иметь права администратора (в консоли разработчика).
2. В правой части окна веб-браузера необходимо выбрать пункт меню **Certificates, Identifiers & Profiles** в разделе **iOS Developer Program**.
3. Если у мобильного приложения, которое будет получать уведомления, отсутствует **App ID**, то его необходимо создать. Порядок создания описан ниже. Если **App ID** есть – следует пропустить этот пункт и перейти к следующему пункту.
 1. В разделе **iOS Apps** выбрать пункт **Identifiers**.
 2. Нажать кнопку **+** над списком **iOS AppIDs**. Если у пользователя недостаточно прав – кнопка создания будет недоступна.
 3. В поле **Name** следует задать описание приложения (латинскими буквами). В разделе **App ID Suffix** выбрать пункт **Explicit App ID**. В поле **Bundle ID** следует указать полный идентификатор мобильного приложения (можно получить в сборщике мобильных приложений). В разделе **App Services** следует отметить пункт **Push Notifications**.
 4. Для перехода к следующему экрану следует нажать **Continue**.
 5. Следует проверить все указанные параметры и нажать кнопку **Submit** для подтверждения создания **App ID**.
 6. Нажатие кнопки **Done** полностью завершает процесс создания **App ID**.
4. Необходимо создать профиль провизии (**provision profile**) для сборки мобильного приложения, которое может получать уведомления. Данный профиль провизии (**provision profile**) в дальнейшем должен быть использован в сборщике мобильных приложений.
 1. Необходимо выбрать пункт меню **Provisioning profiles – All** в левой части экрана **Certificates, Identifiers & Profiles**.
 2. Необходимо нажать кнопку **+** над списком **iOS Provisioning Profiles**.
 3. Вид профиля выбирается в зависимости от того, для чего он создается. Если профиль провизии (**provision profile**) создается для разработки

и отладки приложения, то в разделе [Development](#) необходимо выбрать пункт [iOS App Development](#). Если профиль провизии (provision profile) создается для распространения приложения, то в разделе [Distribution](#) следует выбрать пункт [App Store](#). В зависимости от выбранного типа профиля провизии (provision profile), дальнейшие шаги помощника создания профиля провизии (provision profile) будут различаться.

1. Выбран профиль провизии (provision profile) для разработчика и нажата кнопка [Continue](#).

1. Необходимо выбрать [App ID](#) приложения, для сборки которого создается профиль провизии (provision profile), и нажать кнопку [Continue](#).

2. Необходимо выбрать сертификаты тех разработчиков, которые смогут создавать приложения в Xcode с использованием данного профиля провизии (provision profile) и затем нажать кнопку [Continue](#).

3. Необходимо выбрать устройства, на которых будет работать приложение, собранное с данным профилем провизии (provision profile) и затем нажать кнопку [Continue](#).

4. Необходимо ввести описание создаваемого профиля (латинскими буквами) и нажать кнопку [Generate](#).

2. Выбран профиль провизии (provision profile) для распространения нажата кнопка [Continue](#).

1. Необходимо выбрать [App ID](#) приложения, для сборки которого создается профиль провизии (provision profile), и нажать кнопку [Continue](#).

2. Необходимо выбрать сертификат фирмы-поставщика прикладного решения и затем нажать кнопку [Continue](#).

3. Необходимо ввести описание создаваемого профиля (латинскими буквами) и нажать кнопку [Generate](#).

3. Затем следует скачать созданный профиль провизии (provision profile) с помощью кнопки [Download](#) для дальнейшего использования (в системе Xcode или сборщике мобильных приложений).

5. Для создания сертификата мобильного приложения вначале необходимо сформировать файл запроса сертификата. Для выполнения данной операции необходим компьютер Mac. Не закрывайте окно веб-браузера с открытой консолью разработчика во время создания запроса сертификата.

1. Необходимо запустить утилиту [Связка ключей](#).

2. Выбрать команду [Связка ключей – Ассистент сертификации – Запросить сертификат у бюро сертификации](#).

3. В открывшемся диалоге необходимо заполнить поля, при этом в качестве значения поля [E-mail пользователя](#) рекомендуется указать тот адрес электронной почты, который указывался при регистрации в программе [iOS Developer Program](#). В качестве значения свойства [Запрос](#) необходимо указать [Сохранен на диске](#).

4. После нажатия кнопки [Продолжить](#) будет сформирован файл запроса сертификата [*.certSigningRequest](#).

6. Формируем сертификат для отправителя уведомлений. Для этого необходимо вернуться в окно веб-браузера с открытой консолью разработчика.

1. Необходимо выбрать пункт меню [Certificates – All](#) в левой части экрана [Certificates, Identifiers & Profiles](#).

2. Необходимо нажать кнопку [+](#) над списком [iOS Certificates](#).

3. В зависимости от потребности, необходимо указать пункт [Apple Push Notification service SSL \(Sandbox\)](#) (если нужен сертификат для разработки и отладки приложения) или пункт [Apple Push Notification service SSL \(Production\)](#) (если нужен сертификат для распространяемого приложения). Затем следует нажать кнопку [Continue](#).

4. Необходимо указать [App ID](#) используемого приложения и нажать кнопку [Continue](#).

5. Необходимо нажать кнопку [Continue](#). На данном экране предлагается создать файл запроса сертификата, который уже создан на предыдущем шаге.

6. Необходимо загрузить файл, сформированный на предыдущем шаге (с расширением [.certSigningRequest](#)) и нажать кнопку [Generate](#).

7. После окончания формирования сертификата, его следует скачать с сайта с помощью кнопки [Download](#).

7. Последним шагом необходимо выполнить конвертацию сертификата в необходимые форматы. Для этого необходим компьютер Mac, на котором формировался файл запроса сертификата:

1. Для конвертации также следует использовать программу [Связка ключей](#).

2. Полученный на сайте сертификат загружается в [Связку ключей](#).

3. Выполняется экспорт сертификата из [Связки ключей](#) в формате [.P12](#).

4. Выполняется конвертация получившегося сертификата из формата [.P12](#) в формат [.PEM](#). Для этого следует в консоли MacOS X выполнить следующую команду:

```
openssl pkcs12 -in pushcert.p12 -out pushcert.pem -nodes -clcerts
```

[Копировать в буфер обмена](#)

Получившиеся сертификаты необходимы для:

- Сертификат в формате [.P12](#) – необходим в том случае, если рассылка уведомлений выполняется с помощью сервиса фирмы «1С». Сертификат должен использовать разработчиком получателя для регистрации приложения-получателя уведомлений в сервисе фирмы «1С».

- Сертификат в формате [.PEM](#) – необходим в том случае, если уведомления будут отправляться непосредственно через сервис APNs, и в этом случае сертификат должен быть предоставлен разработчику отправителя или внедренцу отправителя (если интерфейс программы-отправителя уведомлений предусматривает возможность указания сертификата программы-получателя уведомлений).

ВНИМАНИЕ! Описание консоли разработчика приводится по состоянию на момент написания документации.

Следующие действия должен выполнить внедренец приложения-отправителя во время развертывания и настройки приложения-отправителя в компьютерной сети заказчика:

1. Необходимо войти в консоль разработчика (Google Developers Console, <https://console.developers.google.com/project>).
2. Необходимо создать проект (кнопка **Create Project**).
3. После окончания создания проекта перейдите в свойства проекта (**Projects – <Имя вашего проекта> – Overview**). В верхней части формы, над надписью **Project Dashboard**, будет показан номер проекта (**Project Number**). Этот номер проекта необходим для того, чтобы мобильное устройство смогло сформировать идентификатор получателя.
4. Необходимо включить возможность использования сервиса GCM. Для этого необходимо выбрать пункт меню **APIs & auth – APIs** (в левой части окна веб-браузера). Затем необходимо найти **Google Cloud Messaging for Android** в списке **Browse APIs** и включить его (перевести переключатель **Status** в положение **ON**). После включения следует согласиться с тем, что вы принимаете соглашения об использовании, которые предлагаются консолью разработчика.
5. Необходимо получить ключ программного интерфейса (**API Key**). Для этого следует выбрать пункт меню **APIs & auth – Credentials** (в левой части окна веб-браузера). В разделе **Public API Access** следует нажать кнопку **Create new Key**. В открывшемся диалоге выбрать кнопку **Server Key**. Затем необходимо указать ip-адрес сервера, который будет выступать отправителем для вашего приложения. Другими словами в данном окне рекомендуется указать ip-адрес (или адреса) компьютера, с которого прикладное решение будет рассылать уведомления. Для тестовых проверок можно указать ip-адрес **0.0.0.0/0**. Сменить список допустимых ip-адресов можно с помощью команды **Edit allowed IPs**.
6. После окончания формирования ключа, он будет отображен в поле **API KEY**. Получившееся значение необходимо использовать в качестве ключа аутентификации для отправки уведомлений (параметр **ДанныеАутентификации** метода **Отправить()** объекта **МенеджерОтправкиДоставляемыхУведомлений**). Рекомендуется относиться к данному ключу как к конфиденциальной информации. Получение данного ключа третьими лицами позволит им отправлять push-уведомления от вашего имени, например в случае, если при создании ключа не указывались конкретные значения ip-адресов.

Получившийся ключ программного интерфейса (**API KEY**) необходимо предоставить внедренцу приложения-отправителя для указания в настройках приложения. Вышеописанные действия не требуются выполнять, если рассылка уведомлений выполняется через сервис фирмы «1С».

Работа с сервисом фирмы «1С»

Разработчик мобильного приложения должен:

- Выполнить регистрацию в сервисе APNs (если он используется мобильным приложением).
- Необходимо зарегистрироваться на сайте сервиса <https://pushnotifications.1c.com/signup>. Для регистрации необходимо иметь действующий номер подвижной беспроводной радиотелефонной связи (мобильный номер). В процессе регистрации на него будет выслан код подтверждения регистрации. В контактных данных рекомендуется указывать данные ответственного лица разработчика мобильного приложения.
- Необходимо зарегистрировать мобильное приложение, которое будет выступать получателем уведомлений. Для этого следует выбрать пункт меню **Доставляемые уведомления – Мобильные приложения** и нажать кнопку **Зарегистрировать приложение**.
- Необходимо подключить созданное мобильное приложение к используемым сервисам (APNs и GCM). Для этого в списке **Ваши мобильные приложения** нажать ссылку **Подключить к APNs и GCM**. В зависимости от сервиса необходимо нажать кнопку **+ Подключить к APNS** или **+ Подключить к GCM**. В зависимости от используемого сервиса, необходимо предоставить следующие данные:
 - Идентификатор мобильного приложения – для всех сервисов.
 - Для APNs: сертификат мобильного приложения в формате **.P12** (и его пароль).

Разработчик отправителя должен:

- Каких-либо действий не выполняется.

Внедренец отправителя должен:

- Необходимо зарегистрироваться на сайте сервиса <https://pushnotifications.1c.com/signup>. Для регистрации необходимо иметь действующий номер подвижной беспроводной радиотелефонной связи (мобильный номер). В процессе регистрации на него будет выслан код подтверждения регистрации. В контактных данных рекомендуется указывать данные администратора информационной системы.
- Зарегистрировать отправителя уведомлений. Для этого следует выбрать пункт меню **Доставляемые уведомления – Отправители уведомлений** и затем нажать кнопку **Зарегистрировать отправителя**. При регистрации рекомендуется указывать список ip-адресов, с которых будет допустимо рассылать уведомления с помощью данного сервиса. Если адреса не указаны – рассылка уведомлений допускается с любого компьютера, который может авторизоваться в сервисе.

Важно понимать, что регистрацию отправителя выполняет не разработчик отправителя, а внедренец отправителя в тот момент, когда выполняется настройка экземпляра системы-отправителя у пользователя.

- В результате регистрации будет сформирован специальный **API KEY отправителя**. Получившееся значение необходимо указать в настройках конкретного экземпляра приложения-отправителя. Оно будет использоваться в качестве ключа аутентификации для отправки уведомлений (параметр **ДанныеАутентификации** метода **Отправить()** объекта **МенеджерОтправкиДоставляемыхУведомлений**). Рекомендуется относиться к данному ключу как к конфиденциальной информации. Получение данного ключа третьими лицами позволит им отправлять push-уведомления от вашего имени, например в случае, если при создании отправителя не указывались конкретные значения ip-адресов.

Пользователь мобильного приложения должен:

- Указать мобильному приложению параметры сервиса обмена, который предоставляет приложение-отправитель.
- Инициировать выполнения команды для создания идентификатора получателя уведомлений и отправку его отправителю уведомлений.

Рассылка уведомлений

Действия по рассылке уведомлений практически не различаются для различных сервисов. Различия наблюдаются в значениях параметров метода отправки уведомлений. Собственно отправка происходит с помощью метода **Отправить()** менеджера доставки отправляемых уведомлений:

- При рассылке уведомлений в качестве данных аутентификации используется (параметр `ДанныеАутентификации` метода `Отправить()`):
 - Сервис APNs – сертификат мобильного приложения, полученный из консоли разработчика. Рекомендуется относиться к данному ключу как к конфиденциальной информации.
 - Сервис GCM – ключ программного интерфейса приложения (`API KEY`), зарегистрированного в консоли разработчика. Рекомендуется относиться к данному ключу как к конфиденциальной информации.
 - Сервис фирмы «1С» – ключ программного интерфейса отправителя (`API KEY отправителя`), который получен при регистрации отправителя на сайте сервиса. Рекомендуется относиться к данному ключу как к конфиденциальной информации.
 - Если для отправки уведомлений используется сервис фирмы «1С», то параметр `ИспользоватьПромежуточныйСервис` должен быть установлен в значение `Истина`.
- Если отправка выполняется одновременно с использованием нескольких сервисов, то в качестве данных аутентификации может выступать соответствие, где ключом выступает вид сервиса (значение перечисления `ТипПодписчикаДоставляемыхУведомлений`), а в качестве значения – те данные, которые необходимы соответствующему сервису.

Следует понимать, что невозможно одновременно (в одном вызове метода `Отправить()`) отправлять уведомления с использованием APNs или GCM и сервиса фирмы «1С». Если у метода `Отправить()` значение параметра `ИспользоватьПромежуточныйСервис` установлено в значение `Истина`, то значением параметра `ДанныеАутентификации` может быть только `API KEY отправителя`, полученный у сервиса фирмы «1С».

- Перечень получателей уведомлений описывается с помощью свойства `Получатели` объекта `ДоставляемоеУведомление`, которое выступает значением первого параметра метода `Отправить()` менеджера доставки отправляемых уведомлений.

Для обеспечения отправки уведомлений с помощью APNs, на компьютере с системой-отправителем должны быть разрешены исходящие подключения по ip-портам [2195](#) и [2196](#).

26.3.6.10.4. Особенности обработки уведомлений мобильным приложением

Если обработчик уведомления не подключен, то уведомление сохраняется в памяти до окончания сеанса и будет передано обработчику уведомления после его подключения (в этом сеансе). Если до окончания сеанса обработчик не был подключен – уведомление теряется после завершения сеанса.

Также необходимо учитывать следующие особенности работы с уведомлениями в мобильном приложении:

- Если мобильное приложение активно, то уведомление сразу доставляется в мобильное приложение, при этом наклейка и звуковое оповещение игнорируются.
- Если мобильное приложение не активно в момент получения уведомления, то уведомление отображается операционной системой. Если пользователь нажал на уведомление, то будет запущено соответствующее приложение. Если пользователь удалил уведомление – приложение не будет проинформировано о существовании уведомления. В ОС iOS (версии 7 и младше) уведомление удаляется из центра уведомлений либо явным действием пользователя, либо вытесняется другими уведомлениями данного приложения.
- Если мобильное приложение, в момент получения уведомления, работает в фоновом режиме и не активно, то работа с уведомлением выполняется по-разному на разных платформах:
 - На платформе iOS уведомление будет передано в мобильное приложение только после того, как его выберет пользователь из панели уведомлений.
 - На платформе Android уведомление сразу передается в мобильное приложение и одновременно отображается в панели уведомлений. После того, как пользователь выбрал уведомление в панели, происходит активизация приложения, но повторно уведомление в приложение не передается.
- В том случае, если в мобильном приложении развернуто несколько информационных баз, то пользователю предлагается переключиться, если в данный момент он работает с другой информационной базой. Если конкретную информационную базу определить не получается – выдается соответствующее диагностическое сообщение.

26.3.7. Отчеты

Формирование отчетов на мобильной платформе возможно двумя разными способами:

- Обработка существующих данных и формирование собственно отчета на встроенном языке;
- Реализация на мобильной платформе некоторой формы, с помощью которой можно задать параметры формирования отчета (отборы и т. д.), а само формирование передать в удаленную систему, которая вернет табличный документ для отображения на мобильном устройстве.

Выбор одного из этих вариантов зависит от некоторого набора факторов:

- Содержит или нет информационная база мобильного приложения все данные, которые необходимы для построения отчета;
- Наличие и качество канала связи между мобильным устройством и удаленной системой;
- Какова сложность отчета;
- Каким образом построен обмен с удаленной системой;
- Каким образом предполагается работа с отчетом;
- И т. д.

Если принято решение формировать отчет на мобильном устройстве, то необходимо помнить об особенностях мобильной платформы (см. [здесь](#)). Для получения данных доступны выборки, а для построения отчета доступно программное формирование табличного документа (на основании макета) и диаграмм (если таковые требуются).

Если для мобильного приложения доступны постоянные высокоскоростные каналы связи с удаленной системой, которое обладает всеми необходимыми данными для формирования требуемого отчета, то возможным вариантом может служить реализация на мобильном приложении

только интерфейсной части отчета (задание отборов, обработка расшифровки и т. д.). Собственно формирование отчета в этом случае будет происходить в удаленном приложении, а возвращаться будет готовый отчет для отображения на устройстве.

При этом возможны и все промежуточные варианты, например, если реализован периодический обмен, можно настройки отчета сформировать на мобильном устройстве и передать на удаленную систему во время сеанса обмена, а обратно получить сформированный отчет для отображения.

Для взаимодействия с удаленным приложением можно использовать Web-сервис (который предоставляет удаленное приложение), обмен данными с помощью плана обмена и т. д.

При необходимости реализации расшифровки отчета, нужно понимать, что мобильное приложение самостоятельно может выполнить только просмотр значений, указанных в качестве расшифровки ячейки отчета. Остальные действия необходимо реализовывать на встроенном языке, включая получение нового варианта отчета с учетом наложенных отборов и т. д.

26.3.8. Работа с картинками

Мобильное приложение работает на различных мобильных устройствах. Одной из отличительных особенностей этих устройств является разрешение и физический размер их (устройств) экранов. Это отличие характеризуется понятием **dpi** (или **ppi**) – количеством точек на дюйм размера экрана. Чем больше значение dpi, тем меньшее физическое место будет занимать картинка с фиксированными физическими размерами.

Чтобы картинка лучше соответствовала различным экранам, существует возможность указывать картинку набором, в котором каждая картинка соответствует определенному интервалу dpi экрана мобильного устройства. При этом мобильная платформа автоматически подбирает нужную картинку (из набора), опираясь на параметры конкретного физического устройства. Прикладной программист не должен предпринимать никаких действий, чтобы была показана нужная картинка.

Набор картинок представляет собой архив zip-формата, который содержит картинки, описание которых приводится в файле манифеста (**manifest.xml**), расположенной в этом же архиве. Картинка может относиться к одному из вариантов разрешения экрана:

- **ldpi** – соответствует экранам с разрешением менее 140 dpi;
- **mdpi** – соответствует экранам с разрешением от 140 до 180 dpi;
- **hdpi** – соответствует экранам с разрешением от 180 до 270 dpi;
- **xdpi** – соответствует экранам с разрешением более 270 dpi.

Имя каждой картинки может быть любым, а соотношение между именем картинки и разрешением указывается в манифесте. Соотношение размеров картинок в архиве можно выразить следующим соотношением: $m=1,5 \cdot l$, $h=2 \cdot l$, $x=3 \cdot l$. Т.е. если размер картинки на панель инструментов равен 16 точкам, то в наборе должны быть картинки следующих размеров: $ldpi=16$, $mdpi=24$, $hdpi=32$, $xdpi=48$ точек.

Также в архиве могут находиться так называемые шаблонные картинки. Это картинки, информативная часть которых присутствует только в альфа-канале модели RGBA. Значения остальных каналов в такой картинке должны формировать белый цвет ($R=255$, $G=255$, $B=255$). Шаблонную картинку рекомендуется формировать в формате PNG (32 бита на пиксель). При отображении такой картинки поле, отведенное под картинку, будет заполняться текущим цветом фона, а цветом текста (с соответствующей интенсивностью) будут отображаться те места картинки, для которых в альфа-канале установлены значения, отличающиеся от полной прозрачности ($A=0$).

В архиве обязательно должна присутствовать картинка для разрешения **ldpi** (**базовая** картинка). Остальные картинки являются необязательными. В случае отсутствия нужной картинки, система будет использовать ближайшую подходящую.

Архив с картинками может быть помещен:

- в библиотеку картинок;
- в реквизиты объектов конфигурации, имеющие тип **Картинка**;
- в объект **Картинка**, **ХранилищеЗначения**, **ДвоичныеДанные**, созданный во встроенном языке.

При создании архива рекомендуется использовать картинки формата **.png**. Если в архив помещаются картинки, которые используют сжатие (**png**, **jpg** и т. д.), то не рекомендуется создавать архив со сжатием. В противном случае рекомендуется создавать архив со сжатием.

Базовая картинка используется в следующих случаях:

- в прикладном решении, работающим под управлением «1С:Предприятия» для персонального компьютера;
- при экспорте картинки в другие форматы данных;
- при передаче данных с сервера на клиент.

Для устройств на ОС iOS используются следующие картинки из набора:

- обычные экраны – картинка **mdpi**;
- экраны Retina – картинки **xdpi**.

Для устройств на ОС Android используются картинки, соответствующие разрешению экрана используемого устройства.

Манифест (**manifest.xml**) должен соответствовать следующей схеме:

```
<xs:complexType name="PictureVariant">
  <xs:sequence>
  </xs:sequence>
  <xs:attribute name="name" type="xs:Name" use="required"/>
  <xs:attribute name="screenDensity" type="tns:PictureVariantScreenDensity" use="required"/>
  <xs:attribute name="isTemplate" type="xs:boolean" default="false"/>
</xs:complexType>
<xs:simpleType name="PictureVariantScreenDensity">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ldpi"/>
    <xs:enumeration value="mdpi"/>
  </xs:restriction>
</xs:simpleType>
```

```

<xs:enumeration value="hdpi"/>
<xs:enumeration value="xdpi"/>
</xs:restriction>
</xs:simpleType>

```

Если картинка с вариантами создавалась в конфигураторе, то при ее выгрузке в файл, манифест может иметь такой вид:

[Копировать в буфер обмена](#)

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Picture>
  <PictureVariant name="h.png" screenDensity="hdpi"/>
  <PictureVariant name="l.png" screenDensity="ldpi"/>
  <PictureVariant name="m.png" screenDensity="mdpi"/>
  <PictureVariant name="x.png" screenDensity="xdpi"/>
  <PictureVariant name="lc.png" screenDensity="ldpi" isTemplate="true"/>
</Picture>

```

26.3.9. Работа с ролями и пользователями

При работе с ролями в мобильном приложении следует помнить, что из всех прав доступа, которые предоставляет платформа для персонального компьютера, на мобильной платформе доступен только ограниченный набор. Этот набор прав призван обеспечить функционирование различных механизмов, обеспечивающих интерфейс мобильного приложения. Права, которые могут использоваться на мобильной платформе, отмечены символом «*» в полном списке прав (см. [здесь](#)).

Также имеется возможность выполнять ролевую настройку формы (см. [здесь](#)), настраивать ролевую видимость командного интерфейса (см. [здесь](#)).

ВНИМАНИЕ! Права для доступа к данным ([Чтение](#), [Добавление](#), [Изменение](#), [Удаление](#) и т. д.) не поддерживаются мобильной платформой.

Методы управления привилегированным режимом в мобильной платформе используются для совместимости с платформой для персонального компьютера. Сами вызовы методов работы с привилегированным режимом мобильной платформой игнорируются и никаких действий мобильная платформа не выполняет.

При определении прав пользователя действует стандартная схема: если в информационной базе не заданы пользователи, то для определения прав доступа используется свойство конфигурации [ОсновныеРоли](#). Если для доступа к данным мобильного приложения используется пользователь, то права доступа определяются по составу ролей этого пользователя.

Мобильная платформа имеет ограниченный набор средств работы с пользователями информационной базы:

- В информационной базе может существовать только один пользователь. Попытка создать более одного пользователя приводит к генерации исключения.
- Пользователь может быть создан только с помощью программного кода. Интерактивное создание пользователей в конфигураторе не приведет к передаче пользователя на мобильное устройство.
- Мобильная платформа не предлагает средств аутентификации при старте мобильного приложения. Если пользователь указан в списке пользователей, то этот пользователь автоматически будет установлен в качестве пользователя текущего сеанса.

В результате можно рекомендовать следующую схему работы от лица пользователей с разными правами:

1. Выполняется создание пользователя с нужным набором ролей или выполняется изменение набора ролей у существующего пользователя.
2. Выполняется перезапуск мобильного приложения с помощью конструкции `ЗавершитьРаботуСистемы(Ложь, Истина);`.

26.3.10. Полнотекстовый поиск

Описание особенностей использования полнотекстового поиска в мобильной платформе см. [здесь](#).

26.4. Взаимодействие с мобильным устройством при разработке мобильного приложения

Помещение мобильного приложения на мобильное устройство возможно двумя способами:

- Через фирменный магазин приложений App Store (<http://itunes.apple.com/ru/genre/ios>) или Google Play (<http://play.google.com>). Данный способ используется для распространения публичных версий мобильных приложений. Описание подготовки прикладного решения к публикации см. [здесь](#).

Подготовленное к публикации прикладное решение одобряется и публикуется по правилам соответствующего магазина приложений. Подробности этого процесса в данной книге не рассматриваются.

- С помощью версии мобильной платформы для разработчика. Используется для разработки мобильных приложений. Далее будет рассмотрен этот способ.

Для того чтобы начать использовать мобильное приложение, необходимо выполнить некоторые действия:

- На мобильное устройство должна быть установлена мобильная платформа разработчика;
- Мобильное приложение опубликовано на веб-сервере;
- На мобильном устройстве создана информационная база (приложение) на основе опубликованного мобильного приложения.

После выполнения этих операций получается готовая инфраструктура для разработки и исполнения мобильного приложения на мобильном устройстве. При этом система может быть настроена таким образом, что обновление конфигурации будет приводить к тому, что мобильная платформа для разработчика автоматически обновит конфигурацию на мобильном устройстве.

Далее будет более подробно рассмотрено каждое действие.

26.4.1. Установка мобильной платформы разработчика

26.4.1.1. Для ОС iOS

Для разработки мобильного приложения для ОС iOS необходимо выполнить следующие требования:

- Работа с мобильным устройством возможна только с компьютера фирмы Apple (далее – Mac) с установленной операционной системой MacOS X 10.8 и выше.
- На компьютере Mac должен быть один свободный USB-порт, используемый для связи с мобильным устройством.
- На этом компьютере необходимо установить средство разработки Xcode версии 4.5 и выше (<https://developer.apple.com/xcode/>).
- Необходимо зарегистрироваться в программе iOS Developer Program (<https://developer.apple.com/programs/ios/>).
- Для работы необходимо мобильное устройство на iOS, соответствующее системным требованиям.

Для установки мобильной платформы разработчика на мобильное устройство, работающее под управлением iOS, необходимо выполнить следующие действия:

- Подключить мобильное устройство к компьютеру Mac;
- Распаковать файл с проектом мобильной платформы [prjios.zip](#) на компьютере Mac;
- Открыть проект мобильной платформы разработчика в системе Xcode (двойным щелчком мыши по файлу [1cem.xcodeproj](#) или выбрать файл [1cem.xcodeproj](#) с помощью команды **File – Open** в системе Xcode);
- Открыть инструмент **Organizer** с помощью команды меню Xcode **Window – Organizer**;
- В открывшемся окне, в панели слева выбрать подключенный телефон (отмечен зеленым маркером);
- В панели снизу выбрать команду **Add to Portal**;
- Средство **Organizer** запросит необходимые для телефона сертификаты и provisioning profile с сайта Apple и установит их на телефон автоматически. Потребуется ввести пароль разработчика, полученный при регистрации в iOS Developer Program.
- С помощью команды меню Xcode **Product – Edit Scheme...** необходимо открыть диалог настроек текущей схемы запуска. В открывшемся диалоге:
 - В свойстве **Scheme** выбрать значение **1cem**;
 - В свойстве **Destination** выбрать подключенный телефон;
 - В свойстве **Build configuration** выбрать значение **Release**;
 - Нажать кнопку **OK**.
- Запустить проект с помощью команды меню Xcode **Product – Run**.
- Мобильная платформа будет передана на мобильное устройство, запущена и ее иконка появится в списке установленных приложений мобильного устройства.
- Мобильное устройство можно отключить от компьютера Mac и далее запускать мобильную платформу нажатием на иконку «1С:Предприятие» в списке приложений мобильного устройства. Если в момент отключения на мобильном устройстве была запущена мобильная платформа разработчика, которая запущена с помощью Xcode, то она будет остановлена.

26.4.1.2. Для ОС Android

Для разработки мобильного приложения для ОС Android необходимо выполнить следующие требования:

- Установить на компьютере Android SDK (<http://developer.android.com/sdk/index.html>). Минимальные требования к Android SDK следующие:
 - Версия Android SDK Tools – не ниже 20.0.3;
 - Версия Android SDK Platform-tools – не ниже 14;
 - Версия SDK Platform – API 17.
- На компьютере должен быть один свободный USB-порт, используемый для связи с мобильным устройством.
- Для работы необходимо мобильное устройство на Android, соответствующее системным требованиям.

Для установки мобильной платформы разработчика на мобильное устройство, работающее под управлением Android, необходимо выполнить следующие действия:

- В настройках мобильного устройства необходимо установить следующие флажки:
 - **Неизвестные источники**;
 - **Отладка через USB**;
- Скачать с сайта производителя и установить на компьютер USB-драйвер мобильного устройства;
- Подключить мобильное устройство к компьютеру;
- С помощью интерпретатора командной строки ОС Windows выполнить команду:

Пример для ARM-платформы:

```
%ANDROID_SDK%\platform-tools\adb.exe install -r "<Каталог платформы>\1cem-arm.apk"
```

```
%ANDROID_SDK%\platform-tools\adb.exe install -r "<Каталог платформы>\1сем-x86.apk"
```

Где, [Каталог платформы](#) – каталог, в котором расположен соответствующий файл с мобильной платформой. Если мобильная платформа уже установлена на мобильное устройство, то данная команда выполнит ее переустановку.

- Мобильная платформа будет передана на мобильное устройство и ее иконка появится в списке установленных приложений.
- Мобильное устройство можно отключить от компьютера и далее запускать мобильную платформу нажатием на иконку «1С:Предприятие» в списке приложений мобильного устройства.

Для установки мобильной платформы разработчика можно воспользоваться командой конфигуратора [Главное меню – Конфигурация – Мобильное приложение – Использование Android debug bridge – Установить мобильную платформу](#). В этом случае надо будет выполнить дополнительную настройку конфигуратора (подробнее см. [здесь](#)). С установкой мобильной платформы можно выполнить или на физическое мобильное устройство, подключенное к компьютеру или на эмулятор устройства, созданный с помощью менеджера виртуальных устройств Android (входит в состав Android SDK).

26.4.2. Публикация мобильного приложения на веб-сервере

Чтобы мобильное приложение было доступно мобильной платформе, необходимо выполнить публикацию мобильного приложения на веб-сервере. Фактически, публикация делится на две части:

- Подготовка инфраструктуры для работы и начальная выгрузка конфигурации мобильного приложения – выполняется один раз.
- Обновление конфигурации мобильного приложения в подготовленной инфраструктуре.

Для подготовки инфраструктуры служит команда [Конфигурация – Мобильное приложение – Опубликовать...](#) В открывшемся диалоге следует указать флажок [Создавать виртуальный каталог на веб-сервере](#). Если флажок не установлен, то будут недоступны реквизиты [Имя](#) и [Каталог](#), а также при нажатии кнопки [Опубликовать](#) не будет выполняться создание виртуального каталога на веб-сервере.

ВНИМАНИЕ! Для выполнения публикации требуются административные права.

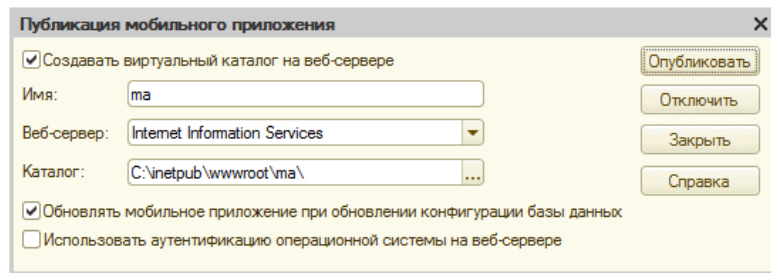


Рис. 355. Публикация мобильного приложения

Затем указать свойства виртуального каталога веб-сервера:

- [Имя](#) – определяет имя виртуального каталога (будет участвовать в URL конфигурации мобильного приложения для указания на мобильном устройстве при создании информационной базы).
- [Веб-сервер](#) – определяет используемый веб-сервер. Если используется веб-сервер Microsoft Internet Information Services, то становится возможным указание флажка [Использовать аутентификацию операционной системы на веб-сервере](#).
- [Каталог](#) – указывает физический каталог на диске, в котором будет находиться файл конфигурации мобильного приложения в формате [.xml](#) и куда будет отображен виртуальный каталог веб-сервера.

Флажок [Обновлять мобильное приложение при обновлении конфигурации базы данных](#) управляет возможностью автоматического обновления конфигурации мобильного приложения при обновлении конфигурации базы данных. Если флажок установлен – обновление будет выполняться.

Операцию обновления конфигурации также можно выполнить принудительно, с помощью команды [Конфигурация – Мобильное приложение – Обновить публикуемое приложение](#). Если к моменту вызова данной команды публикация мобильного приложения не выполнялась – будет открыт диалог публикации, описанный выше.

ВНИМАНИЕ! На веб-сервере публикуется конфигурация базы данных, а не редактируемая конфигурация.

Для выполнения собственно публикации необходимо нажать кнопку [Опубликовать](#). При этом выполняются следующие действия:

- Если установлен флажок [Создавать виртуальный каталог на веб-сервере](#), то создается виртуальный каталог на веб-сервере;
- Создается каталог на диске, в который отображается виртуальный каталог;
- Предлагается выгрузить конфигурацию мобильного приложения в каталог (указанный в реквизите [Каталог](#) диалога публикации);
- Проверяется, соответствует конфигурация базы данных редактируемой конфигурации или нет;
- Если конфигурация информационной базы не соответствует редактируемой конфигурации – предлагается выполнить обновление конфигурации базы данных. Однако эту операцию можно не выполнять, если требуется опубликовать именно конфигурацию базы данных;
- Выполняется проверка мобильного приложения (см. [здесь](#));
- Если ошибок нет, то выполняется выгрузка конфигурации информационной базы в файл, в противном случае выгрузка не производится.

Файл с конфигурацией мобильного приложения, выгруженный с помощью команды [Публиковать...](#), имеет фиксированное имя [1сема.xml](#). Поиск (и обновление) именно этого файла выполняется при выполнении команды [Обновить публикуемое приложение](#).

СОВЕТ. При создании виртуального каталога, файл конфигурации (*1cema.xml*) устанавливается в качестве его (каталога) страницы по умолчанию. Это позволяет в диалоге создания информационной базы на мобильном устройстве указывать URL в сокращенном виде.

Кнопка [Отключить](#) диалога публикации мобильного приложения выполняет отмену публикации: удаляет виртуальный каталог на веб-сервере и физический каталог.

26.4.3. Запуск и перезапуск приложения на мобильной платформе

26.4.3.1. Общая информация

Запуск мобильного приложения осуществляется непосредственно на мобильном устройстве или из конфигуратора. На устройстве должна быть установлена мобильная платформа разработчика.

Запуск из конфигуратора возможен при одновременном выполнении двух условий: разработка выполняется с использованием ОС Android и выполнена соответствующая настройка конфигуратора (см. [здесь](#)).

26.4.3.2. Перезапуск мобильного приложения, опубликованного на веб-сервере

Если мобильное приложение запущено с помощью мобильной платформы разработчика и для приложения установлен флажок [Перезапуск из конфигуратора](#) (см. [здесь](#)), то существует возможность перезапускать мобильное приложение из конфигуратора. Это можно сделать с помощью команды главного меню [Отладка – Начало отладки – Мобильное приложение: запустить](#).

Если конфигурация была модифицирована (были произведены изменения), то конфигуратор выводит вопрос: [Редактируемая конфигурация отличается от конфигурации базы данных. Произвести обновление конфигурации базы данных?](#) Для сохранения внесенных изменений следует выбрать кнопку [Да](#).

Если выбрана кнопка [Нет](#), то мобильное приложение будет перезапущено без обновления конфигурации.

Если при публикации мобильного приложения был установлен флажок [Обновлять мобильное приложение при обновлении конфигурации базы данных](#), то будет выполнено автоматическое обновление публикации конфигурации мобильного приложения.

Мобильное приложение выполняет проверку обновления при открытии в том случае, если установлен флажок [Перезапуск из конфигуратора](#) в свойствах мобильного приложения на мобильном устройстве. Если этот флажок сброшен, то при открытии наличие обновлений не проверяется. Однако такую проверку можно выполнить вручную (см. [здесь](#)).

26.4.3.3. Запуск и перезапуск мобильного приложения при использовании Android debug bridge

Запуск и перезапуск мобильного приложения можно выполнять с помощью конфигуратора. Это можно сделать с помощью команды главного меню [Отладка – Начало отладки – Мобильное приложение: запустить](#). Чтобы при этом использовался Android debug bridge, конфигуратора должен быть соответствующим образом настроен (см. [здесь](#)). Кроме того, конфигурация должна быть опубликована в файловой системе (публикация на веб-сервере в этом случае не является обязательной) (см. [здесь](#)).

Если конфигурация была модифицирована (были произведены изменения), то конфигуратор выводит вопрос: [Редактируемая конфигурация отличается от конфигурации базы данных. Произвести обновление конфигурации базы данных?](#) Для сохранения внесенных изменений следует выбрать кнопку [Да](#).

Если выбрана кнопка [Нет](#), то мобильное приложение будет перезапущено без обновления конфигурации.

Если при публикации мобильного приложения был установлен флажок [Обновлять мобильное приложение при обновлении конфигурации базы данных](#), то будет выполнено автоматическое обновление публикации конфигурации мобильного приложения.

После перезапуска мобильной платформы разработчика будет обнаружена новая конфигурация и будет выдано предложение выполнить обновление.

26.4.4. Управление приложениями на мобильной платформе разработчика

26.4.4.1. Создание приложения

Для того чтобы создать приложение на мобильной платформе разработчика, следует:

- Запустить «1С:Предприятие» на мобильном устройстве;
- Выбрать команду добавления приложения;
- В открывшемся окне ввести URL веб-сервера, на котором опубликовано мобильное приложение (см. [здесь](#)) в поле [Адрес](#);
- При необходимости, следует указать имя пользователя и пароль для авторизации на веб-сервере, на котором опубликовано мобильное приложение и нажмите кнопку [Загрузить](#);
- Затем следует указать имя приложения и нажать кнопку [Готово](#);
- После закрытия окна будет создано приложение.

Список приложений будет пропущен, и будет сразу запущено приложение, если оно в списке единственное. В этом случае для перехода в список приложений следует выбрать команду [Список приложений](#) в главном меню приложения.

26.4.4.2. Изменение свойств приложения

Свойства приложения изменяются в специальном окне. В зависимости от операционной системы, доступ к этой команде выполняется по-разному:

- Для ОС iOS: в правой части строки с именем нужного приложения нажмите картинку с изображением знака больше.
- Для ОС Android: выполните долгое нажатие на нужном приложении. В открывшемся контекстном меню выберите команду [Изменить](#).

В открывшемся окне можно изменять свойства приложения, которые указывались при его создании: наименование приложения, URL расположения конфигурации, имя и пароль пользователя для доступа к веб-серверу, на котором располагается конфигурация мобильного приложения.

Также в этом окне можно выполнить следующие действия:

- Запустить мобильное приложение. Для этого необходимо нажать кнопку [Открыть](#).
- Проверить наличие новой версии конфигурации мобильного приложения на веб-сервере. Для этого необходимо нажать кнопку [Проверить обновления](#). В случае наличия новой версии конфигурации будет предложено выполнить обновление мобильного приложения.
- Удалить приложение с помощью кнопки [Удалить](#).

ВНИМАНИЕ! После удаления приложения, данные информационной базы восстановить невозможно.

- Настроить возможность автоматического обновления и перезапуска мобильного приложения в случае, если на веб-сервере будет обнаружена новая версия конфигурации (см. [здесь](#)). Для этого следует изменить состояние флажка [Перезапуск из конфигууратора](#).

Список приложений будет пропущен, и будет сразу запущено приложение, если оно в списке единственное. В этом случае для перехода в список приложений следует выбрать команду [Список приложений](#) в главном меню приложения.

26.4.4.3. Удаление приложения

Для удаления приложения следует выбрать команду [Удалить](#) и подтвердить свое действие. В зависимости от операционной системы, доступ к этой команде выполняется по-разному:

- Для ОС iOS: нажмите кнопку [Изменить](#), затем нажмите картинку в левой части строки с именем удаляемого приложения. Затем в правой части этой же строки нажмите кнопку [Удалить](#).
- Для ОС Android: выполните длинное нажатие на удаляемом приложении. В открывшемся контекстном меню выберите команду [Удалить](#).

Список приложений будет пропущен, и будет сразу запущено приложение, если оно в списке единственное. В этом случае для перехода в список приложений следует выбрать команду [Список приложений](#) в главном меню приложения.

ВНИМАНИЕ! После удаления приложения, данные информационной базы восстановить невозможно.

26.4.5. Сборка мобильного приложения перед публикацией в магазине приложений

Мобильное приложение, предназначенное для публикации в магазине приложений, состоит из мобильной платформы и мобильного приложения и дополнительных файлов (иконки, логотипы и т. д.), «упакованные» в один контейнер. Этот файл можно установить на мобильное устройство.

Для формирования мобильного приложения предназначена специальная конфигурация ([Сборщик мобильных приложений](#)), которая входит в состав дистрибутива мобильной платформы. Конфигурация предоставляет следующие возможности:

- Хранить разные версии мобильных конфигураций;
- Хранить разные пакеты графической информации (иконки и заставки);
- Хранить разные версии мобильной платформы. Сборщик мобильных приложений работает с мобильной платформой версии 8.3.4 и старше. Работа с предыдущими версиями мобильной платформы не поддерживается;
- Собирать и хранить собственно мобильные приложения;
- Сборка для ОС Android выполняется как для архитектуры ARM, так и для архитектуры x86;
- При наличии компьютера Mac, подключенного к сети, в которой развернут сборщик мобильных приложений, имеется возможность выполнять полный цикл сборки мобильного приложения для ОС iOS – с автоматически созданием пакета приложения (.ipa);

Результатом работы сборщика выступают следующие объекты:

1. Мобильное приложение для ОС Android (архитектура ARM);
2. Мобильное приложение для ОС Android (архитектура x86);
3. Проект мобильного приложения для сборки на компьютере Apple (.zip-файл);
4. Пакет приложения для ОС iOS (.ipa-файл). Файл с пакетом приложения формируется в том случае, если в сборщике мобильных приложений настроены параметры соединения с компьютером Mac, который может выполнять сборку пакета приложений.

Для того чтобы сборщик мобильных приложений мог выполнять какие-либо действия на стороне компьютера Mac, его необходимо настроить. Для этого необходимо выполнить следующие действия:

- Установить все доступные обновления ОС Mac OS X;
- Установить программу Xcode. Получить пакет можно здесь: <https://developer.apple.com/xcode/>;
- Скачать и установить в связку ключей Система сертификат Apple Worldwide Developer Relations (<http://developer.apple.com/certificationauthority/AppleWWDRCA.cer>);
- С помощью средств портала разработчика Apple создать и установить в связку ключей Система сертификат дистрибуции для Apple AppStore или для корпоративного распространения:
 - Для этого потребуется создание личного ключа средствами связки ключей;
 - Следует обратить внимание на то, что все личные ключи и сертификаты должны располагаться в связке Система;
 - После создания личного ключа, на вкладке [Доступ](#) диалога свойств ключа установить настройку [Разрешить всем программам получать доступ к этому объекту](#).
- С помощью вкладки [Accounts](#) настроек Xcode установить provisioning profile, соответствующий сертификату дистрибуции;

- В настройках **Общий доступ**, системных настроек компьютера Mac, необходимо разрешить режим **Удаленный вход**.
- Здесь же следует получить имя компьютера, которое затем указать в настройках сборщика мобильного приложения.
- Также следует убедиться, что ip-порт номер 22 (SSH) компьютера Mac доступен для компьютера, на котором функционирует серверная часть сборщика мобильных приложений.

Сборка мобильного приложения выполняется на стороне сервера. На компьютере (или компьютерах), на котором исполняется серверная часть «1С:Предприятия», должно быть установлено следующее программное обеспечение:

1. Пакет Android SDK. Пакет можно скачать отсюда: <http://developer.android.com/sdk/index.html>.
2. Пакет Java SDK. Пакет можно скачать отсюда: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.
3. Пакет Apache Ant. Пакет можно скачать отсюда: <http://ant.apache.org/bindownload.cgi>.
4. Пакет PuTTY (используются утилиты **pscp** и **plink**). Пакет можно скачать отсюда: <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>.

Также сборщик мобильных приложений содержит встроенную документацию (справка), которая содержит описание правил работы со сборщиком, системные требования и форматы архивов, которые будут загружаться в прикладное решение.

26.5. Публикация в магазине приложений

Подготовка приложения для размещения в магазине приложений делается аналогично проверке мобильного приложения перед публикацией в магазине приложений.

26.5.1. Для ОС iOS

Перед отправкой приложения в Apple AppStore необходимо создать приложение в портале iTunes Connect и заполнить всю требуемую информацию о приложении. Затем следует нажать кнопку **Ready to Upload Binary**, при этом статус вашего приложения в iTunes Connect должен измениться на **Waiting For Upload**.

26.5.1.1. Из архива проекта

После формирования файла с архивом приложения для ОС iOS следует скопировать файл архива на компьютер Mac и выполнить следующие действия:

- Распаковать файл с проектом мобильного приложения **<ИдентификаторПриложения>.zip** на компьютере Mac;
- Открыть проект мобильной платформы разработчика в системе Xcode (двойным щелчком мыши по файлу **1csm.xcodeproj** или выбрать файл **1csm.xcodeproj** с помощью команды **File – Open** в системе Xcode);
- С помощью команды меню Xcode **Product – Edit Scheme...** необходимо открыть диалог настроек текущей схемы запуска. В открывшемся диалоге:
 - В свойстве **Scheme** выбрать значение **1csm**;
 - В свойстве **Build configuration** выбрать значение **Release**;
 - Нажать кнопку **OK**.
- Создать архив приложения с помощью команды Xcode **Product – Archive**;
- Открыть инструмент **Organizer** с помощью команды меню Xcode **Window – Organizer**;
- В верхней панели переключится на архивы **Archives**;
- Выбрать созданный архив и нажать кнопку **Distribute...**;
- Следовать указаниям помощника.

26.5.1.2. Бинарный файл (.ipa)

Вручную

Отправка подготовленного пакета приложения (.ipa) в Apple App Store выполняется с помощью утилиты **Application Loader**, входящей в комплект средств разработки Xcode. Для запуска утилиты загрузчика приложений необходимо сначала запустить Xcode, затем выбрать меню **Xcode – Open Developer Tool – Application Loader**.

В загрузчике необходимо ввести данные вашей учетной записи в программе для разработчиков Apple. Затем необходимо выбрать **Deliver Your App**. В открывшемся диалоге выбрать приложение, которое вы хотите отправить. Если его нет в списке, значит в iTunes Connect для этого приложения не заполнена необходимая информация. Далее необходимо нажать **Next**, затем **Choose**, и выбрать требуемый .ipa файл. Для отправки файла необходимо нажать **Send** и дождаться завершения процесса.

С помощью сборщика мобильных приложений

Необходимо открыть элемент списка мобильных приложений, для которого необходимо выполнить загрузку приложения, и выбрать команду **Загрузить в AppStore**. Сборщик выполнит загрузку, используя

имя пользователя и пароль доступа к iTunes Connect, который можно указать в диалоге настройки параметров поставщика на закладке **Параметры для ОС iOS** (если настройку выполняет администратор сборщика) или в диалоге **Сервис – Параметры доступа к iTunes Connect** (если публикацию может выполнять обычный пользователь).

Публикация выполняется с помощью компьютер Mac, на котором выполнялась сборка мобильного приложения.

26.5.2. Для ОС Android

Для выкладки приложения в Google Play необходимо создать приложение на портале Google Play, заполнить требуемую информацию о приложении и загрузить [.apk](#)-файл(ы).

При создании приложения следует выбрать [Prepare Store Listing](#). После заполнения информации о приложении необходимо перейти на закладку [APK](#) и переключиться в расширенный режим, нажав кнопку [Switch to advanced mode](#). Далее, для каждой архитектуры процессора (ARM и x86) следует нажать [Upload new APK to Production](#), и выбрать соответствующий файл приложения. Только расширенный режим позволяет загружать файлы приложений для различных архитектур процессоров. В результате оба приложения появятся в списке с соответствующими архитектурами x86 и armeabi. Теперь необходимо нажать кнопку [Publish now](#).

Глава 27. Инструменты разработки

27.1. Редактор формы

27.1.1. Общая схема

Редактор формы позволяет выполнять все действия, которые возникают при создании и изменении формы. Редактор формы представляет собой группу из нескольких связанных между собой редакторов:

- реквизитов – закладка [Реквизиты](#);
- команд – закладка [Команды](#);
- элементов – закладка [Элементы](#);
- параметров – закладка [Параметры](#);
- модуля – закладка [Модуль](#);
- командного интерфейса – закладка [Командный интерфейс](#).

В нижней части окна расположена форма в режиме предварительного просмотра.

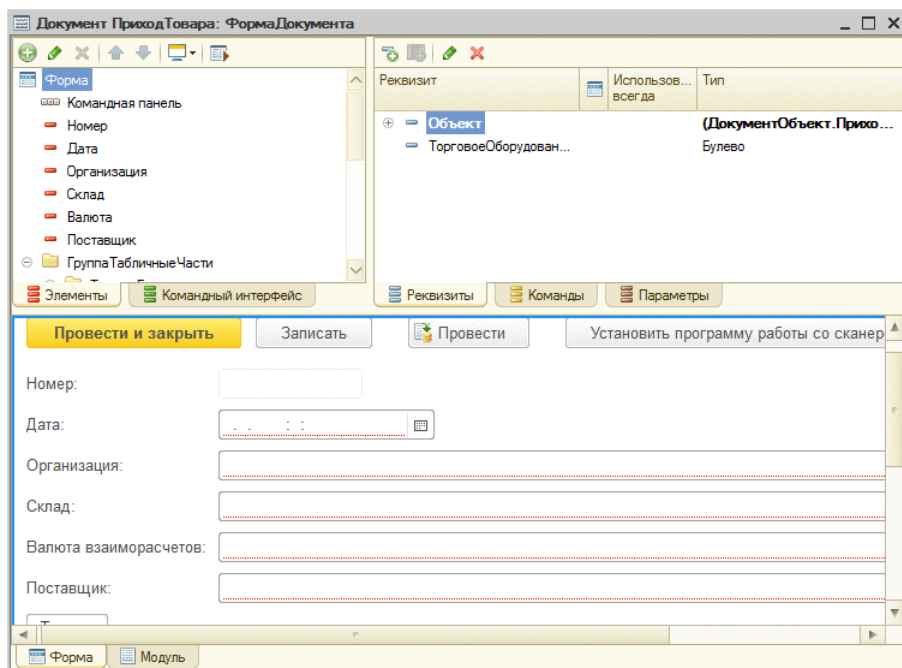
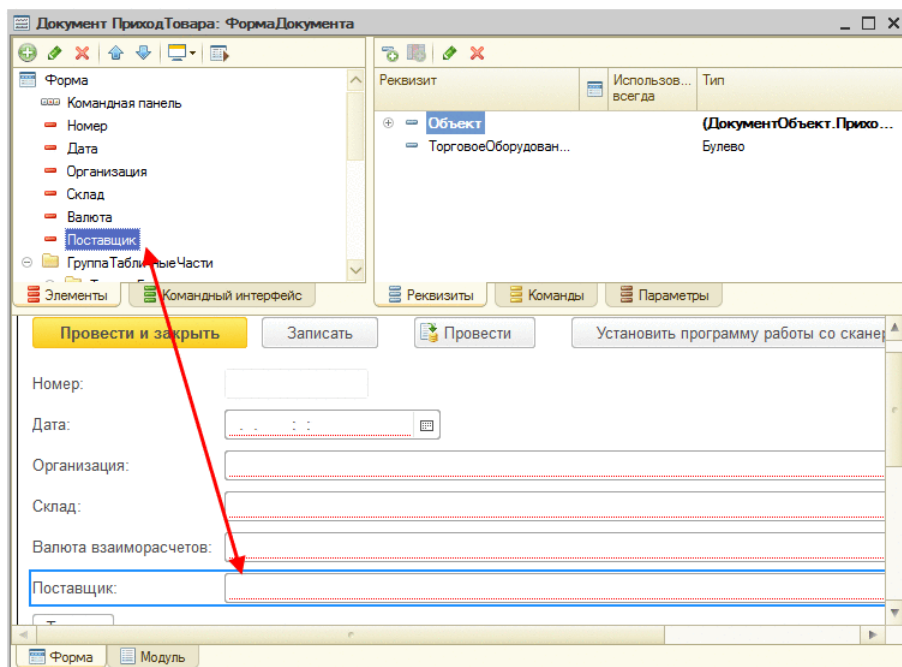


Рис. 356. Редактор формы

В редакторах доступно множественное выделение. В этом случае в палитре свойств будут отражены только те свойства, которые присутствуют у всех выделенных объектов. Изменения, которые сделаны в палитре свойств, будут применены для всех выделенных объектов.

Добавление элементов формы обычно осуществляется перетаскиванием реквизита формы на панель элементов. При этом имя элемента формы становится равным имени реквизита, а в качестве данных элемента формы установлен путь к реквизиту. В дальнейшем имя элемента можно изменить. Система автоматически определяет как элемент формы, с помощью которого будет отображаться реквизит, так и вид этого элемента, если такое возможно. Аналогичным образом выполняется добавление на форму команд (как команд формы, так и глобальных команд). Если реквизит формы или команда размещена на форме, то справа от названия у нее выводится специальный маркер серого цвета.

При выборе элемента формы на закладке [Элементы](#) происходит автоматическое позиционирование на выбранный элемент в окне предварительного просмотра. Если элемент выделяется в окне предварительного просмотра, то происходит автоматическое выделение этого элемента в дереве элементов формы на закладке [Элементы](#).



Таким образом, можно выделить элементы формы и в редакторе элементов и в окне предварительного просмотра. При этом возможности, доступные в окне предварительного просмотра, практически совпадают с таковыми в списке элементов. В частности, доступны контекстные меню элементов, открытие окна свойств, перетаскивание элементов и т. д. Если выделить элемент в окне предварительного просмотра, вокруг элемента возникает рамка, которая в зависимости от элемента может быть нескольких цветов.

Цвет рамки	Описание
Синяя	Выделение обычного элемента
Оранжевая	Выделение автоматически созданных кнопок и подменю командных панелей. Служит для индикации того, что для данного элемента невозможно выполнить никаких настроек (свойства, положение)
Серая	Выделение родительского элемента, если выбранный элемент невозможно выделить в окне предварительного просмотра. Например, если в списке элементов выбрать контекстное меню поля ввода, то поле ввода будет выделено в окне предварительного просмотра серым цветом
Зеленая	Используется при перетаскивании элементов и отображает элемент, в который выполняется перетаскивание (например, таблицу или группу формы), и индикатор того, где будет размещен перетаскиваемый элемент

Щелчок мышью по заголовку окна в окне предварительного просмотра эквивалентен выбору элемента **Форма** (на закладке **Элементы**). Контекстное меню, которое возникает в этом случае, также эквивалентно контекстному меню, соответствующему элементу **Форма**.

Если необходимо выделить родительский элемент (например, группу, в которую входит поле) для какого-либо другого элемента, это можно сделать двумя последовательными щелчками левой кнопкой мыши на подчиненном элементе. Такое выделение сработает, если интервал между щелчками составляет от 1 до 2 секунд. При меньшем времени система может воспринять щелчки как двойной щелчок, при большем – как отдельные действия. Если в момент первого щелчка было выделено несколько элементов, то выделение родительского элемента не будет выполнено.

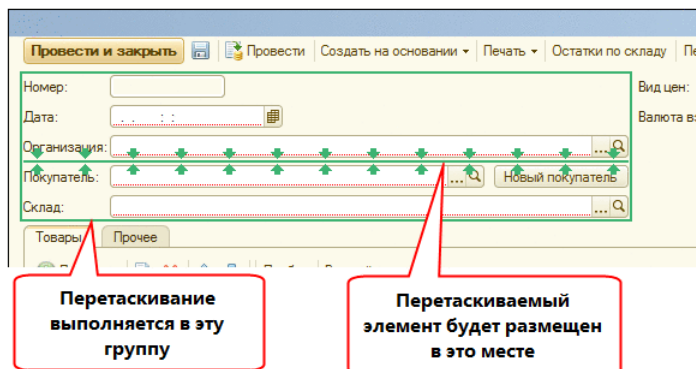


Рис. 358. Индикация перетаскивания

Для быстрого перехода к связанным объектам существует команда **Перейти**. Эта команда расположена в контекстном меню объекта, в редакторах элементов, фрагментов командного интерфейса, реквизитов, команд и окне предварительного просмотра. С помощью этой команды можно, например, быстро перейти от элемента формы (выделенного в списке элементов или окне предварительного просмотра) к реквизиту формы, который отображается этим элементом. Если доступно несколько объектов, к которым возможен переход, будет открыто окно с предложением выбрать нужный объект.

Также в редакторе формы возможно быстрое создание (или переход) обработчиков событий того или иного элемента формы с помощью подменю **События** контекстного меню элемента.

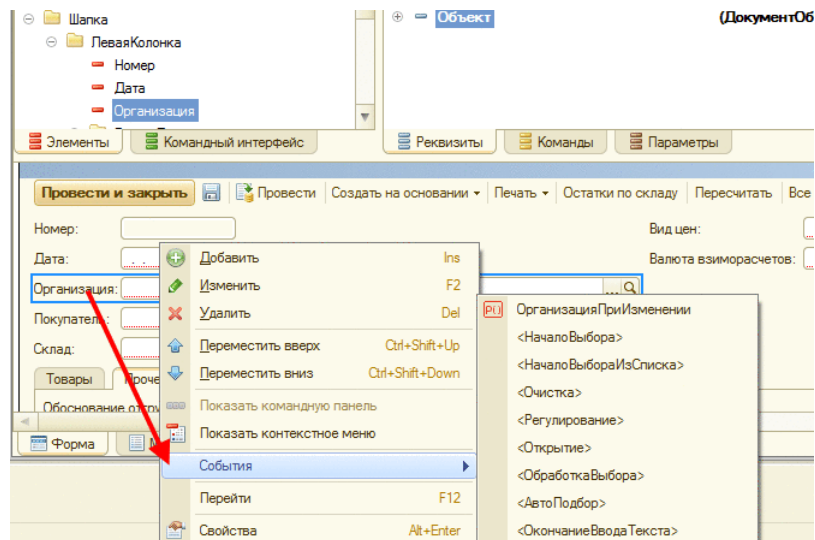


Рис. 359. Меню «События»

В состав этого меню входят все события, которые доступны для данного элемента формы в окне свойств этого элемента. Если в меню указано имя события в угловых скобках, это значит, что обработчик события не определен (например, для события **Очистка** элемента **Организация** на [рис. 359](#)). Для создания обработчика следует выбрать нужное событие – произойдет переход в редактор модуля формы с созданием необходимого обработчика.

Если обработчик для события уже создан, то выбор соответствующего пункта в подменю **События** вызовет переход к существующему обработчику в редакторе формы (например, обработчик **ОрганизацияПриИзменении()** на [рис. 359](#)).

Для кнопки подменю **События** не формируется, а команда перехода к обработчику соответствующей команды расположена непосредственно в контекстном меню элемента.

Если выполняется создание клиентского обработчика, то система предлагает на выбор несколько вариантов его создания:

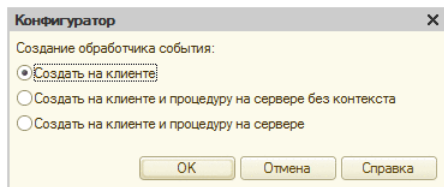


Рис. 360. Создание клиентского обработчика

Первый вариант (**Создать на клиенте**) предполагает создание только клиентского обработчика событий. Вариант **Создать на клиенте и процедуру на сервере без контекста** предполагает создание клиентского обработчика события, внеконтекстной серверной процедуры и вызов процедуры из клиентского обработчика. Последний вариант отличается от предыдущего тем, что серверная процедура создается контекстной. Если в каком либо клиентском обработчике события не рекомендуется использовать контекстные серверные вызовы (информация об этом доступа в Синтакс-помощнике), то использование последнего варианта (с контекстной серверной процедурой) недоступно. Удаление обработчика события не приводит к удалению серверной процедуры (если при создании обработчика события она создавалась).

При удалении реквизита (из списка реквизитов формы) на экран выводится вопрос **Удалить связанные элементы?** Если ответ отрицательный, то у связанных элементов формы очищается свойство **Данные**. При положительном ответе выполняется удаление элементов формы, которые связаны с удаляемым реквизитом. Если связь с этим реквизитом для элемента не является основной (данные подвала, данные, отображаемые в заголовке группы, и т. д.), то такой элемент не удаляется, но очищается связь с удаляемым реквизитом. Вне зависимости от ответа на вопрос об удалении будут удалены команды командного интерфейса формы, связанные с удаляемым реквизитом.

Если удалить команды формы, также предлагается удалить связанные элементы. При отрицательном ответе элементы формы не удаляются, но очищается свойство элемента **Команда**. При положительном ответе удаляются кнопки, связанные с удаляемой командой.

При разработке формы, которая должна отображаться и в интерфейсе **Такси** и в интерфейсе 8.2, требуется видеть, как форма отображается в обоих режимах интерфейса. Это можно сделать непосредственно в редакторе формы.

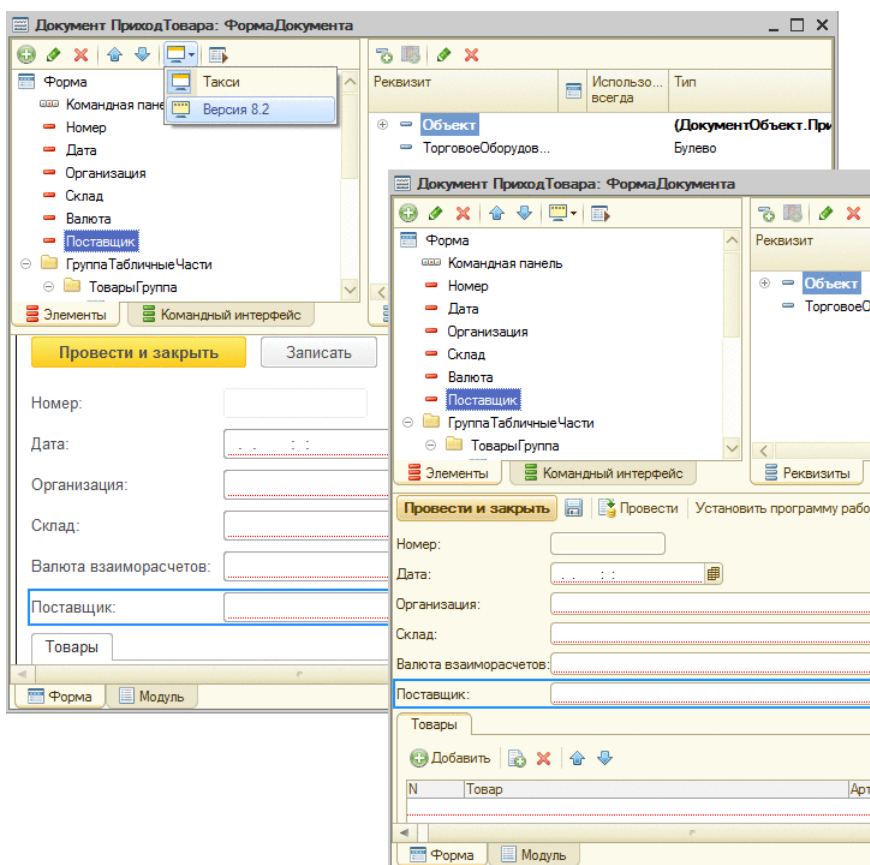


Рис. 361. Переключение режима интерфейса

На [рис. 361](#) приведен пример того, как меняется отображение при смене режима интерфейса с интерфейса **Такси** (верхняя левая часть рисунка) на интерфейс версии 8.2 (нижняя правая часть рисунка). Однако следует помнить, что в режиме тестирования формы в интерфейсе **Такси**, некоторые элементы изображаются в интерфейсе 8.2.

Кнопка **Вариант интерфейса** доступна только в том случае, когда свойство **Режим совместимости интерфейса** установлено в значение **Такси. Разрешить Версия 8.2** или **Версия 8.2. Разрешить Такси**. В остальных случаях кнопка не отображается в редакторе формы.

Если нужно открыть форму для проверки, следует нажать сочетание клавиш **Ctrl + R**. Форма будет открыта в отдельном окне.

При переименовании или удалении реквизита, параметра или элемента формы, система ищет случаи использования переименованного (удаляемого) объекта в модуле формы и во всей конфигурации (если это разрешено в настройках конфигуратора, см. [здесь](#)).

27.1.2. Ролевая настройка формы

В редакторе формы имеется возможность выполнить ролевую настройку поведения формы.

Данная возможность предоставляется для следующих свойств:

- **Просмотр** – свойство реквизита формы. Отсутствие возможности просмотра исключает реквизит из данных формы (реквизит не передается с сервера). Изменить эту настройку пользователь не может.
- **Редактирование** – свойство реквизита формы. Отсутствие возможности редактирования делает связанный с реквизитом элемент формы доступным только для просмотра. Вне зависимости от состояния данного свойства редактирование будет недоступно, если недоступен просмотр реквизита формы. Изменить эту настройку пользователь не может.
- **Пользовательская видимость** – свойство элемента формы. Определяет видимость элемента формы по умолчанию. При этом пользователь может самостоятельно изменить это свойство в редакторе настройки формы.
- **Использование** – свойство команды формы. Если у команды выключено использование, то в командном интерфейсе отсутствуют все связанные с ней кнопки. Изменить эту настройку пользователь не может.

Редакторы всех вышеперечисленных свойств выглядят одинаково и имеют одинаковый принцип работы. Рассмотрим работу редактора на примере свойства [Просмотр](#).

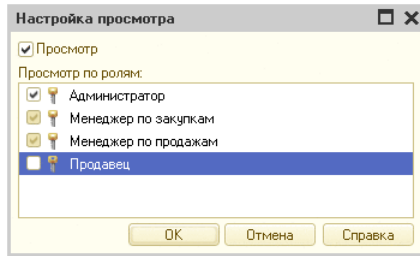


Рис. 362. Редактирование свойства «Просмотр» реквизита формы

Свойство [Просмотр](#) (в верхней части окна редактора) определяет состояние просмотра для каждой роли, у которой флажок находится в «третьем» состоянии (серый фон флажка). Затем состояния просмотра для всех доступных ролей складываются «по ИЛИ», и результат этого сложения будет определять итоговое свойство [Просмотр](#) для редактируемого объекта.

27.2. Редактор текстов

Текстовый редактор системы «1С:Предприятие» предоставляет пользователю все основные функции, необходимые при редактировании текстов. При работе с текстовым редактором доступны операции с блоками текста, функции поиска и замены, цветовое выделение синтаксических элементов программных модулей.

В системе «1С:Предприятие» текстовый редактор используется в двух режимах: для редактирования текстовых документов и как составная часть редактора форм для редактирования текстов модулей.

В данной главе приводится описание особенностей работы редактора текстов при редактировании текстов модулей.

Так как работа с любым текстовым редактором в системе Microsoft Windows осуществляется примерно одинаковым образом, в данном разделе будет дано описание специфических возможностей текстового редактора системы «1С:Предприятие».

ПРИМЕЧАНИЕ. В текстовом редакторе невозможно ввести символы, недопустимые с точки зрения спецификации XML версии 1.0 (<http://www.w3.org/TR/xml/>). При попытке ввода такого символа он просто игнорируется, при вставке из буфера обмена недопустимые символы пропускаются и во вставленный текст не попадают.

Таблицу сочетаний клавиш для редактора текстов можно посмотреть в справке при использовании программы.

27.2.1. Редактирование модулей

Редактирование модулей выполняется в процессе создания формы объекта конфигурации, а также при разработке модулей (модуль приложения, модуль внешнего соединения, общие модули, модули прикладных объектов).

При создании формы текстовый редактор выступает как составная часть редактора форм и вызывается щелчком мыши на закладке [Модуль](#) в окне редактора форм, в остальных случаях текстовый редактор вызывается в виде отдельного окна. Для открытия модуля управляемого приложения, модуля внешнего соединения, модуля сеанса и модуля обычного приложения нужно щелкнуть правой кнопкой мыши по имени конфигурации (самая верхняя строка дерева конфигурации) и в контекстном меню выбрать соответствующий пункт меню.

Для редактирования модуля объекта выбрать объект и в контекстном меню выбрать пункт [Открыть модуль объекта](#).

Для редактирования модуля менеджера (если такой модуль есть у объекта) следует выбрать объект и в контекстном меню выбрать пункт [Открыть модуль менеджера](#).

Чтобы отредактировать модуль команды, нужно дважды щелкнуть мышью по команде или в контекстном меню команды выбрать пункт [Открыть модуль команды](#).

Для редактирования общего модуля в окне [Конфигурация](#), в ветви [Общие – Общие модули](#) выбрать нужный модуль и в контекстном меню выбрать пункт [Открыть модуль](#).

Процесс редактирования текста программного модуля ничем не отличается от процесса редактирования текстовых документов – можно использовать все возможности редактора текстов.

В этом разделе будут описаны специфические режимы редактора текстов, которые доступны при редактировании модулей.

27.2.1.1. Выделение цветом синтаксических конструкций

Для удобства редактирования текстов модулей текстовый редактор имеет функцию выделения цветом элементов встроенного языка системы «1С:Предприятие» – ключевых слов, констант (не объектов конфигурации) различных типов, операторов, комментариев и других. Цвета, которыми будут выделяться разные типы синтаксических конструкций, можно установить в окне [Параметры](#), которое открывается выбором пункта [Сервис – Параметры](#) (о настройке параметров текстового редактора см. [здесь](#)).

Названия встроенных функций цветом не выделяются (цвет совпадает с цветом идентификаторов).

В общем случае когда текстовый редактор вызывается для редактирования текста модуля, эта функция включается автоматически. Однако в некоторых случаях текст модуля может быть расположен во внешнем текстовом файле. Тогда при открытии такого файла конфигуратор не распознает модуль и будет считать его обычным текстовым документом. В этом случае при редактировании текста будет недоступно выделение цветом синтаксических конструкций и автоматическое форматирование текста модуля. Для указания конфигуратору, что редактируется модуль, а не текстовый документ, служит пункт [Текст – Встроенный язык](#), выполняющий функцию переключателя.

Когда он включен (слева от слов [Встроенный язык](#) в меню появляется отметка), текстовый редактор считывает загруженный в него текст текстом модуля и выделяет цветом найденные синтаксические конструкции.

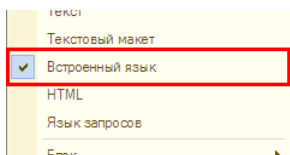


Рис. 363. Включено выделение цветом

Если этот режим включен при редактировании обычного текстового документа, для вывода текста будет также использоваться шрифт, установленный для текстов модулей в настройке параметров конфигулятора (пункт [Сервис – Параметры](#), закладка [Тексты](#), реквизит [Шрифт](#)).

Настройка параметров системы «1С:Предприятие» позволяет отключить режим выделения цветом синтаксических конструкций. Тогда выбор пункта [Встроенный язык](#) не включит выделение цветом синтаксических конструкций модуля, а будет использовать только установки шрифта модуля и шага табуляции.

Если режим выделения цветом синтаксических конструкций выключен, для вывода текста используются цвета операционной системы.

27.2.1.2. Группировка

В модулях или в текстовых документах, просматриваемых в режиме [Встроенный язык](#), некоторые синтаксические конструкции автоматически объединяются в группы. К таким конструкциям относятся [Если ... Тогда ... КонецЕсли](#), [Пока ... Цикл ... КонецЦикла](#), [Процедура ... КонецПроцедуры](#) и другие.

Группы текста позволяют лучше воспринимать различные части текста, а также переносить и копировать группу целиком. Отображение группировок можно увидеть на [рис. 364](#).

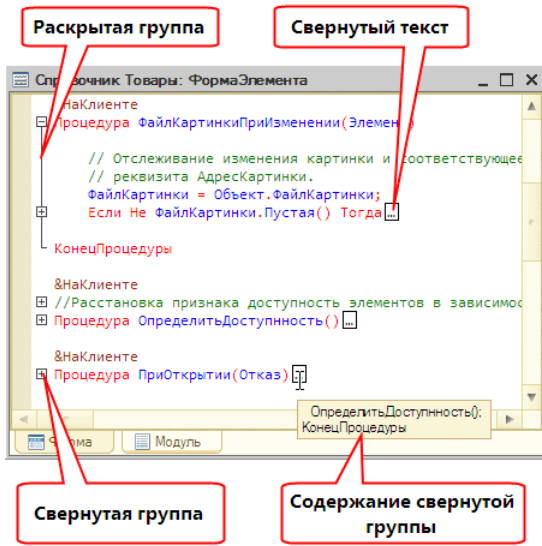


Рис. 364. Группировки в текстовом редакторе

Свернуть и развернуть группу можно с помощью мыши. Для этого достаточно щелкнуть левой кнопкой мыши по маркеру группы.

Если при этом нажать клавишу **Ctrl**, то сворачивание и разворачивание будет производиться также для подчиненных групп (условий, циклов и проч.).

Для отображения свернутого текста необходимо подвести курсор к маркеру, как показано на [рис. 364](#). Если текст группы достаточно большой, то выводится только его начальная часть.

В процессе разработки возникает необходимость группировать фрагменты модуля. Например, нужно выделить логически связанные процедуры и функции или выделить фрагменты алгоритма в рамках одной процедуры или функции. Это можно сделать с помощью директив препроцессора `#Область ... #КонецОбласти` (`#Region ... #EndRegion`).

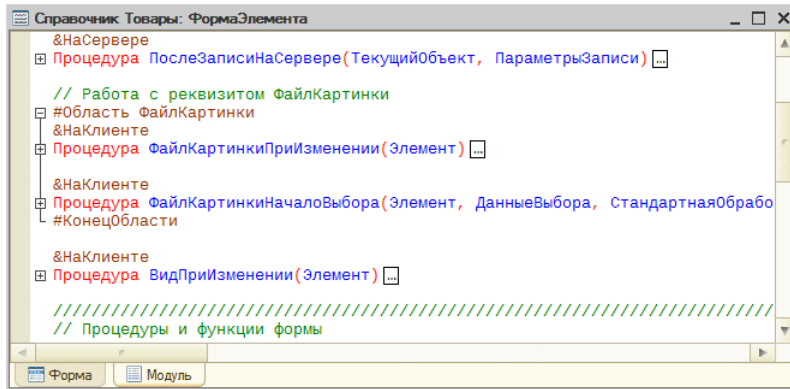


Рис. 365. Группировка строк

Все строки, которые находятся внутри области, ограниченные директивами препроцессора, в редакторе модуля можно свернуть (аналогично другим конструкциям встроенного языка). Каждая область должна обладать собственным именем, которое формируется по правилам встроенного языка. В примере, показанном на [рис. 365](#), именем области является идентификатор `ФайлКартинки`. Области могут быть вложенными, но не могут пересекаться с другими синтаксическими конструкциями языка. Далее приведены примеры использования областей, как корректные, так и некорректные:

```
// Верный пример использования области
#Область Верно
...
#Если Клиент Тогда
...
#КонецЕсли
...
#КонецОбласти
// Вложенные области
#Область АлгоритмРасчета
...
#Область Этап1
...
#КонецОбласти
...
#Область Этап2
...
#КонецОбласти
...
#КонецОбласти
// Неверный пример использования области
#Область Неверно
...
#Если Клиент Тогда
...
#КонецОбласти
...
#КонецЕсли
// Неверный пример использования области
#Область Неверно
Процедура ПолучитьДанные()
...
#КонецОбласти
...
КонецПроцедры
```

[Копировать в буфер обмена](#)

Область может описываться комментарием, располагающимся перед началом области (`#Область`). Этот комментарий также может группироваться.

С помощью команд, выполняемых с клавиатуры, можно легко управлять состоянием групп.

Клавиши	Действия
Ctrl + Num-	Сворачивает группу (курсор может быть в любом месте группы)
Ctrl + Num+	Разворачивает группу (курсор должен быть в первой строке группы)
Ctrl + Shift + Num-	Сворачивает все группы
Ctrl + Shift + Num+	Разворачивает все группы
Ctrl + Shift + R	Обновить группировки

При написании текста модуля, во время завершения написания синтаксической конструкции, автоматически создаются группы для всех синтаксических конструкций любого уровня вложенности. Обновление группировок выполняется с помощью пункта **Текст – Группировки – Обновить группировки** или автоматически при сохранении текста согласно настройке показа групп.

Первым считается уровень описания процедур, функций и областей, вторым – синтаксических конструкций, вложенных только в тело процедуры или функции, но не в тело других синтаксических конструкций. Режим показа групп можно настроить. О настройке режима см. [здесь](#).

27.2.1.3. Работа с закладками

В процессе работы с текстом модуля разработчик может отметить какую-либо строку текста для того, чтобы потом быстро перейти к отмеченному фрагменту. Такая отметка называется **закладкой**.

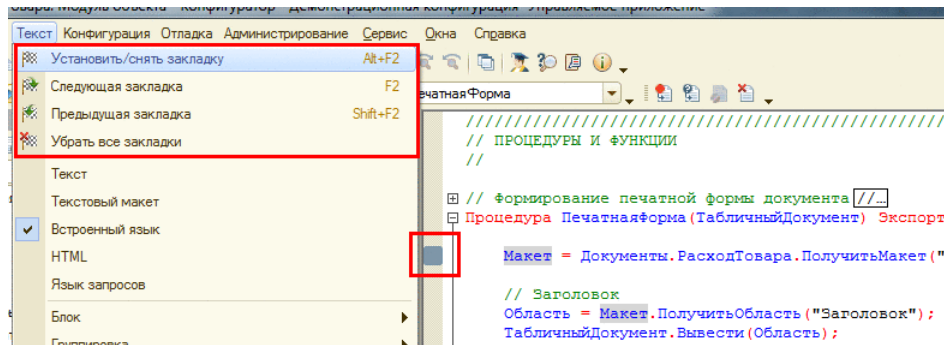


Рис. 366. Закладки в тексте

Для установки следует воспользоваться командой **Текст – Установить/снять закладку**. Закладка отмечается цветовой меткой в колонке, левее текста модуля.

Также имеется возможность перейти к следующей (команда **Текст – Следующая закладка**) или предыдущей закладке (команда **Текст – Предыдущая закладка**), а также удалить все установленные закладки (**Текст – Убрать все закладки**).

Список всех установленных закладок можно увидеть в специальном списке, который открывается с помощью команды **Правка – Список закладок**.

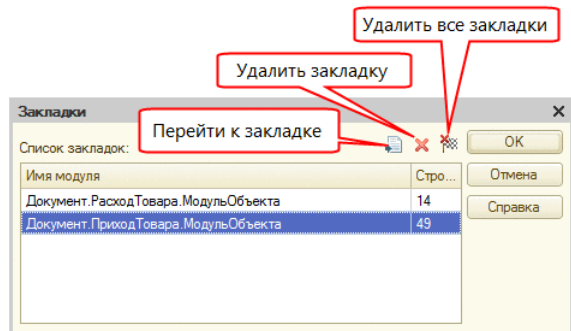


Рис. 367. Список закладок

С помощью данного списка можно перейти сразу к нужной закладке, удалить одну, конкретную, закладку или сразу все. Следует помнить, что переход к закладке осуществляется только с помощью специальной кнопки. Нажатие на кнопку **OK** не вызывает переход к отмеченной закладке.

Также следует отметить, что система сохраняет список закладок между сеансами редактирования документов и работы конфигуратора.

27.2.1.4. Форматирование модуля

Редактор текстов системы «1С:Предприятие» включает ряд режимов, облегчающих разработку модулей.

27.2.1.4.1. Форматирование синтаксических конструкций

Хорошим стилем написания модулей считается использование синтаксического отступа – выделения лидирующими пробелами (табуляцией) управляющих конструкций встроенного языка системы «1С:Предприятие», например, так, как это показано в приведенном ниже фрагменте модуля.

```

Процедура СледующийЭлемент(Справочник, Выборка)
Пока Истина Цикл
    Если Выборка.Следующий() = Ложь Тогда
        Выборка = Справочник.Выбрать();
        Продолжить;
    Иначе
        Прервать;
    КонецЕсли;
    Если Выборка.ЭтоГруппа Тогда
        Продолжить;
    КонецЕсли;
    Возврат;
КонецЦикла;
КонецПроцедуры

```

В данном фрагменте строки модуля, расположенные внутри структурных операторов **Если ... Тогда ... КонецЕсли** и **Пока ... Цикл ... КонецЦикла**, смещены вправо, чтобы подчеркнуть их «вложенность». Текст модуля, отформатированный с использованием синтаксического отступа, удобнее в восприятии и проще в отладке.

Текстовый редактор системы «1С:Предприятие» предоставляет функции автоматического форматирования управляющих конструкций встроенного языка. Для настройки автоматического форматирования в режиме установки параметров системы (меню **Сервис** главного меню конфигуратора, закладка **Модули**) можно выбрать один из двух видов отступа.

Синтаксический отступ выполняет автоматическое форматирование текста модуля, смещая вправо текст, расположенный внутри управляющих конструкций типа **Если ... Тогда ... КонечЕсли** и **Пока ... Цикл ... КонечЦикла** и подобных. Смещение выполняется за счет добавления в начало строк необходимого количества знаков табуляции.

«Обыкновенный» отступ автоматически выравнивает текст строки по левой границе предыдущей строки.

Если автоотступ отключен, никаких дополнительных символов в текст добавляться не будет.

Кроме автоматического форматирования текста модуля в процессе ввода можно также отформатировать уже введенный текст. Для этого необходимо выделить блок текста, который требуется отформатировать, и выбрать пункт **Текст – Блок – Форматировать**. При этом текстовый редактор проанализирует текст модуля и выполнит его форматирование, при котором содержимое каждой синтаксической конструкции будет сдвинуто вправо на величину табуляции независимо от первоначального расположения строк (лидирующих пробелов). В пустые строки устанавливаются знаки табуляции в соответствии с синтаксической конструкцией.

Блок текста также можно целиком сдвигать вправо или влево с шагом табуляции. Для этого необходимо выделить блок текста и выбрать команду **Текст – Блок – Сдвинуть вправо** (**Текст – Блок – Сдвинуть влево**).

Текстовый редактор системы «1С:Предприятие» осуществляет автоматическое удаление пробелов на концах строк. Это выполняется при записи модуля.

27.2.1.4.2. Вставка/удаление признака комментария

При отладке модулей зачастую бывает необходимо на время «отключить» некоторые строки модуля, чтобы они не исполнялись при работе системы. Обычно это выполняется путем превращения таких строк в комментарии – добавлением перед ними признака комментария «//». Позднее, чтобы опять «включить» закомментированные строки в работу, признак комментария удаляется.

Для выключения (и последующего включения) больших фрагментов модуля удобно использовать режим автоматической установки признаков комментария у всех строк выделенного блока или текущей строки (ее выделять не обязательно).

Для этого следует выделить блок текста или установить указатель на нужную строку и выбрать пункт **Текст – Блок – Добавить комментарий** (**Текст – Блок – Удалить комментарий**).

При удалении комментария, если перед строкой установлено несколько признаков комментария, удаляется только один.

27.2.1.4.3. Вставка/удаление переноса строки

При написании значения строковых констант для переноса строки используется символ «\».

Текстовый редактор позволяет легко добавлять или удалять этот символ в предварительно выделенные строки.

Для вставки/удаления символа переноса у всех строк выделенного блока или текущей строки (ее выделять не обязательно) нужно указать область и выбрать пункт **Текст – Блок – Добавить перенос строки** для вставки и **Текст – Блок – Удалить перенос строки** для удаления.

Символ переноса вставляется в первую значимую (не пробел и не знак табуляции) позицию каждой строки.

27.2.1.5. Переход по процедурам и функциям модуля

При значительном количестве имеющихся в модуле описаний процедур и функций удобно использовать режим поиска процедур, который предоставляет редактор текстов системы «1С:Предприятие».

Если воспользоваться командой **Текст – Процедуры и функции**, то на экран будет выдано окно, содержащее список всех процедур и функций редактируемого модуля.

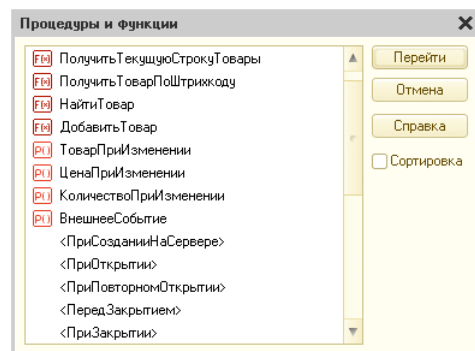


Рис. 368. Список процедур и функций

Имена процедур и функций в списке выдаются в порядке их расположения в модуле. Если установлен флажок **Сортировка**, список будет отсортирован по алфавиту.

Имена процедур и функций, уже расположенных в форме, имеют пиктограмму слева от наименования.

Для перехода к нужной процедуре или функции необходимо выделить ее имя в списке и нажать кнопку **Перейти**.

В списке процедур и функций в угловых скобках показываются наименования событий, процедура-обработчик которых еще не создана. Созданные процедуры и функции имеют перед наименованием пиктограмму **(P)** и **F(x)** соответственно. Состав событий, которые могут быть обработаны, определяются типом объекта и видом формы, а также составом элементов управления, расположенных в форме. При выборе такой строки в модуль добавляется текст predefinedной процедуры, а в соответствующее событие категории **События** – ссылка на эту процедуру.

ВНИМАНИЕ! Создание процедур-обработчиков событий, определенных системой для данного модуля, необходимо выполнять в палитре свойств для формы в категории свойств **События**, или с помощью окна **Процедуры и функции**, или из поля списка **Процедуры и функции**. При простом копировании процедур-обработчиков событий из других модулей обработчики событий для формы не будут иницированы системой, и скопированные процедуры не будут вызываться для обработки событий.

При перемещении указателя в строку модуля наименование текущей процедуры или функции показывается в поле выбора процедуры панели инструментов **Модуль** (команда **Процедуры и функции**). С помощью этого списка можно также перейти к нужной процедуре или функции.

Для перехода к процедуре, функции и переменным достаточно указать мышью их наименование и нажать клавишу **F12**. Переход возможен только для процедур, функций и переменных, расположенных в данном модуле, или к экспортируемым процедурам, функциям и переменным модуля приложения, общих модулей и модулей объектов. Для того чтобы вернуться к точке, из которой был осуществлен переход к определению, необходимо нажать сочетание клавиш **Ctrl+ "="** (рядом с клавишей "=").

```

ОбщегоНазначенияКлиентСервер.УстановитьЭлементОтбора (
    ГруппаОтбора,
    "Ссылка",
    Неопределено,
    ВидСравненияКомпоновкиДанных.Всписке, , Ложь
);

```

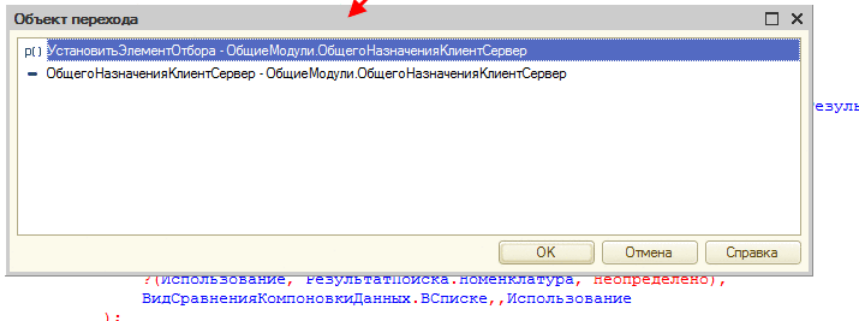


Рис. 369. Перейти к определению

Если записанное выражение состоит из частей, определенных в разных местах конфигурации, перед переходом на экран будет выведен список таких объектов для выбора перехода. Переход может осуществляться к определению переменной, определению объекта метаданных, тип которого имеет текущее выражение, или определению процедуры или функции, используемой в выражении (например, в модуле объекта, тип которого имеет выражение).

Например, для выражения `Спр.НайтиПоКоду()`, где `Спр` определен как справочник `Валюты`, при нажатии клавиши **F12** будет выведен список перехода к определению переменной `Спр` и определению объекта метаданных `Справочники.Валюты` в дереве объектов конфигурации.

Существует возможность выполнить поиск мест использования переменной, процедуры или функции. Поиск будет выполняться по всей конфигурации. Для выполнения такого поиска следует поместить курсор «внутри» того имени, использование которого нужно найти, и вызвать команду контекстного меню редактора текстов **Найти использование** (или нажать сочетание клавиш **Alt+F12**).

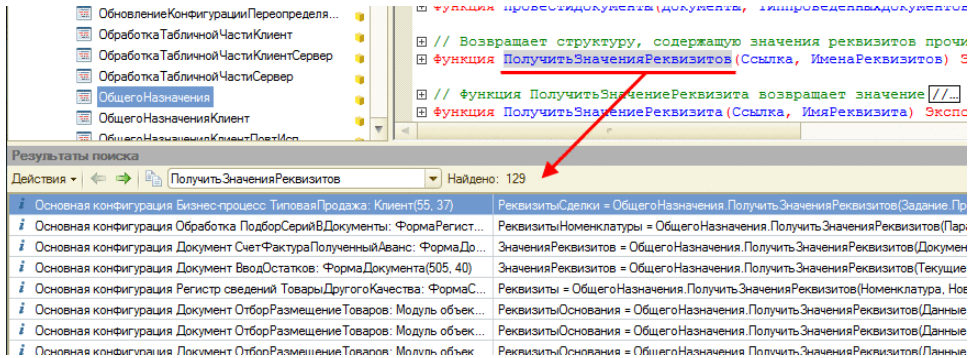


Рис. 370. Найти использование

После выполнения поиска открывается окно с результатами поиска. Из этого окна можно перейти в соответствующий модуль с помощью двойного щелчка мышью по требуемой строке.

Первый символ в результатах поиска описывает достоверность определения каждого найденного использования. Символ «!» означает, что использование определено полностью и однозначно, а символ «i» означает, что использование переменной, процедуры или функции определено не четко.

27.2.1.6. Контекстная подсказка при вводе текстов модулей

Текстовый редактор системы «1С:Предприятие» предоставляет средство контекстного ввода выражений с использованием системных объектов, их свойств, методов, процедур и функций, наименований объектов, определенных в конфигурации, а также переменных, процедур и функций, определенных в общих модулях, модулях прикладных объектов и модулях форм. В список включаются предопределенные элементы справочников, планов счетов, планов видов характеристик и планов видов расчетов.

В список контекстной подсказки могут быть включены шаблоны текстов и ключевые слова.

ПРИМЕЧАНИЕ. Контекстная подсказка текста не поддерживается системой для текстовых документов с установленным свойством **Встроенный язык**.

Ввод осуществляется из списка, который выводится в виде контекстного меню в месте текущего расположения курсора (с учетом близости границ экрана).

Список вызывается нажатием комбинации клавиш **Ctrl + <клавиша Пробел>** на любой стадии набора выражения или автоматически после ввода символа «.» после выражения, представляющего собой объект, имеющий свойства и/или методы (при разрешенном вызове контекстной подсказки – см. [здесь](#)).

Состав списка зависит от контекста выполнения программного модуля (см. раздел «Общее описание языка» справки по встроенному языку) и предварительно введенного текста.

На начальном этапе, когда текст еще не введен или введены только начальные символы выражения, состав списка определяется контекстом выполнения.

Список представлен в виде строк текста, отсортированных по алфавиту.

Если при открытии списка был введен фрагмент текста, то список позиционируется на первой строке, наименование которой максимально включает набранный или выделенный текст (от начала наименования). Если набранный текст не содержится в строках списка, то список позиционируется на строке, в которой максимально представлен набранный текст.

При открытом списке можно продолжить набор текста. В этом случае список будет последовательно позиционироваться на строках, совпадающих с набираемым текстом.

Список можно просмотреть стандартным способом. При нажатии клавиши **Enter** содержимое выбранной строки переносится в модуль, заменяя выделенный или набранный текст.

ПРИМЕЧАНИЕ. При переносе текста в модуль не вставляются параметры методов и используется контекст клиентского приложения.

В левой части строк имеются пиктограммы, показывающие вид объекта и тип места его расположения.

Пиктограмма	Объект
черта (черная)	<ul style="list-style-type: none"> Свойства глобального контекста, Системные наборы значений,

	<ul style="list-style-type: none"> • Перечисления
черта (зеленая)	<ul style="list-style-type: none"> • Свойства объектов универсальных коллекций значений, • Свойства интерфейсных объектов, • Реквизиты прикладного объекта, • Предопределенные элементы
черта (синяя)	Экспортируемые переменные модулей
черта (красная)	Локальные переменные модуля
P() (черная)	Процедуры глобального контекста
P() (зеленая)	Процедуры интерфейсных и прикладных объектов
P() (синяя)	Экспортируемые процедуры других модулей
P() (красная)	Локальные процедуры модуля
F() (черная)	Функции глобального контекста
F() (зеленая)	Функции интерфейсных объектов и прикладных объектов
F() (синяя)	Экспортируемые функции других модулей
F() (красная)	Локальные функции модуля
Цветные строки	Ключевые слова (<i>Если, Цикл, Попытка</i> и др.)
Картинка шаблонов	Шаблон текста

Шаблон текста включается в список только в том случае, если в нем определена строка автозамены.

Если в результате набора или после выбора из списка текст будет представлять выражение, имеющее свойства или методы, то после ввода символа «.» на экран автоматически будет выведен список, содержащий возможный набор свойств и методов, предоставляемых данным выражением.

Например, при вводе текста «Справочники.» выводится список, содержащий наименования всех справочников, описанных в данной конфигурации. После выбора конкретного справочника и ввода символа «.» на экран снова будет выведен список, но, в отличие от предыдущего, в нем будут содержаться наименования процедур и функций работы со справочником, а также предопределенные элементы справочника. При выборе метода, возвращающего значение некоторого типа, также имеющего свойства и методы, контекстный ввод может быть продолжен (нужно в конце наименования ввести открывающую и закрывающую скобки). Список контекстной подсказки содержит только возможный набор, определяемый типом введенного выражения.

Для переменных модулей также можно использовать механизм контекстной подсказки. При использовании объявления переменных следует помнить о следующей особенности: для того, чтобы явно объявленные переменные попадали в контекстную подсказку, каждая переменная должна быть оформлена «своим» оператором *Перем* (см. [здесь](#)). В противном случае контекстная подсказка будет отображать только первую переменную из каждого оператора *Перем*. В следующем примере будут отображаться все объявленные переменные:

```
Перем ИмяМакета;
Перем СкрыватьСкидку;
Перем СальдоВзаиморасчетов;
```

[Копировать в буфер обмена](#)

В следующем примере контекстная подсказка будет отображать только одну переменную: *ИмяМакета*:

```
Перем ИмяМакета, СкрыватьСкидку, СальдоВзаиморасчетов;
```

[Копировать в буфер обмена](#)

Контекстная подсказка может использоваться при вводе оператора *Новый*, а также для переменных, созданных с помощью оператора *Новый*.

Контекстная подсказка может использоваться при вводе различных ключевых слов (например, *Если, Для, Цикл* и др.). Ключевые слова входят в список, вызываемый стандартным образом нажатием комбинации клавиш **Ctrl + <клавиша Пробел>** на любой стадии набора слова.

Показ в списке ключевых слов можно настроить (см. [здесь](#)).

Если для какой-либо переменной или метода список состоит только из одной строки, то нажатие комбинации клавиш **Ctrl + <клавиша Пробел>** приводит к непосредственной вставке этой строки.

Контекстная подсказка также позволяет отображать перечень параметров какого-либо метода и его тип. Параметры отображаются как для объектов глобального контекста и объектов встроенного языка, так и для методов прикладного решения, описание которых соответствует определенным требованиям. Для операторов встроенного языка контекстная подсказка по параметрам не отображается. Отображение параметров начинается после ввода символов "(" или "," (можно настроить, подробнее см. [здесь](#)) или после нажатия комбинации клавиш **Ctrl + Shift + <клавиша Пробел>**, если она нажата после символа "(" вызова процедуры или функции.

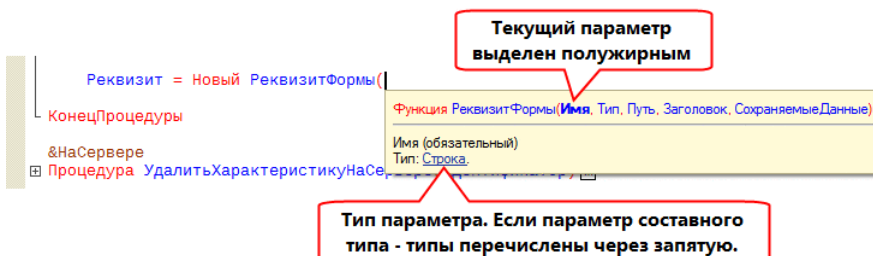


Рис. 371. Контекстная подсказка параметров

Если у метода есть несколько вариантов синтаксиса, то будет отображен первый по порядку вариант. Однако перед описанием синтаксиса будут присутствовать стрелки вверх и вниз, с помощью которых можно переключаться между различными вариантами. Переключаться можно также с помощью клавиш **Ctrl + <Стрелка вверх>** и **Ctrl + <Стрелка вниз>**.

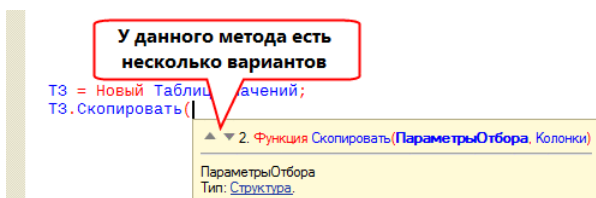


Рис. 372. Несколько вариантов синтаксиса в контекстной подсказке параметров

Под описанием параметра отображается тип значения, которое можно передавать в этот параметр. Если значение может быть нескольких типов, эти типы будут перечислены через запятую. Имя типа является гиперссылкой, нажатие на которую приведет к открытию синтакс-помощника (см. [здесь](#)) с информацией о типе. Контекстная подсказка параметров закрывается после ввода символа ")" или нажатия клавиши Esc при открытом окне контекстной подсказки параметров.

Для того чтобы контекстная подсказка показывала описание параметров и их типы для методов прикладного решения, перед началом метода должен располагаться комментарий, выполненный по определенным правилам. В общем случае, описание процедуры или функции состоит из следующих секций:

- секция **Описание** – содержит краткое описание назначения и/или принципов работы процедуры или функции, может быть единственной секцией для функций без параметров;
- секция **Параметры** – описывает параметры процедуры или функции. Если параметров нет – секция пропускается;
- секция **Возвращаемое значение** – описывает тип и содержание возвращаемого значения функции. Для процедур эта секция отсутствует;
- секция **Пример** – содержит пример использования процедуры или функции.

Общий формат комментария следующий:

```
<Комментарий> =
  [<СекцияЗаголовка>]
  [<СекцияПараметров>]
  [<СекцияВозвращаемогоЗначения>]
  [<СекцияПримера>]
<НоваяСтрока> = Начало строки модуля
<Имя> = Идентификатор встроенного языка
<Тип> = <Имя> | <Имя> "." <Имя>
<СписокТипов> = <Тип>{ " , " <Тип> }
<СтрокаТекста> = Произвольный текст без перевода на другую строку
<Текст> = <СтрокаТекста> { <НоваяСтрока> <СтрокаТекста> }
<СекцияЗаголовка> = [<Текст>]
<СекцияПараметров> = <НоваяСтрока> ("Параметры:" | "Parameters:") {<ОписаниеПараметра> {<ОписаниеТипа> } }
<ОписаниеПараметра> = <НоваяСтрока> <Имя> ["-" <СписокТипов>] "-" [<Текст>]
<ОписаниеТипа> = <НоваяСтрока> "-" <СписокТипов> "-" [<Текст>]
<СекцияВозвращаемогоЗначения> = <НоваяСтрока> ("Возвращаемое значение:" | "Returns:") <ОписаниеЗначения>
<ОписаниеЗначения> = <НоваяСтрока> [<СписокТипов> "-" ] [<Текст>]
<СекцияПримера> = <НоваяСтрока> ("Пример:" | "Sample:") <НоваяСтрока> <Текст>
```

[Копировать в буфер обмена](#)

В результате, комментарий перед началом процедуры или функции, оформленный в соответствии с описанным форматом, приведет к тому, что контекстная подсказка параметров методов будет адекватно показывать подсказку для метода прикладного решения.

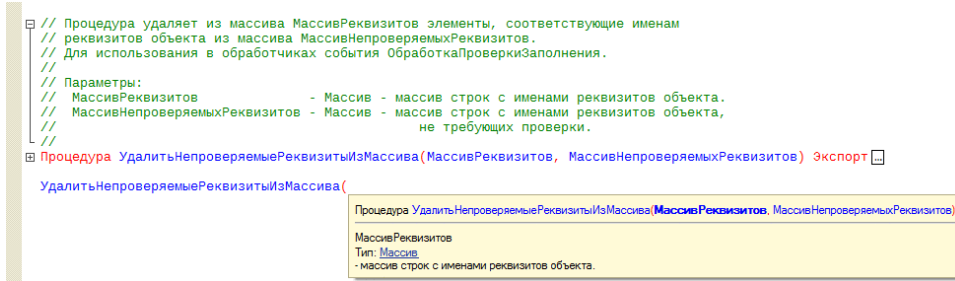


Рис. 373. Контекстная подсказка параметров методов прикладного решения

Таким образом, механизм контекстной подсказки текстового редактора системы «1С:Предприятие» предоставляет способ быстрого и правильного набора текстов модулей.

27.2.1.7. Средства рефакторинга

Рефакторинг – процесс переработки кода на встроенном языке с целью добиться улучшения читаемости кода и общей внутренней структуры при полном и точном сохранении изначального смысла и поведения. Конфигуратор предлагает несколько встроенных инструментов, которые могут помочь при проведении рефакторинга прикладного решения. Доступ к этим инструментам выполняется с помощью подменю **Рефакторинг** контекстного меню редактора текста встроенного языка или с помощью одноименного подменю меню **Текст** главного меню конфигуратора.

Далее средства рефакторинга будут рассмотрены более подробно. В этом рассмотрении будут использоваться следующие термины:

- **Блок** – множество строк между ключевыми словами **Тогда...КонечЕсли**, **Тогда...Иначе**, **Тогда...ИначеЕсли**, **Цикл...КонечЦикла**, **Попытка...Исключение**, **Исключение...КонечПопытки**, **Процедура...КонечПроцедуры**, **Функция...КонечФункции**;
- **Терминальный вызов** – вызов, являющийся последней выполняемой инструкцией метода встроенного языка:
 - Располагается перед ключевым словом **КонечПроцедуры**, **КонечФункции** или **Возврат**;
 - Является последним в рекурсивно-последнем блоке инструкций метода.
- **Асинхронный метод** – метод, который может использоваться в цепочках вызовов с использованием обработчиков оповещений.

27.2.1.7.1. Выделить фрагмент

Данная команда преобразует выделенный, синтаксически целостный, фрагмент кода в отдельную процедуру или функцию. Тип выделяемого фрагмента зависит от итоговых областей видимости переменных родительского метода и структуры фрагмента.

В том случае, когда выделенный фрагмент может выступать источником в операторе присваивания (см. [здесь](#)), фрагмент будет выделен как функция. Так же в функцию выделяется фрагмент, в котором инициализируется единственная переменная, которая используется в оставшейся части родительского метода. В остальных случаях фрагмент выделяется в процедуру.

В случае использования переменных выделяемого фрагмента в оставшейся части родительского метода, эти переменные передаются выделяемому методу в виде параметров, с их предварительной инициализацией или объявлением (при наличии такой необходимости).

Если родительский метод (из которого выделяется фрагмент) предваряется какой-либо директивой компиляции (см. [здесь](#)), то выделяемый метод также будет предваряться точно такой же директивой компиляции.

Код нового метода размещается сразу после кода родительского метода (из которого выделяется фрагмент) и соответствующим образом форматируется (см. [здесь](#)).

27.2.1.7.2. Переименовать

Данная команда предназначена для изменения имени переменной или метода. Если обращение к экспортной переименоваемой переменной или методу нельзя определить однозначно, то переименование не выполняется и выводится список использования переменной или метода.

Для экспортных переменных и методов модуля формы производится анализ уникальности идентификатора в рамках всей конфигурации и отсутствия синонимов среди методов и свойств объектов платформы. В случае выполнения этого условия пользователю предлагается возможность глобальной замены идентификатора во всех местах его возможного использования.

В случае переименования методов выполняется дополнительная проверка передачи их имени в качестве строкового литерала:

- в конструкторы объектов:
 - Действие;
 - ОписаниеОповещения.
- методы глобального контекста:
 - УстановитьДействие ();
 - ПодключитьОбработчикИзмененияДанных ();
 - ОтключитьОбработчикОжидания ();
 - ПодключитьОбработчикОжидания ();
 - ПодключитьОбработчикОповещения ().

Переименование метода-обработчика в модуле формы приводит к автоматически замене ссылки на переименоваемый обработчик в соответствующих свойствах элементов формы и командах. Когда выполняется переименование метода-обработчика из палитры свойств элемента формы или команды – автоматически переименоываются все вызовы данного обработчика из программного кода (если таковые встречаются).

27.2.1.7.3. Создать описание метода

Команда создает перед объявлением метода комментарий, который будет использоваться для контекстной подсказки при вводе текста модулей (см. [здесь](#)).

27.2.1.7.4. Создать обработку оповещения

Данная команда предназначена для упрощения применения асинхронных методов встроенного языка (см. [здесь](#)). Команда становится доступной после того, как введено имя метода и открывающая круглая скобка (курсор должен располагаться на имени метода). Дальнейшее поведение команды эквивалентно поведению команды рефакторинга [Преобразовать вызов](#), за исключением переименования метода, к которому применена команда рефакторинга. Если метод уже содержит передачу объекта типа [ОписаниеОповещения](#), то перед продолжением работы пользователю предлагается подтвердить выполнение команды.

27.2.1.7.5. Работа с нерекондуемыми синхронными вызовами

Данный раздел описывает команды рефакторинга, которые помогают при переработке кода прикладного решения для использования асинхронных вызовов взамен синхронных вызовов (см. [здесь](#)).

Преобразовать вызов

Команда заменяет тривиальные и простые случаи использования синхронных методов на их асинхронные аналоги (см. [здесь](#)).

Тривиальным считается случай, преобразование которого затрагивает только один метод и не требует операций, отличных от выделения фрагмента. При наличии следующих признаков, использование модального вызова считается тривиальным:

1. Вызов расположен в процедуре, не вызываемой из других процедур и функций, за исключением рекурсивно-терминальных вызовов. При анализе на терминальность допустима неопределенность принадлежности вызывающих методов. Если данный признак не наблюдается, то случай может быть простым или требовать преобразование в асинхронную процедуру (см. [здесь](#)).
2. Вызов расположен вне блоков цикла, за исключением случая инструкций [Прервать](#) или [Возврат](#), следующих на том же уровне вложенности.
3. Вызов расположен вне блоков перехвата исключения.
4. Уровень вложенности модального вызова не превышает 2, за исключением рекурсивно-последних инструкций блока. Уровень вложенности модального вызова можно уменьшить с помощью команды выделения фрагмента (см. [здесь](#)). Если выделяемый фрагмент содержит оператор [Возврат](#), то для выделения такого фрагмента следует использовать команду выделения фрагмента в асинхронную процедуру (см. [здесь](#)).
5. Параметры процедуры не изменяются после вызова модального диалога – ни в самой процедуре, ни в одном из вложенных вызовов.

Простым считается случай, не удовлетворяющий только первому условию тривиальности и, требующий дополнительно только преобразование процедуры или функции, содержащей синхронный вызов в асинхронный вид. При обнаружении такого синхронного вызова, и если такое преобразование возможно, система автоматически предлагает выполнить преобразование в асинхронную процедуру (см. [здесь](#)).

Результатом преобразования являются:

- Исходная процедура с заменой синхронного вызова на асинхронный вызов;
- Процедура обработки оповещения. Если синхронный вызов являлся последней инструкцией родительского блока – решение о создании пустой процедуры принимает пользователь
- Фрагмент-окончание – общий для исходной процедуры и процедуры обработки оповещения код, в случае, когда вызов синхронного метода не входит в единственную последовательность выполнения кода.

Если случай не соответствует одному или нескольким требованиям, пользователю предоставляется возможность выполнить частичное преобразование: максимальный синтаксически-корректный фрагмент кода будет выделен в процедуру обработки оповещения, и после вызова асинхронного метода будет вызвано исключение. В код модуля будет так же добавлен комментарий с описанием причин несоответствия.

Преобразовать в асинхронную процедуру

Команда используется для преобразования метода к виду, который допускает использование метода в асинхронных вызовах. Преобразование затрагивает все уже существующие последовательности вызовов с участием выбранного метода, однако методы, которые вызываются из преобразуемого метода, не подвергаются преобразованию. Данную команду следует применять в тех случаях, когда необходимо преобразовать модальный вызовы (один или несколько) внутри функции или процедуры, вызываемой из других методов. После выполнения данной команды будет необходимо преобразовать модальный вызов в его блокирующий аналог с помощью команды преобразования модального вызова (см. [здесь](#)).

Преобразование методов, изменяющих свои ссылочные параметры, не поддерживается в целях предотвращения труднодиагностируемых логических ошибок. Также не поддерживается преобразование последовательностей вызовов, содержащих непосредственный вызов процедуры обработки событий форм и объектов метаданных в связи со строго фиксированным списком их формальных параметров. Если преобразование затрагивает несколько модулей – от пользователя требуется дополнительное подтверждение.

Рассмотрим пример преобразования вызова в асинхронный вид.

ПРИМЕЧАНИЕ. Данный пример служит исключительно для демонстрации процесса преобразования и не является примером выполнения каких-либо действий.

Допустим, в модуле формы имеется следующая цепочка вызовов:

```
&НаКлиенте
Перем ПроцедураВыполнена;
&НаКлиенте
Перем ФункцияВыполнена;
&НаКлиенте
```

[Копировать в буфер обмена](#)

```

Процедура КомандаФормы(Команда)
    ПроцедураВыполнена = Ложь;
    ФункцияВыполнена = Ложь;
    ВыполнитьПроцедуру();
КонiecПроцедуры
&НаКлиенте
Процедура ВыполнитьПроцедуру()
    Результат = ВыполнитьФункцию();
    ПроцедураВыполнена = Истина;
КонiecПроцедуры
&НаКлиенте
Функция ВыполнитьФункцию()
    ПоказатьДиалог();
    ФункцияВыполнена = Истина;
    Возврат Истина;
КонiecФункции
&НаКлиенте
Процедура ПоказатьДиалог()
    Предупреждение("Предупреждение!");
КонiecПроцедуры

```

Выполнять преобразование следует для метода `ПоказатьДиалог()`.

Цепочка вызовов начинается при вызове из обработчика команды формы (процедура `КомандаФормы()`) процедуры `Процедурал()`. Действия, выполняемые в методах `ВыполнитьПроцедуру()` и `ВыполнитьФункцию()` в данном примере являются демонстрацией наличия некоторого программного кода после вызова другой процедуры.

Цепочка вызовов выглядит следующим образом: `КомандаФормы() -> ВыполнитьПроцедуру() -> ВыполнитьФункцию() -> ПоказатьДиалог()`.

Т.к. в самом «низу» стека вызовов вызывается модальный диалог, то всю цепочку вызовов методов необходимо преобразовать к асинхронному виду. Метод `Предупреждение()` может быть заменен на асинхронный аналог, для других процедур этот аналог будет создаваться системой.

Выполним преобразование метода `ПоказатьДиалог()` и отказаться от предложения создания пустых обработчиков оповещения. В результате получится следующий программный код:

[Копировать в буфер обмена](#)

```

&НаКлиенте
Перем ПроцедураВыполнена;
&НаКлиенте
Перем ФункцияВыполнена;
&НаКлиенте
Процедура КомандаФормы(Команда)
    ПроцедураВыполнена = Ложь;
    ФункцияВыполнена = Ложь;
    ВыполнитьПроцедуру(Неопределено);
КонiecПроцедуры
&НаКлиенте
Процедура ВыполнитьПроцедуру(Знач Оповещение)
    Результат = Неопределено;
    ВыполнитьФункцию(Новый ОписаниеОповещения("ВыполнитьПроцедуруЗавершение", ЭтотОбъект, Новый Структура("Оповещение", Оповещение)));
КонiecПроцедуры
&НаКлиенте
Процедура ВыполнитьПроцедуруЗавершение(Результат1, ДополнительныеПараметры) Экспорт
    Оповещение = ДополнительныеПараметры.Оповещение;
    Результат = Результат1;
    ПроцедураВыполнена = Истина;
    ВыполнитьОбработкуОповещения(Оповещение);
КонiecПроцедуры
&НаКлиенте
Процедура ВыполнитьФункцию(Знач Оповещение)
    ПоказатьДиалог(Новый ОписаниеОповещения("ВыполнитьФункциюЗавершение", ЭтотОбъект, Новый Структура("Оповещение", Оповещение)));
КонiecПроцедуры
&НаКлиенте
Процедура ВыполнитьФункциюЗавершение(Результат, ДополнительныеПараметры) Экспорт
    Оповещение = ДополнительныеПараметры.Оповещение;
    ФункцияВыполнена = Истина;
    ВыполнитьОбработкуОповещения(Оповещение, Истина);
    Возврат;
КонiecПроцедуры
&НаКлиенте
Процедура ПоказатьДиалог(Знач Оповещение)
    Предупреждение("Предупреждение!");
    ВыполнитьОбработкуОповещения(Оповещение);
КонiecПроцедуры

```

Во время преобразования не был изменен вызов модального вызова `Предупреждение()` в методе `ПоказатьДиалог()`, т. к. замена модального вызова не является задачей описываемой команды. Также стоит отметить, что все преобразуемые методы теперь получают в качестве входного параметра описание оповещения. Теперь цепочка вызовов выглядит следующим образом: `КомандаФормы() -> ВыполнитьПроцедуру() -> ВыполнитьФункцию() -> ПоказатьДиалог() -> ВыполнитьФункциюЗавершение() -> ВыполнитьПроцедуруЗавершение()`. Вызов методов с суффиксом `Завершение` выполняется с помощью передачи оповещений.

Выделить в асинхронную процедуру

Данная команда преобразует выделенный, синтаксически целостный, фрагмент кода в отдельную процедуру или функцию, одновременно преобразуя выделяемый метод к асинхронному виду.

Найти вызовы модуля

Команда выводит список использования нереконструируемых синхронных методов в текущем модуле.

Преобразовать вызовы модуля

Команда выполняет преобразование всех тривиальных (см. [здесь](#)) случаев использования нереконструируемых синхронных вызовов в их асинхронные версии (см. [здесь](#)). Преобразование выполняется в рамках текущего модуля.

Анализ нереконструируемых синхронных вызовов конфигурации

С помощью данной команды (**Главное меню – Конфигурация – Рефакторинг – Анализ нереконструируемых синхронных вызовов конфигурации**) во всей конфигурации выполняется поиск нереконструируемых синхронных вызовов и анализ возможности их преобразования. Если в каком-то случае преобразование будет невозможно – система укажет причину этой невозможности.

Флажок **Отображать методы объектов неопределенного типа** указывает системе, что следует включать в результат поиска методы, которые являются синхронными, но однозначно нельзя определить объект, у которого они вызываются.

Если в диалоге настройки поиска выбран флажок **Производить преобразование найденных вызовов**, то выполняется преобразование только тривиальных (см. [здесь](#)) случаев использования синхронных вызовов с выводом результатов операции.

27.2.1.8. Синтаксический контроль модуля

Редактируемый модуль может быть проверен на правильность использования синтаксических конструкций встроенного языка.

Для выполнения синтаксического контроля модуля необходимо воспользоваться пунктом **Текст – Синтаксический контроль**.

Синтаксический контроль выполняется в следующей последовательности:

- общие модули,
- модуль управляемого приложения,
- модуль объекта,
- модуль формы.

При этом контроль модулей выполняется, если модуль еще не проходил контроля или был модифицирован.

При контроле модуля проверяются только те модули, которые в списке расположены до данного модуля. Например, при проверке модуля приложения проверяются только общие модули. Модуль внешнего соединения проверяется только при его редактировании.

При наличии ошибок их список будет выдан в окне сообщений с указанием полного адреса месторасположения и описания ошибки. При подведении указателя мыши к строке, содержащей сообщение об ошибке, он принимает вид увеличительного стекла. Для перехода к строке модуля, вызвавшей ошибку, следует дважды щелкнуть мышью по этому сообщению. Если модуль, содержащий ошибку, закрыт, он будет открыт автоматически.

Если ошибки не обнаружены, в окно сообщений будет выдано сообщение об отсутствии ошибок в модуле.

В режиме настройки параметров конфигулятора (пункт [Сервис – Параметры](#), закладка [Текст модуля](#), реквизит [Проверять автоматически](#)) можно включить режим автоматической проверки модуля. В этом случае, если модуль был изменен, при закрытии окна модуля или при сохранении конфигурации в целом будет выполняться синтаксический контроль модуля.

Режим автоматической проверки удобно использовать, когда производится отладка какого-либо элемента конфигурации.

Для полного синтаксического контроля всех модулей конфигурации за один проход следует выбрать пункт [Конфигурация – Синтаксический контроль модулей](#).

В процессе исправления ошибок в модулях можно получить подсказку по встроенному языку, вызвав синтакс-помощник и найдя в нем описание нужного элемента встроенного языка (см. [здесь](#)).

Подсказку по конкретному элементу языка (оператору, процедуре, функции, свойству, методу) можно получить, если поместить курсор в модуле на этот элемент языка и нажать клавиши **Ctrl + F1**. В синтакс-помощнике будет выдано описание выбранного элемента встроенного языка.

27.2.1.9. Ограничение доступа к модулям конфигурации

Для некоторых модулей можно установить пароль доступа. Основное назначение пароля – защита авторских прав разработчиков конфигураций.

При установке пароля существуют следующие ограничения:

- не защищается модуль управляемого приложения;
- не защищаются модули форм;
- не защищаются модули команд;
- не защищаются модули, включающие директивы препроцессора;
- не защищаются клиентские общие модули, работающие в управляемом режиме (тонкий клиент, веб-клиент и управляемый режим толстого клиента).

27.2.1.9.1. Установка пароля доступа

Для установки пароля нужно открыть требуемый модуль и выбрать команду [Текст – Установить пароль](#). Пункт доступен, если модуль открыт для записи.

Если модуль содержит директивы препроцессора, то программа выводит предупреждение: [Защищенный модуль не должен содержать директив препроцессора. Продолжить?](#) Если нажать кнопку [Нет](#), то попытка установки пароля не производится. Если нажать кнопку [Да](#), то установка пароля становится возможной, и подразумевается, что в дальнейшем директивы будут удалены. Если директивы не удалены, то в режиме 1С:Предприятие система выдаст предупреждение: [Ошибка компиляции: модуль <наименование модуля>](#). Исходный текст модуля недоступен, и скомпилированный образ отсутствует.

На экран выводится диалог ввода пароля.

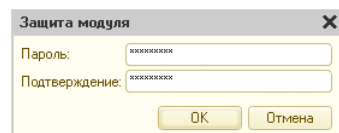


Рис. 374. Установка пароля на модуль

Нужно ввести пароль и повторить его. Для установки пароля нажать кнопку [OK](#), для отказа от установки – кнопку [Отмена](#).

27.2.1.9.2. Открытие защищенного модуля

Если на модуль установлен пароль доступа, то при попытке открыть модуль выводится диалог ввода пароля.

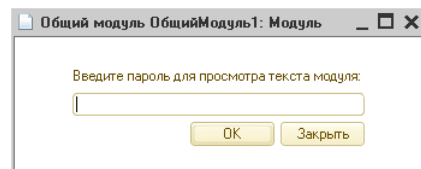


Рис. 375. Открытие модуля с установленным паролем

Если пароль указан верно, то модуль открывается. Если введен неправильный пароль, то выводится предупреждение [Некорректный пароль](#) и модуль не открывается.

27.2.1.9.3. Смена пароля

Для смены пароля нужно открыть модуль и выбрать команду [Текст – Установить пароль](#). На экран выводится диалог ввода пароля. Следует ввести прежний пароль. Если пароль введен правильно, то на экран выводится диалог ввода нового пароля (см. [здесь](#)).

Для снятия пароля нужно очистить поля для ввода пароля и, не вводя значения пароля, нажать кнопку [OK](#).

27.2.2. Редактирование текстовых макетов

Если в конструкторе макетов в качестве типа макета выбран [Текстовый макет](#), то открывается текстовый редактор в режиме ввода текстового макета.

Режим редактирования текстового макета также доступен для текстовых документов с установленным расширением [Текстовый макет](#).

27.2.2.1. Формат текстового макета

Весь текст макета делится на области. Области должны следовать друг за другом и не могут пересекаться или включаться друг в друга. В тексте макета области выделяются следующим образом:

[Копировать в буфер обмена](#)

```
#Область XXXXX
#КонецОбласти
```

Где `XXXXX` – имя области.

Конец области указывать не обязательно. Объявление начала области означает окончание предыдущей области.

Для хранения вариантов области на нескольких языках после имени области может следовать код языка, для которого написана область.

[Копировать в буфер обмена](#)

```
#Область Заголовок1 RU
#КонецОбласти
#Область Заголовок1 EN
#КонецОбласти
```

Если для какого-либо варианта области код языка не указан, то этот вариант области будет возвращаться, когда в метод `ПолучитьОбласть()` передан код языка, не указанный ни в одном из вариантов описаний областей.

Каждая область состоит из служебной и текстовой частей.

[Копировать в буфер обмена](#)

```
#Область XXXXX
[ служебная часть ]
[ текстовая часть ]
#КонецОбласти
```

27.2.2.2. Служебная часть области

Служебная часть области не имеет явных границ. В нее входят все строки от начала области, которые начинаются специальным символом «#». В служебной части описываются поля текстового макета, используемые внутри области. Если нет надобности в служебной части, она может отсутствовать.

Управляющие конструкции текстового макета выделяются цветом.

27.2.2.2.1. Структура служебной части области

Служебная часть области состоит из описаний некоторых общих параметров области и описаний форматов полей, входящих в область. Все описания, сделанные в служебной части области, действуют только в пределах области. Если в некоторой области описан формат поля и оно же встречается в другой области без описания формата, для него в другой области будет использоваться формат по умолчанию.

В начале служебной части области может следовать ключевое слово общей области `#ЗаменаСимвола А В`. С его помощью будет производиться замена символов внутри строк области;

- `А` – символ, который будет заменен;
- `В` – символ, которым будет заменен символ `А`.

Символы должны быть заключены в одинарные кавычки.

Например, если написать `#ЗаменаСимвола "@" "#"`, то внутри строк символы «@», не входящие в названия полей, будут заменены на «#».

Имеет смысл использовать данное ключевое слово в тех случаях, когда в оформлении строк макета должен входить символ `#` (его непосредственное использование указывало бы на служебный характер текста).

27.2.2.2.2. Описания полей

После описания замены следует описание форматов полей, используемых в области.

Поле создается с помощью ключевого слова `#Поле ПППП`, где `ПППП` – имя поля, формат которого будет описываться.

Далее на следующих строках размещаются ключевые слова для описания поля. Описание поля действует до начала описания следующего поля.

`#Выравнивание {Левое | Правое | Центр | ПоШирине}` – указывает выравнивание поля в пределах отведенных для него знакомест.

[Копировать в буфер обмена](#)

```
#Поле ВидРаботы
#Выравнивание ПоШирине
```

`#Формат <Форматная строка>` – указывает форматную строку для вывода поля.

[Копировать в буфер обмена](#)

```
#Поле Дата1
#Формат "ДФ=dd.MM.yy"
#Поле Вр1
#Формат "ЧЦ=4; ЧДЦ=0; ЧРД=."
```

`#Забивать <Параметр>` – указывает необходимость забивки области, отведенной для поля, символами «#». Если `<Параметр> = Истина` и содержимое поля не помещается в отведенное место, будет выполняться забивка.

[Копировать в буфер обмена](#)

```
#Поле ИтогоОтчета
#Формат "ЧЦ=18; ЧДЦ=2; ЧРД=."
#Забивать Истина
```

27.2.2.3. Текстовая часть области

Текстовая часть области состоит собственно из строк текстового макета. Она начинается после последней строки, принадлежащей служебной части области, и продолжается до конца области. Внутри строк текстового макета можно указать имя поля: `{ПППП}`.

Количество знакомест, отводимых под поле, соответствует количеству знакомест, указанных скобками (включая скобки). Если размер поля составляет один символ, допускается использование одной скобки. Например:

[Копировать в буфер обмена](#)

```
! [Имя ] ! [Код ] ! [Наименование ] !
```

Если имя поля прижато к левой скобке, считается, что поле имеет левое выравнивание, а если к правой скобке, то правое выравнивание. Если пробелы есть слева и справа от имени поля, оно будет центрироваться в пределах отведенных под него знакомест.

Если имя поля больше, чем количество знакомест, отводимых под поле, имя поля указывается с помощью ключевого слова `Поля`.

Пример:

```
#Поле Номер
#Формат "ЧЦ=3; ЧДЦ=0"
#Поле Код
#Формат "ЧЦ=5; ЧДЦ=0"
#Поле Пометка
#Формат "ЧЦ=1"
```

[Копировать в буфер обмена](#)

```
! [ ] [ Код ]! [ Наименование ]! [ ]! #Поля Номер Пометка
```

Параметры в ключевом слове **Поля** указываются только для тех полей, имена которых не указаны в теле макета в скобках.

27.2.2.3.1. Автоматический перенос

Если текст не умещается в отведенный размер поля, может возникнуть необходимость использовать автоматический перенос текста на новую строку. Для этого предлагается использовать указание расположения поля в макете в фигурных и угловых скобках:

- `{pppp}` – область поля ограничена фигурными скобками. Они показывают, что автоперенос текста, хранящегося в поле `pppp`, возможен на эту строку, и указывают область для не поместившегося ранее в поле `pppp` текста. При этом, даже если в строке нет необходимости выводить ни одно из полей, строка все равно будет выведена.
- `<pppp>` – область поля ограничена угловыми скобками. Они показывают, что автоперенос текста, хранящегося в поле `pppp`, возможен, и указывают область для не поместившегося на предшествующей строке в поле `pppp` текста. При этом, если все поля, имеющиеся в строке, указаны в угловых скобках, но ни одно из них не потребовалось для вывода текста, строка не будет выведена. Если при выводе текста из поля в строку, в которой поле размещено в угловых скобках, текст все равно не уместится, такая строка будет размножена до тех пор, пока не удастся вывести весь текст из поля.

Пример:

```
! [ YY ]! [ XX ]! [ ZZ ]!
! ! { XX }! !
! ! <XX >! !
```

[Копировать в буфер обмена](#)

В этом примере указывается, что поле `xx` размещается на первой строке. На второй строке размещается та часть поля `xx`, которая не поместилась на первой строке. На третьей строке размещается часть поля `xx`, которая не поместилась ни на первой, ни на второй строке. При этом если в поле `xx` для третьей строки текста не осталось, она совсем не выводится, а если осталось, то третья строка будет выводиться до тех пор, пока не закончится текст из поля `xx`.

27.2.2.4. Описание ключевых слов текстового макета

Область (Area)

Описание:

Указывается начало области макета, а если предшествующая область не закончена, то конец предшествующей.

Синтаксис:

```
#Область <Имя области> <Код языка>
```

Параметры:

```
<Имя области> Обязательный
```

Указывается имя области, по которому производится ее получение методом `ПолучитьОбласть()` текстового документа.

```
<Код языка> Необязательный
```

Указывается код языка области. Если предполагается использование конфигурации на нескольких языках, то в текстовом макете можно указать несколько областей с одинаковыми именами, но тогда каждая область должна содержать код языка. Если код языка не указан, то такая область выбирается в том случае, когда область запрашивается с кодом языка, отсутствующим в описании областей.

КонецОбласти (EndOfArea)

Описание:

Указывается явный конец области макета.

Синтаксис:

```
#КонецОбласти
```

ЗаменаСимвола (ReplaceChar)

Описание:

С помощью данного ключевого слова будет производиться замена символов внутри строк области.

Синтаксис:

```
#ЗаменаСимвола <Заменяемый символ> <Символ замены>
```

Параметры:

```
<Заменяемый символ> Обязательный
```

В одинарных кавычках указывается символ, который будет заменен.

```
<Символ замены> Обязательный
```

Указывается символ, на который будет заменен `<Заменяемый символ>`.

Поле (Field)

Описание:

Указывается поле, для которого необходимо указать ключевые слова форматирования.

Синтаксис:

```
#Поле <Имя поля>
```

Параметры:

<Имя поля> Обязательный

Имя поля.

Выравнивание (Align)

Описание:

Указывается выравнивание при выводе содержимого поля.

Синтаксис:

#Выравнивание <Параметр выравнивания>

Параметры:

<Параметр выравнивания> Обязательный

Значение выравнивания поля. Может принимать следующие значения:

- Лево (Left),
- Право (Right),
- Центр (Center),
- ПоШирине (Justify).

Формат (Format)

Описание:

Указывается формат вывода поля.

Синтаксис:

#Формат <Форматная строка>

Параметры:

<Форматная строка> Обязательный

Определяет формат представления значения поля.

Пример:

"ДФ=dd.MM.yy"

[Копировать в буфер обмена](#)

Забивать (Block)

Описание:

Если значение параметра *Истина*, то устанавливается забивка, при которой в поле, значение которого не помещается в отведенное место, выводится символ «#» во всем отведенном месте.

Синтаксис:

#Забивать <Параметр>

Параметры:

<Параметр> Необязательный

Определяет необходимость забивки поля. Может принимать значения:

- Истина (True),
- Ложь (False).

Поля (Fields)

Описание:

Указывается список полей, имена которых нельзя указать в месте их расположения.

Синтаксис:

#Поля <Имя поля 1> <Имя поля 2> ... <Имя поля N>

Параметры:

<Имя поля> Обязательный

Имя поля. Обязательно указывать все имена полей, которые не указаны в тексте области макета.

27.2.2.5. Пример печати расходной накладной

Предполагается, что объект конфигурации *Документ* с именем *Расходная накладная* имеет в списке макетов текстового типа макет с именем *ПечатьТекст*:

#Область Шапка RU

Расходная накладная N [НомерДокумента]

От: [От]

Кому: [Кому]

 | N| Наименование | Цена |Штук| Сумма |

#Область Строка RU

#Поле Цена

#Формат "Цц=10; ЧДц=2; ЧРД=."

#Поле Штук

#Выравнивание Право

[Копировать в буфер обмена](#)


```

#Формат "ЧЦ=4; ЧДЦ=0; ЧРД=."
#Поле Сумма
#Формат "ЧЦ=12; ЧДЦ=2; ЧРД=."
|-----|
|[ ]|[Наименование ]|[ Цена ]|[ ]|[ Сумма ]|#Поля Номер Штук
| |<Наименование >| | | |
#Область Подвал RU
#Поле ИтогоШтук
#Выравнивание Право
#Формат "ЧЦ=4; ЧДЦ=0; ЧРД=."
#Поле ИтогоСумма
#Выравнивание Право
#Формат "ЧЦ=12; ЧДЦ=2; ЧРД=."
=====
Итого [ ] [ИтогоСумма] #Поля ИтогоШтук
Директор: [Директор ]

```

Макет содержит следующие области:

- Шапка – для вывода заголовка отчета;
- Строка – для вывода табличной части;
- Подвал – для вывода итоговых данных.

В форме документа размещена кнопка *Печать*. Ее нажатие вызывает событие *Нажатие()*, процедура-обработчик которого размещена в модуле формы.

```

Процедура ПечатьВТекст(Кнопка)
ТекДок = Новый ТекстовыйДокумент();
ПечатьТекст(ТекДок);
ТекДок.Показать();
КонецПроцедуры

```

[Копировать в буфер обмена](#)

В процедуре создается текстовый документ *ТекДок* и вызывается процедура *ПечатьТекст(ТекДок)*, которая заполняет текстовый документ на основе данных расходной накладной. Процедура расположена в модуле документа. После заполнения документа он выводится на экран.

Текст процедуры ПечатьТекст():

```

Процедура ПечатьТекст(ТекДок) Экспорт
// Получение макета
Макет = ПолучитьМакет("ПечатьТекст");
// Установим код языка
Макет.КодЯзыкаМакета = "RU";
// Заголовок
Область = Макет.ПолучитьОбласть("Шапка");
Область.Параметры.НомерДокумента = Номер;
Область.Параметры.От = Формат(Дата, "ДФ=dd.MM.yyyy");
Область.Параметры.Кому = Контрагент;
ТекДок.Вывести(Область);
// Обработка табличной части "Состав"
СтрИтого = Новый Структура("ИтогоШтук, ИтогоСумма", 0, 0);
Для Каждого СтрСостава Из Состав Цикл
Область = Макет.ПолучитьОбласть("Строка");
Область.Параметры.Номер = СтрСостава.НомерСтроки;
Область.Параметры.Наименование = СтрСостава.Номенклатура;
Область.Параметры.Штук = СтрСостава.Количество;
Область.Параметры.Цена = СтрСостава.Цена;
Область.Параметры.Сумма = СтрСостава.Сумма;
ТекДок.Вывести(Область);
СтрИтого.ИтогоШтук = СтрИтого.ИтогоШтук + СтрСостава.Количество;
СтрИтого.ИтогоСумма = СтрИтого.ИтогоСумма + СтрСостава.Сумма;
КонецЦикла;
// Подвал
Область = Макет.ПолучитьОбласть("Подвал");
Область.Параметры.Заполнить(СтрИтого);
РС = РегистрыСведений.ОтветственныеСотрудники;
Область.Параметры.Директор = РС.ПолучитьПоследнее(Дата).Директор;
ТекДок.Вывести(Область);
КонецПроцедуры

```

[Копировать в буфер обмена](#)

27.2.3. Редактирование текстов шаблонов

Помимо возможностей текстового редактора модулей, редактор текстов шаблонов позволяет создавать и редактировать имеющиеся шаблоны (подробнее о шаблонах см. [здесь](#)).

27.2.4. Редактор текста запросов

В данном режиме текстовый редактор помимо основных возможностей обладает рядом дополнительных.

Синтаксические конструкции языка запросов выделяются цветом. Описание языковых конструкций см. [здесь](#) или раздел «Работа с запросами» справки по встроенному языку.

В список доступных команд текстового редактора добавляются команды установки и снятия комментария.

В режиме 1С:Предприятие пользователям, обладающим административными правами, предоставляется возможность вызова конструктора запросов.

Помимо текста запроса выполняется редактирование текстов запросов для текстового документа, у которого установлено расширение [Язык запросов](#); для поля текстового документа с установленным расширением [Язык запросов](#).

27.3. Редактор командного интерфейса конфигурации

Редактор командного интерфейса конфигурации позволяет настроить начальный порядок разделов в панели разделов (см. [здесь](#)) и начальную видимость разделов в разрезе ролей.

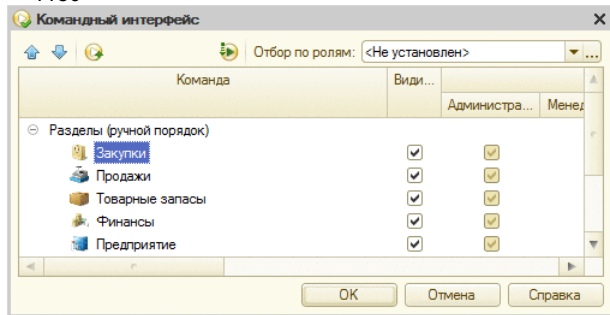


Рис. 376. Редактор «Командный интерфейс конфигурации»

В колонке **Команда** имеется возможность задавать порядок расположения разделов.

В колонках **Видимость** и **Видимость по ролям** задается видимость команд открытия подсистем по умолчанию.

27.3.1. Общие правила установки видимости

В табличном поле размещается общая колонка управления видимостью (колонка **Видимость**) и столько колонок, сколько ролей определено в конфигурации.

Изначально порядок отображения команд и настройки их видимости устанавливаются системой. Однако при этом можно вручную настраивать видимость для всех команд или отбирать команды по ролям, которым они доступны.

Установка видимости объекта в командном интерфейсе может осуществляться несколькими способами:

- Редактирование колонки **Видимость**. В этом случае видимость команды задается сразу для всех ролей. Видимость команды для конкретной роли будет определяться по состоянию колонки **Видимость** в том случае, если для конкретной роли выбрано особое состояние флажка видимости (см. видимость подсистемы **Продажи** на [рис. 376](#)).
- Указание конкретного значения (установленное или сброшенное) видимости для конкретной роли означает, что команда будет по умолчанию видна (или не видна – в зависимости от состояния флажка) для данной роли. Общее состояние (колонка **Видимость**) в этом случае игнорируется. Так, на [рис. 376](#) для роли **Администратор** видимость раздела **Товарные запасы** отключена (несмотря на то, что общая видимость этого раздела включена), а для роли **Менеджер по продажам** по умолчанию видима команда перехода к подсистеме **Предприятие** (несмотря на то, что по умолчанию видимость этого раздела отключена).

Возможно множественное изменение видимости сразу для нескольких команд в списке.

Для того чтобы в списке **Команда** показать только видимые команды, следует нажать кнопку **Скрыть невидимые по умолчанию**.

27.3.2. Отбор по ролям

В поле **Отбор по ролям** можно задать несколько ролей, которые будут определять текущее отображение списка команд. В список будут включаться только команды, доступные этим ролям. Для отключения отбора нужно выбрать **Не установлен**. Список выбора позволяет быстро включать один из нескольких последних установленных отборов.

27.4. Настройка рабочей области начальной страницы

27.4.1. С помощью редактора рабочей области начальной страницы

Данный редактор позволяет настроить общую схему расположения форм на начальной странице, а также состав форм, которые могут быть отображены на начальной странице.

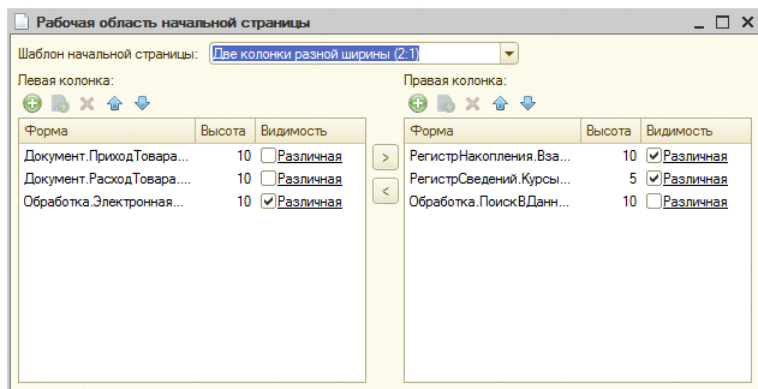


Рис. 377. Редактор «Рабочая область начальной страницы»

Общую схему расположения форм на начальной странице можно задать с помощью поля **Шаблон начальной страницы**:

- **Одна колонка** – формы будут отображаться в одну колонку.
- **Две колонки одинаковой ширины** – для форм будут доступны две колонки одинаковой ширины.
- **Две колонки разной ширины (2:1)** – будут отображаться две колонки, причем левая колонка будет в два раза шире правой.

После выбора необходимых форм можно указать порядок их расположения на начальной странице, а также высоту каждой формы (колонка **Высота**).

Редактор видимости работает аналогично другим ролевым редакторам свойств (см. [здесь](#)).

При разработке интерфейса начальной страницы следует обращать особое внимание на то, чтобы этот интерфейс содержал наиболее важные формы. На начальной странице нужно размещать те формы, с которыми наиболее часто выполняется работа пользователя с тем или иным составом ролей.

Следует помнить, что формы, на которые у пользователя нет прав, не будут отображены на начальной странице вне зависимости от состояния колонки **Видимость**.

27.4.2. Программная

Настройку состава и положения форм на начальной странице (рабочем столе) можно выполнить также из встроенного языка. Для выполнения настройки следует воспользоваться методами `ПолучитьСоставФорм()` и `УстановитьСоставФорм()` объекта `НастройкиНачальнойСтраницы`. Таким образом, имеется возможность получить текущее состояние начальной страницы (рабочего стола) клиентского приложения, модифицировать этот состав необходимым образом и установить обратно.

Настройки состава и расположения форм начальной страницы (рабочего стола) хранятся в стандартном хранилище системных настроек. Ключ настроек – `Общее/НастройкиНачальнойСтраницы` (см. [здесь](#)). Чтобы изменить настройки необходимо в начале прочитать их из хранилища:

НастройкиНачальнойСтраницы = ХранилищеСистемныхНастроек.Загрузить ("Общее/НастройкиНачальнойСтраницы");

Затем настройки начальной страницы (рабочего стола) можно модифицировать и поместить обратно в хранилище:

Копировать в буфер обмена

ХранилищеСистемныхНастроек.Сохранить ("Общее/НастройкиНачальнойСтраницы", "", НастройкиНачальнойСтраницы);

После сохранения нового состава форм, необходимо выполнить перестроение интерфейса клиентского приложения с помощью вызова метода ОбновитьИнтерфейс().

Среди разнообразных вариантов применения механизма, можно выделить несколько основных сценариев использования:

- Необходимо предложить пользователю добавить новую форму на начальную страницу (рабочий стол). Это можно сделать с помощью программного кода, который вызывается из обработчика события ПриНачалеРаботыСистемы. После изменения настроек начальной страницы (рабочего стола) необходимо обновить интерфейс, чтобы изменения стали доступны.
- Для каких-либо пользователей необходимо поддерживать фиксированный набор форм, который определяется не на этапе конфигурирования, а некоторыми данными информационной базы. В этом случае настройку начальной страницы (рабочего стола) рекомендуется выполнять в обработчике события УстановкаПараметровСеанса модуля сеанса. В этом случае не требуется вызывать метод ОбновитьИнтерфейс(), т. к. считывание настроек начальной страницы (рабочего стола) выполняется после выполнения данного обработчика.
- Необходимо однократно изменить состав форм начальной страницы (рабочего стола) для всех пользователей информационной базы. В этом случае рекомендуется написать программный модуль, который переберет настройки начальной страницы (рабочего стола) для всех пользователей и внесет в них необходимые изменения. Следует помнить, что изменение интерфейса, выполненные таким образом, будут доступны пользователю только при следующем входе в систему.

Рассмотрим пример, в рамках которого необходимо при каждом начале работы системы принудительно размещать на начальной странице две формы: форму списка справочника Товары и стандартную форму полнотекстового поиска (формаПоиска). Для реализации примера необходим общий модуль с установленными признаками Сервер и Вызов сервера. Пусть общий модуль имеет имя РаботаСИнтерфейсом. В общем модуле необходимо поместить следующую процедуру:

Копировать в буфер обмена

```
Процедура ПростаяНачальнаяСтраница() Экспорт
НачальнаяСтраница = Новый НастройкиНачальнойСтраницы;
СоставФорм = Новый СоставФормНачальнойСтраницы;
СоставФорм.ЛеваяКолонка.Добавить ("Справочник.Товары.ФормаСписка");
СоставФорм.ПраваяКолонка.Добавить ("ФормаПоиска");
НачальнаяСтраница.УстановитьСоставФорм(СоставФорм);
ХранилищеСистемныхНастроек.Сохранить ("Общее/НастройкиНачальнойСтраницы", "", НачальнаяСтраница);
КонiecПроцедуры
```

В модуле управляемого приложения необходимо разместить обработчик события ПриНачалеРаботыСистемы:

Копировать в буфер обмена

```
Процедура ПриНачалеРаботыСистемы()
РаботаСИнтерфейсом.МинимальныйИнтерфейс();
ОбновитьИнтерфейс();
КонiecПроцедуры
```

В результате, вне зависимости от того, какие настройки начальной страницы (рабочего стола) настройки осуществил пользователь в предыдущем сеансе работы, каждый новый сеанс будет начинаться с фиксированных настроек начальной страницы (рабочего стола).

При формировании настроек начальной страницы (рабочего стола) можно указывать как полные имена форм объектов конфигурации (например, Справочник.Товары.Форма.ИмяФормы), так и стандартные имена форм (например, Справочник.Товары.ФормаСписка). Также нужно помнить, что программная настройка начальной страницы (рабочего стола) не позволяет изменить шаблон начальной области (рабочего стола). При программной настройке будет использоваться тот шаблон начальной страницы (рабочего стола), который выбран при настройке начальной страницы (рабочего стола) при помощи соответствующего редактора (см. [здесь](#)).

27.5. Редактор командного интерфейса основного раздела

Редактор позволяет настроить состав команд каждой командной панели, порядок отображения и видимость элементов командного интерфейса по ролям.

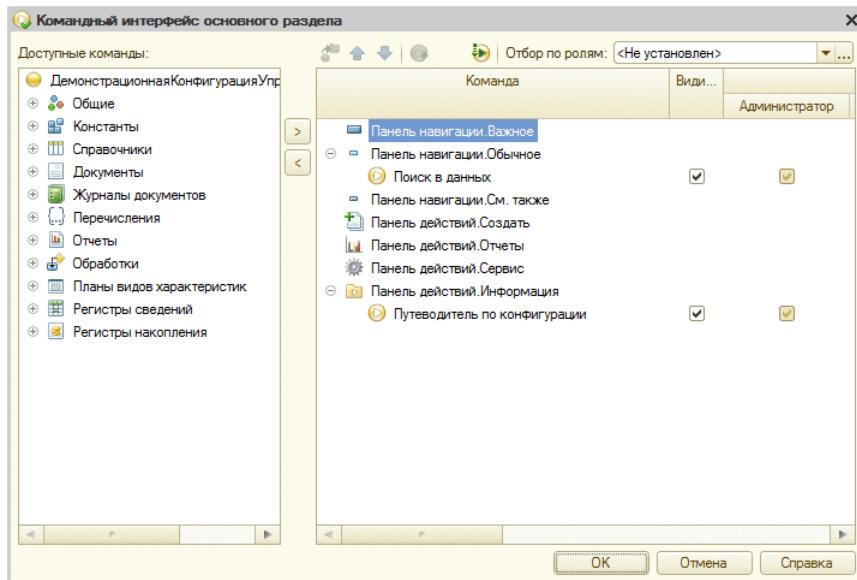


Рис. 378. Редактор «Командный интерфейс основного раздела»

В редакторе располагаются два табличных поля. В одном (справа) выполняется собственно редактирование командного интерфейса, а другое (слева) содержит иерархический список доступных команд конфигурации, которые могут быть добавлены в интерфейс основного раздела. Команды, которые попадают в список **Доступные команды**, отвечают следующим условиям:

- команда не имеет параметра;
- группа, которая указана для команды, относится к панели навигации или панели действий.

В табличном поле редактора интерфейса, в колонке **Команда**, выводится структура команд, сгруппированных по группам панели действий (**Важное**, **Обычное**, **См. также**), панели навигации (**Создать**, **Отчеты**, **Сервис**) и группы команд, определенных в ветке дерева метаданных **Общие – Группы команд**. В табличном поле размещена колонка **Общая видимость команды**. Также система размещает колонки настройки видимости команд для каждой определенной в конфигурации роли.

Для выбора команды следует раскрыть нужную ветвь таблицы доступных команд и выбрать команду. Чтобы перенести команду в командный интерфейс, следует указать команду и нажать кнопку **Добавить команду в основной раздел** (или нажать клавишу **Enter**, или дважды щелкнуть мышью строку с командой). Выбранная команда переносится в панель и группу, определенную для команды при настройке текущего объекта метаданных (свойство **Группа**), команда которого переносится. Если кнопка **Добавить команду в основной раздел** недоступна, то выбранную команду перенести в основной раздел нельзя. Например, если для перечисления не установлено свойство **Использовать**

Изначально порядок отображения команд в командном интерфейсе рабочего стола и настройки их видимости устанавливаются системой. При этом можно настраивать видимость как для всех команд или отобразить команды по ролям, которым они доступны.

Можно использовать кнопку [Переместить команду](#) для перемещения команд между группами. Также команды можно перемещать, перетаскивая их на нужное место мышью в пределах групп одной панели. Для установки порядка команд в группе нужно использовать кнопки перемещения вверх и вниз.

Нажатие кнопки [Установить свойства по умолчанию](#) заменяет настройки видимости команды и вхождение в группу на настройки, установленные системой по умолчанию.

Если в конфигурации нет ни одной подсистемы, то редактор имеет другой вид:

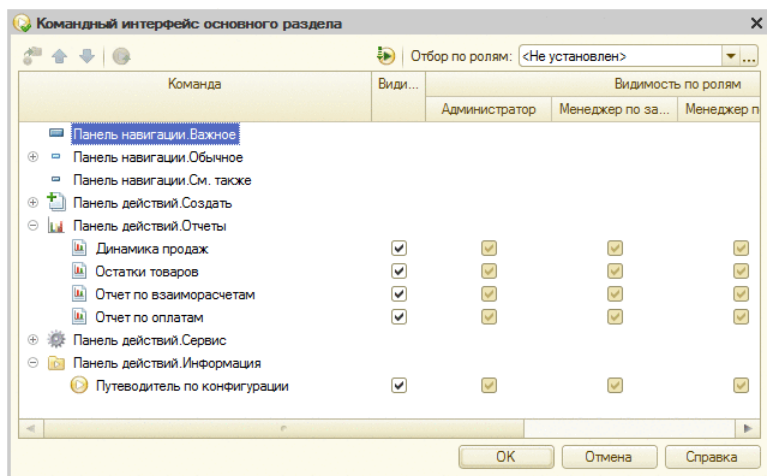


Рис. 379. Нет подсистем

В этом случае на рабочем столе доступны все команды системы, и необходимо только расставить доступность команд по умолчанию.

Общие правила управления видимостью см. [здесь](#). Описание отбора по ролям см. [здесь](#).

27.6. Редактор командного интерфейса

Редактор командного интерфейса позволяет настроить состав команд каждой командной панели, порядок отображения и видимость элементов командного интерфейса по ролям.

В табличном поле в первой (слева) колонке выводится полный список команд, сгруппированный по группам (как системные *Важное*, *Обычное*, *См. также*, *Отчеты*, *Сервис* и др., так и группы, определенные в ветке дерева метаданных *Группы команд*) панелей действий и навигации. Затем в табличном поле следует колонка для управления общей видимостью команд.

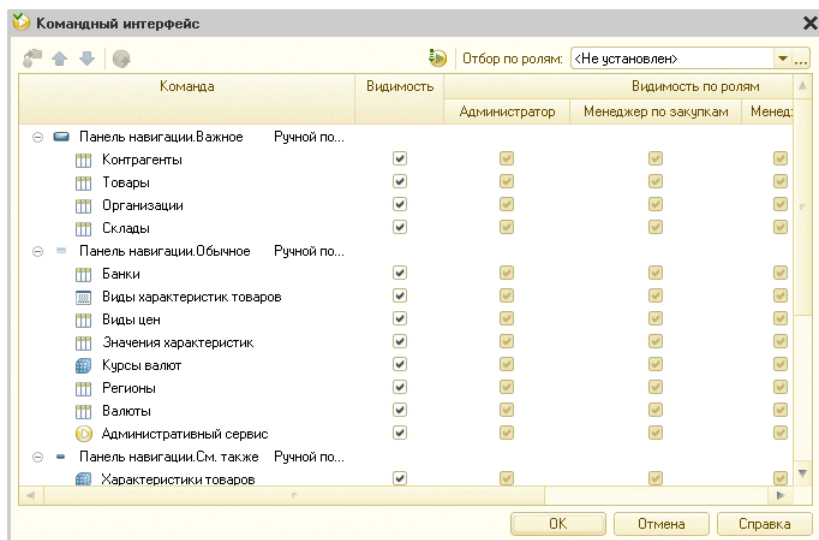


Рис. 380. Редактор «Командный интерфейс»

Система устанавливает порядок команд автоматически. При необходимости ручной настройки порядка отображения команд в группах следует использовать кнопки перемещения или перетаскивать команды. При этом рядом с группой будет отображаться строка *Ручной порядок команд*. Чтобы отменить изменения порядка, нужно у группы в контекстном меню выбрать пункт контекстного меню [Восстановить автоматический порядок команд](#).

Можно использовать кнопку [Переместить команду](#) для перемещения команд между группами. Также команды можно перемещать, перетаскивая их на нужное место мышью в пределах групп одной панели. Для установки порядка команд в группе нужно использовать кнопки перемещения вверх и вниз.

Нажатие кнопки [Установить свойства по умолчанию](#) заменяет настройки видимости команды и вхождение в группу на настройки, установленные системой по умолчанию.

Общие правила управления видимостью см. [здесь](#). Описание отбора по ролям см. [здесь](#).

27.7. Редактор «Все подсистемы»

Редактор *Все подсистемы* предназначен для удобного решения следующих задач:

- редактирование командного интерфейса любой подсистемы,
- задание состава и свойств любой подсистемы,
- настройка порядка следования подсистем,
- редактирование состава подсистем.

ПРИМЕЧАНИЕ. Редактор командного интерфейса подсистемы можно также открыть из свойства подсистемы [Командный интерфейс](#) по ссылке [Открыть](#).

В списке [Подсистемы](#) устанавливается порядок подсистем в дереве конфигурации. Этот порядок не влияет на порядок следования подсистем в панели разделов.

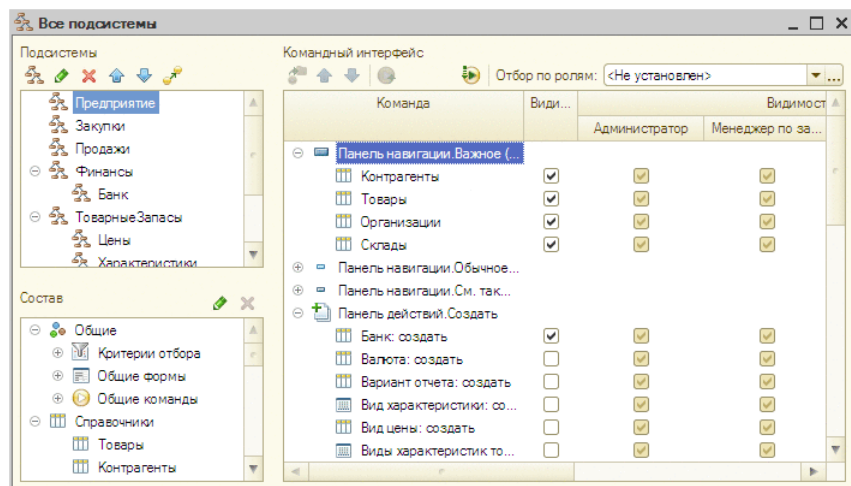


Рис. 381. Редактор «Все подсистемы»

Для того чтобы установить порядок подсистем на панели разделов, следует в списке [Подсистемы](#) перейти на корневой элемент и настроить порядок подсистем в списке [Команды подсистем](#).

Для редактирования свойств подсистемы нужно использовать палитру свойств или редактор подсистемы (выбрать подсистему и нажать кнопку [Изменить текущий элемент](#)).

Чтобы переместить подсистемы (вместе с подчиненными подсистемами), нужно выбрать подсистему и нажать кнопку [Переместить подсистему](#). В открывшемся окне выбора объекта указать подсистему, в которую будет перенесена текущая подсистема. Запрещается перенос в подсистемы, входящие в текущую подсистему.

Общие правила управления видимостью см. [здесь](#). Описание отбора по ролям см. [здесь](#).

27.7.1. Настройка состава подсистемы

Под списком подсистем находится иерархический список [Состав](#), в котором формируется состав метаданных, относящийся к текущей подсистеме списка подсистем. Следует учитывать, что в показываемый состав не включаются метаданные, относящиеся к подчиненным подсистемам, если в этих подсистемах эти метаданные не определены.

Для изменения состава следует нажать кнопку [Редактировать состав подсистемы](#). Открывается окно выбора объекта, которое содержит дерево метаданных. Объекты, у которых установлены флажки, включены в состав подсистемы.

В окне выбора нужно указать те объекты, которые должны быть включены в состав подсистемы. Нажатие кнопки **OK** закрывает окно выбора и формирует состав подсистемы.

27.7.2. Настройка командного интерфейса подсистемы

Настройка командного интерфейса подсистемы осуществляется в табличном поле, расположенном справа от списка подсистем и состава команд подсистемы.

Общие правила управления видимостью см. [здесь](#). Описание отбора по ролям см. [здесь](#).

27.8. Конструктор запросов

Конструктор запросов позволяет сформировать текст запроса в модуле и отредактировать имеющийся запрос.

Для вызова конструктора запросов нужно открыть модуль, выбрать процедуру, в которую необходимо поместить запрос, установить указатель в той части процедуры, в которой располагается или должно располагаться тело запроса, выбрать пункт [Текст – Конструктор запросов](#).

Если запроса нет, то на экран выводится вопрос: [Не найден текст запроса. Создать новый?](#) При выборе **Да** на экран выводится окно [Конструктор запросов](#). Если запрос редактируется, то на экран выводится окно [Конструктор запросов](#), содержащее данные текущего запроса.

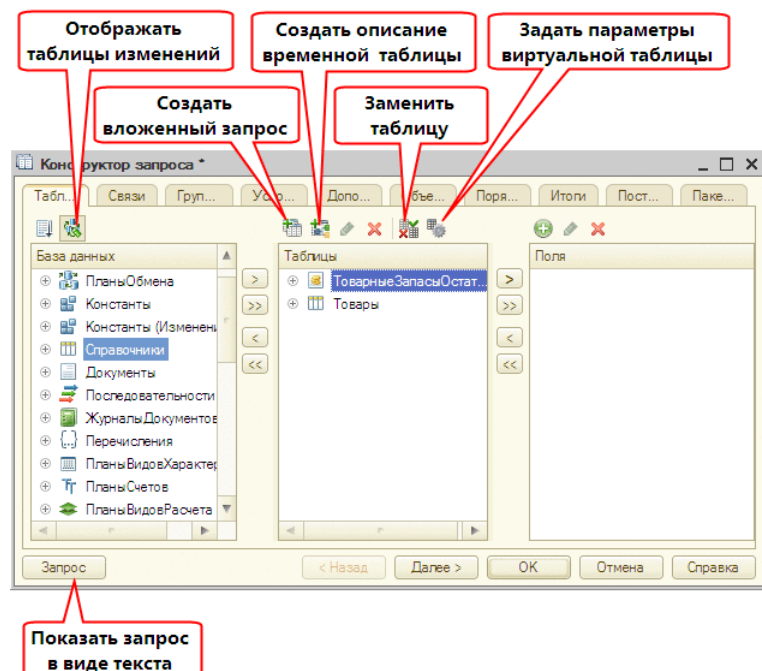


Рис. 382. Конструктор запроса в конструкторе выходной формы

С помощью кнопки [Упорядочить список](#), расположенной над списком [База данных](#), можно упорядочить список объектов. Повторное нажатие кнопки отменяет упорядочивание.

С помощью кнопки [Отображать таблицы изменений](#) можно отобразить таблицы изменений объектов конфигурации в списке объектов в виде отдельной ветки [Константы \(Изменения\)](#). Повторное нажатие кнопки скроет ветку в списке.

Для того чтобы создать вложенный запрос, нужно нажать кнопку [Создать вложенный запрос](#) (см. [рис. 382](#)). В открывшемся конструкторе запроса сформировать требуемый запрос и нажать кнопку [OK](#). Вложенный запрос отобразится в списке [Таблицы](#).

Для использования созданной временной таблицы следует нажать кнопку [Создать описание временной таблицы](#) (см. [рис. 382](#)). В отобразившейся форме ввести имя таблицы, имена нужных полей и, при необходимости, указать тип значения полей.

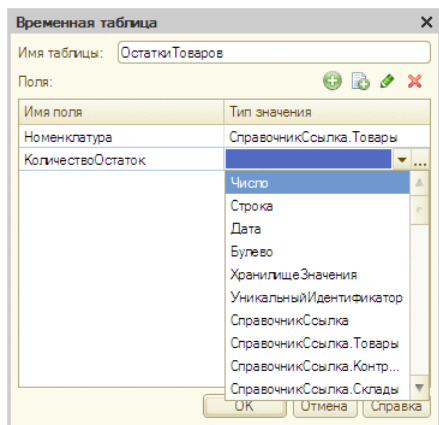


Рис. 383. Описание временной таблицы

С помощью кнопок [Далее >](#) конструктора запроса нужно последовательно пройти по закладкам и выбрать необходимые исходные данные, указать группировки и условия, установить нужный порядок и описать итоговые данные. В результате работы конструктора создадутся форма и макет, которые будут располагаться в соответствующих ветвях. По кнопке [Запрос](#) в любой момент можно открыть окно с текстом сформированного на основании указанных данных запроса.

Для корректировки данных следует использовать кнопку [< Назад](#).

На закладке [Таблицы и поля](#) выбрать нужные объекты и перенести их в разделы [Таблицы](#) и [Поля](#).

Для указания дополнительных условий можно воспользоваться режимом формирования произвольных выражений в запросе. Для этого в списке [Поля](#) контекстного меню нужно выбрать пункт [Добавить](#). На экран выводится окно произвольного выражения.

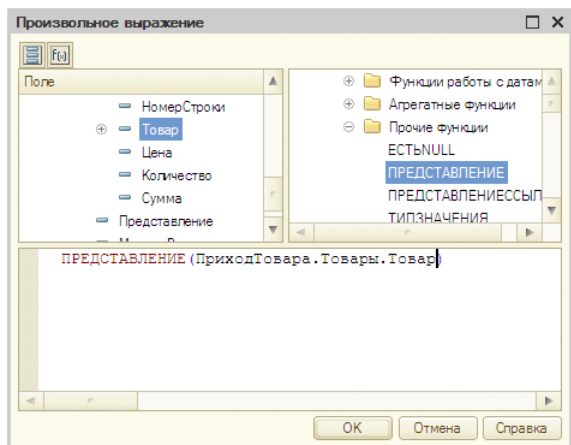


Рис. 384. Конструктор произвольного выражения

В нижнем поле формируется текст выражения. Выражение можно набирать с помощью клавиатуры. Для удобства ввода наименований реквизитов можно перетаскивать мышью нужные поля из списка полей и выбирать нужные функции языка запросов из списка, также перетаскивая их мышью в поле ввода выражения.

Если указано несколько таблиц, то в форму конструктора запроса добавляется закладка [Связи](#).

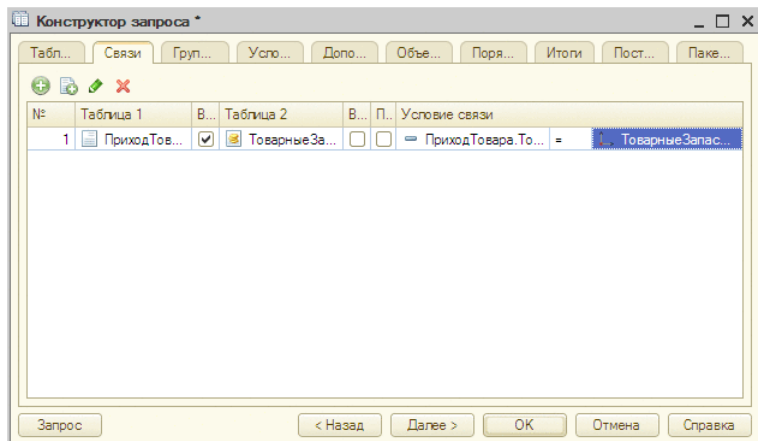


Рис. 385. Определение условий связи между таблицами

На закладке [Связи](#) формируются условия, которые накладываются на связи между полями таблиц. Для ввода нового условия нужно нажать кнопку [Добавить](#) и в колонке [Таблица1](#) выбрать одну из таблиц; в колонке [Таблица2](#) выбрать таблицу, поля которой связаны с полями первой. Ниже списка условий расположены элементы управления, с помощью которых формируется условие связи таблиц.

На закладке [Группировка](#), если требуется, нужно выбрать реквизиты, по которым будет выполнена группировка.

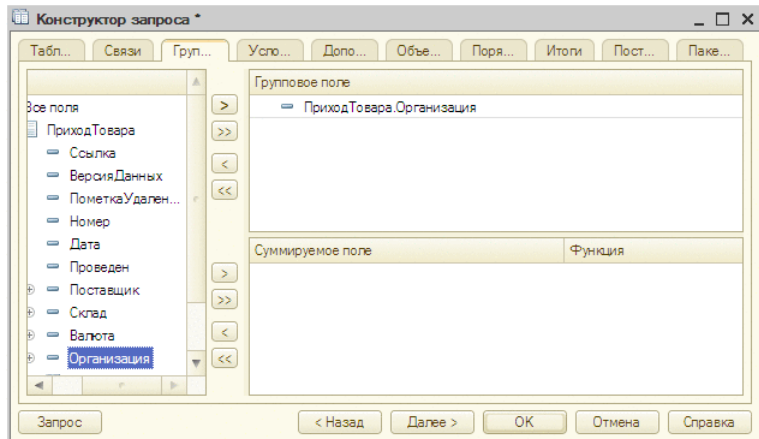


Рис. 386. Группировка результата запроса

На закладке **Условия**, если требуется, следует указать условия, по которым будет выполняться отбор исходных данных.

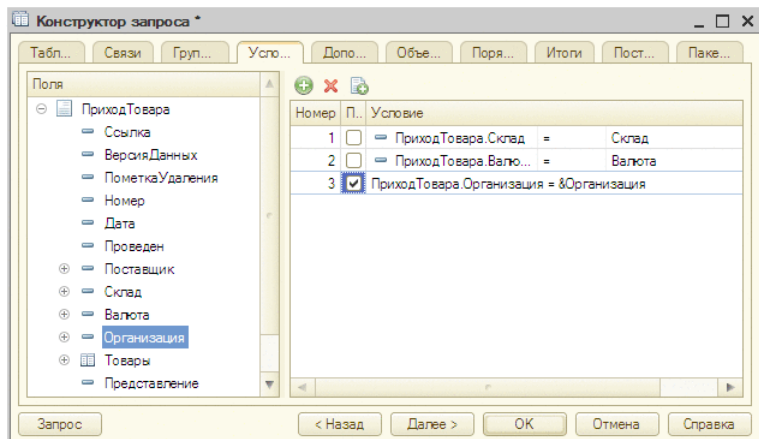


Рис. 387. Условия запроса

По каждому выбранному полю необходимо выбрать вид условия (для произвольного условия в колонке **Произвольное** установить флажок). Если флажок не установлен, то следует выбрать вид условия и указать наименование параметра. Если флажок **Произвольное** установлен, то можно воспользоваться окном формирования произвольных выражений (см. выше).

На закладке **Дополнительно** указываются дополнительные условия.

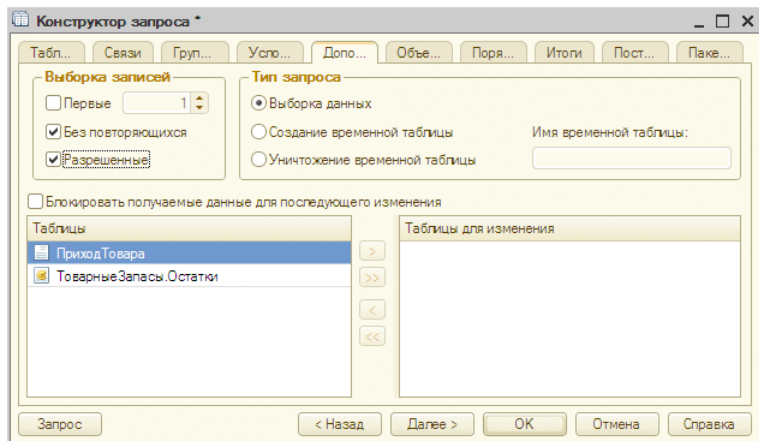


Рис. 388. Дополнительные параметры запроса

Если на закладке **Дополнительно** выбрано создание временной таблицы, то на появившейся закладке **Индекс** можно будет выбрать поля, по которым будет построен индекс.

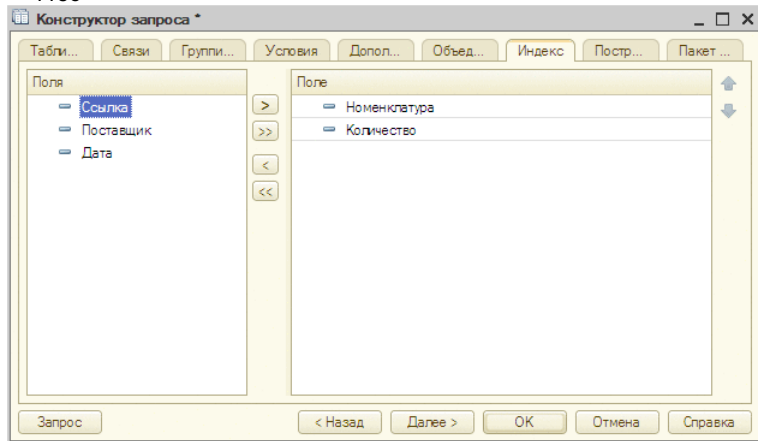


Рис. 389. Индекс временной таблицы

На закладке **Объединения/Псевдонимы**, если требуется, нужно ввести псевдонимы полей.

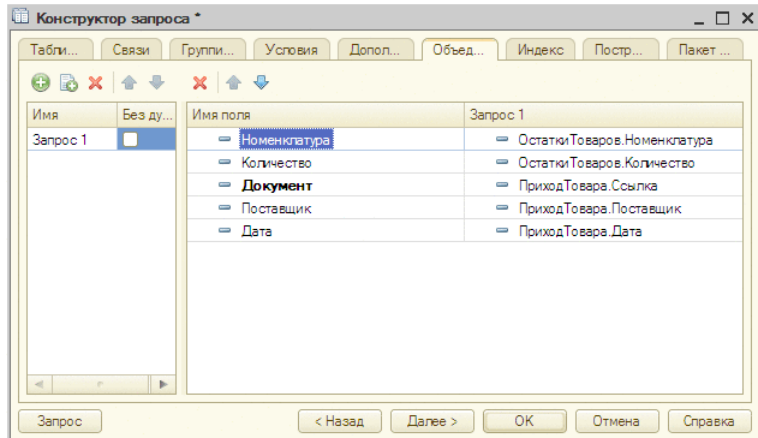


Рис. 390. Объединения/псевдонимы запроса

В таблице показано соответствие выбранных полей и исходных данных. Имена полей и соответствие можно изменить. Для изменения имени нужно выбрать поле и нажать клавишу **Enter**, ввести новое имя поля. Для изменения соответствия в колонке **Запрос** выбрать нужную строку и нажать клавишу **Enter**. В выпадающем списке выбрать нужное значение.

Если требуется выбрать только уникальные значения, то следует установить флажок в колонке **Без дубл.**

Псевдонимы полей, которые изменены пользователем, или были загружены из текста запроса, или при генерации псевдонима конструктор определил, что данный псевдоним обязателен, выделяются жирным шрифтом.

На закладке **Порядок**, если требуется, нужно указать порядок вывода полученных данных.

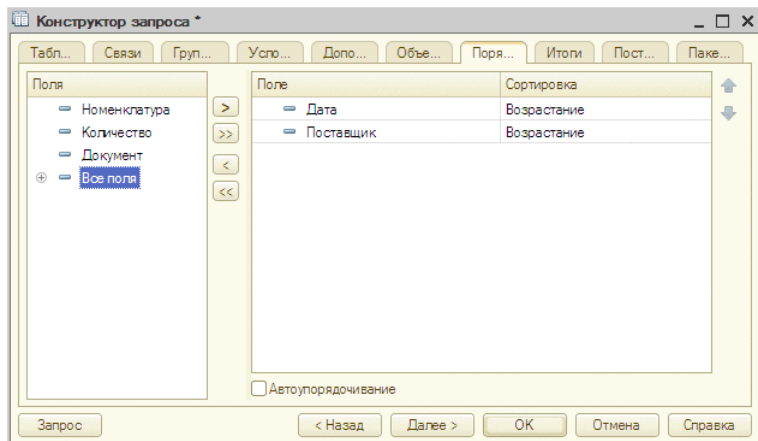


Рис. 391. Настройка упорядочивания результата запроса

Как показано на [рис. 391](#), выбран порядок вывода данных, отсортированных по дате, а в пределах одной даты выполняется сортировка по **Поставщику**.

На закладке **Итоги**, если требуется, следует указать, по каким полям нужно выводить промежуточные итоговые данные, а также указать необходимость формирования общих итогов.

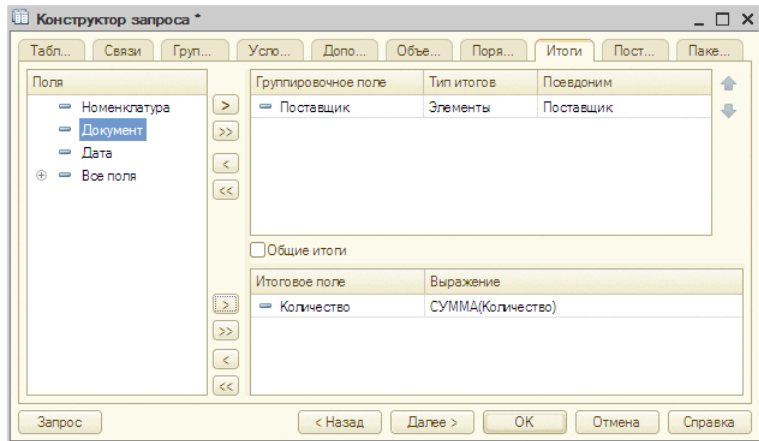


Рис. 392. Установка итогов для результата запроса

При нажатии кнопки >>, расположенной рядом с групповыми полями, в поля для группировки будут помещены все ссылочные поля. При нажатии кнопки >>, расположенной рядом с суммируемыми полями, в список суммируемых полей будут помещены все поля, имеющие числовой тип.

На закладке **Построитель** выполняется настройка построителя отчетов. Выбираются таблицы и поля, указываются условия и порядок представления, а также описываются итоговые данные.

- На закладке **Таблицы** редактируются параметры построителя отчета для виртуальных таблиц, а также отмечаются необязательные таблицы. Для редактирования параметров виртуальных таблиц необходимо выделить таблицу, для которой требуется настроить параметры, вызвать команду **Параметры виртуальной таблицы**, после чего в появившемся диалоге ввести необходимые параметры таблицы для построителя отчета. Для того чтобы отметить таблицу как необязательную, необходимо снять флажок **Обязательная** напротив имени таблицы, которую нужно пометить. Кроме того, для необязательных таблиц можно указать номер группы. Необязательные таблицы, находящиеся рядом в списке соединений и имеющие одинаковые номера групп, будут объединены в одну необязательную группу. Необязательные таблицы с различными номерами групп будут разнесены по различным группам необязательных таблиц.
- На закладке **Поля** выбираются поля, которые построитель отчета будет использовать в качестве доступных полей для вывода в отчет.
- На закладке **Условия** выбираются поля, которые построитель отчета будет использовать в качестве доступных полей для отбора.
- На закладке **Порядок** выбираются поля, которые построитель отчета будет использовать в качестве доступных полей для упорядочивания результата.
- На закладке **Итоги** выбираются поля, которые построитель отчета будет использовать в качестве доступных полей для группировки отчета.

На закладке **Пакет запросов** при работе с пакетными запросами создается последовательность запросов пакета. Каждый запрос редактируется и настраивается в этом же окне, а на всех предыдущих закладках можно переключаться между настраиваемыми запросами.

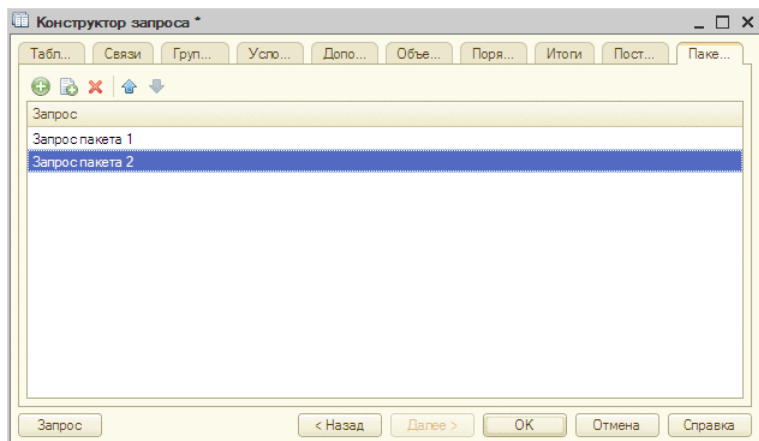


Рис. 393. Пакет запросов

По нажатию кнопки **OK** в программном модуле формируется текст запроса. Если происходит попытка открыть конструктором некорректный текст запроса, то при вызове конструктора запросов происходит автоматическая установка курсора в строку запроса, в которой обнаружена ошибка, и выдается диагностическое сообщение.

В зависимости от того, откуда вызывается конструктор запроса (система компоновки данных, запрос с обработкой результата и т. д.), в форме конструктора могут добавляться новые закладки, описание которых можно увидеть в описании того механизма, откуда вызывается конструктор запроса.

27.9. Конструктор запроса с обработкой результата

Конструктор предназначен для генерации программного кода обработки результатов запроса. Данный конструктор помогает создавать следующие варианты обработки результатов запроса:

- простой обход результатов запроса,
- вывод результатов запроса в табличный документ,
- вывод результатов запроса в диаграмму.

Для вызова конструктора предназначен пункт **Конструктор запроса с обработкой результата** контекстного меню редактора модулей.

При вызове конструктор ищет в текущей строке код, который мог быть создан конструктором. Если такой код найден, то конструктор загружает найденный код (включая имена макетов и т. д.).

Окно конструктора запроса будет отличаться от описанного выше редактора запросов (см. [здесь](#)).

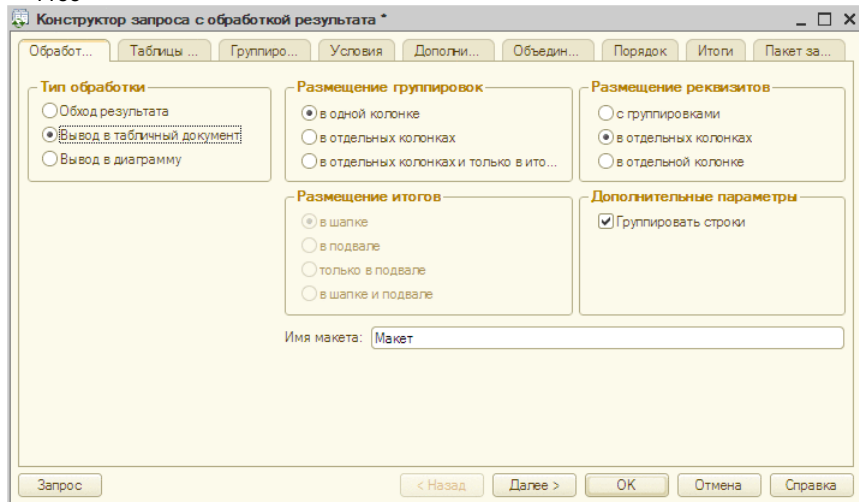


Рис. 394. Конструктор запроса с обработкой результата

В конструкторе, открытом в таком режиме:

- На закладке **Обработка результата** происходит выбор типа обработки результата запроса:
 - **Обход результата**,
 - **Вывод в табличный документ**,
 - **Вывод в диаграмму**.
- При выборе режима **Вывод в табличный документ** и **Вывод в диаграмму** отображаются параметры вывода в данные объекты.

После нажатия кнопки **OK** генерируется программный код, который будет вставлен в текущую позицию модуля. Если конструктор вызывался над уже сгенерированным кодом, старый код заменяется.

Если вывод осуществлялся в табличный документ, то в объекте метаданных, в модуле объекта или модуле формы которого он вызывался, создается макет для вывода в табличный документ. Он же модифицируется при повторном вызове конструктора. Данный макет удаляется, если ранее конструктор использовался для создания табличного документа, а теперь вызывается для генерации другого способа обработки результатов запроса. Если конструктор вызывается в общем модуле, то создается общий макет.

ПРИМЕЧАНИЕ. Если у конфигурации установлен режим редактирования конфигурации для режимов запуска в управляемом приложении, то в меню объектов метаданных недоступна команда вызова конструктора выходных форм.

27.10. Конструктор движений регистров

Конструктор движений регистров применяется для документов.

Для запуска конструктора движений регистров в окне редактирования документа, на закладке **Движения** нужно указать состав регистров, движения которых осуществляет данный документ, и нажать кнопку **Конструктор движений**. Если для выбранного документа определен состав движений хотя бы по одному регистру, то указанный пункт контекстного меню будет доступен.

При запуске конструктор запрашивает выбор регистра, по которому будет произведено формирование процедуры **ОбработкаПроведения()**, и открывает окно конструктора.

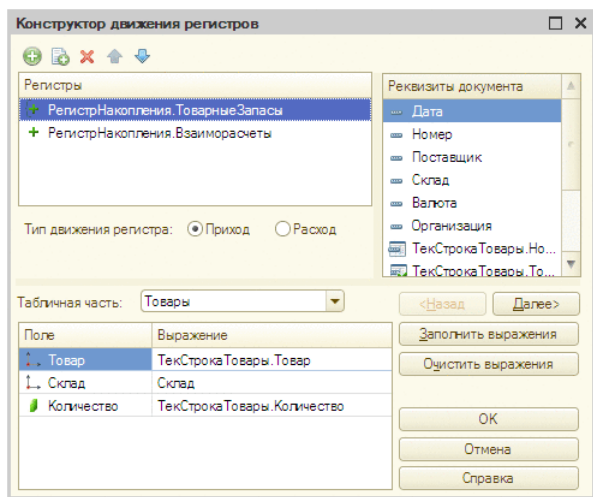


Рис. 395. Конструктор движений регистров

Первоначально список содержит только указанный регистр. Список можно изменить, добавляя регистры из списка регистров, отмеченных на закладке **Движения** окна редактирования документа.

Для каждого регистра в зависимости от его типа необходимо указать предварительные настройки: для регистров накопления остатков указать тип движения; для регистров бухгалтерии, не поддерживающих корреспонденцию, – вид корреспонденции и счет; для регистров, поддерживающих корреспонденцию, – счет дебета и счет кредита.

Если у документа есть табличные части и их данные должны влиять на состояние регистров, то включение табличной части производится в списке выбора **Табличная часть**.

Затем для каждого регистра производится заполнение формул атрибутов по данным реквизитов документа.

В список формул нужно поместить формулы, определяющие, как вычислять движения регистра по выбранным реквизитам документа.

Эти формулы можно создавать «вручную» следующим образом. Формула вводится в колонке **Формула** для каждого атрибута регистра, выделенного в списке. Можно вручную набрать ее в этом поле (а также вручную редактировать в нем созданную ранее формулу). Правильность написания формул конструктор не проверяет.

Можно также двойным щелчком в списке реквизитов документа помещать данные соответствующего реквизита в формулу. Конструктор не проверяет соответствие типов выбранных реквизитов.

Если есть хотя бы один тип, принадлежащий как реквизиту, так и измерению/ресурсу регистра и при этом их наименования совпадают, то для таких подчиненных объектов можно применить режим [Автозаполнение](#). Наименование реквизитов при нажатии кнопки [Заполнить](#) заносится в колонку [Формула](#) и может быть отредактировано.

В результате работы конструктора в модуле объекта создается процедура [ОбработкаПроведения\(\)](#). В начало процедуры конструктор помещает предупреждение: [Данный фрагмент построен конструктором. При повторном использовании конструктора внесенные вручную изменения будут потеряны!](#)

27.11. Конструктор печати

Конструктор печати предназначен для создания макета с заданными именованными областями и процедуры печати для объекта конфигурации.

Конструктор печати работает с объектами конфигурации:

- Справочники,
- Документы,
- Журналы документов,
- Планы видов характеристик,
- Планы счетов,
- Планы расчетов,
- Планы обмена.

Основные приемы работы с конструктором печати будут рассмотрены ниже на примере работы со справочником. Для других видов объектов конфигурации работа с конструктором может несколько отличаться.

Для запуска конструктора печати в окне [Конфигурация](#) нужно указать требуемый объект. В контекстном меню выбрать пункт [Конструкторы – Конструктор печати](#).

В связи с тем, что результатом работы конструктора печати является процедура печати, располагаемая в модуле менеджера объекта, конструктор сначала проверяет доступность данного модуля. Если модуль объекта имеет ограничение доступа, на экран выводится запрос пароля. После ввода пароля на экран выводится окно конструктора.

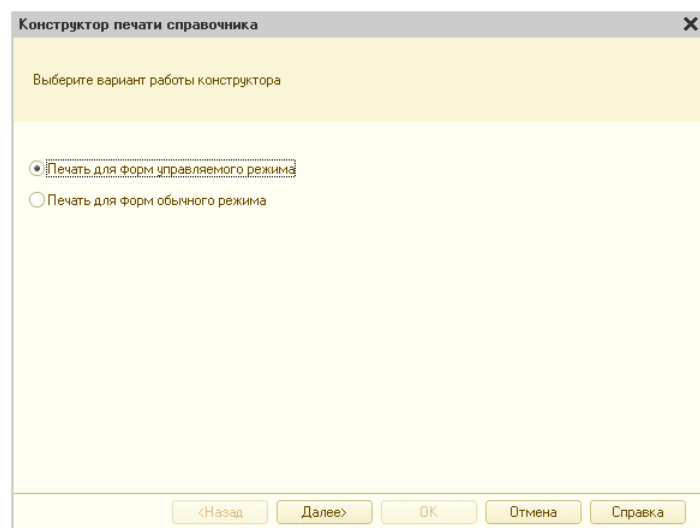


Рис. 396. Выбор варианта работы

Вначале конструктор предлагает выбрать, для какого варианта запуска будет формироваться модуль печати.

Затем конструктор предлагает создать новую команду печати или изменить существующую команду. В случае создания новой команды предлагается указать ее имя. По умолчанию для команды предлагается имя [Печать](#). Указанное имя будет использоваться в качестве:

- имени команды объекта;
- имени серверного метода модуля команды;
- имени метода менеджера объекта, который непосредственно реализует процедуру печати.

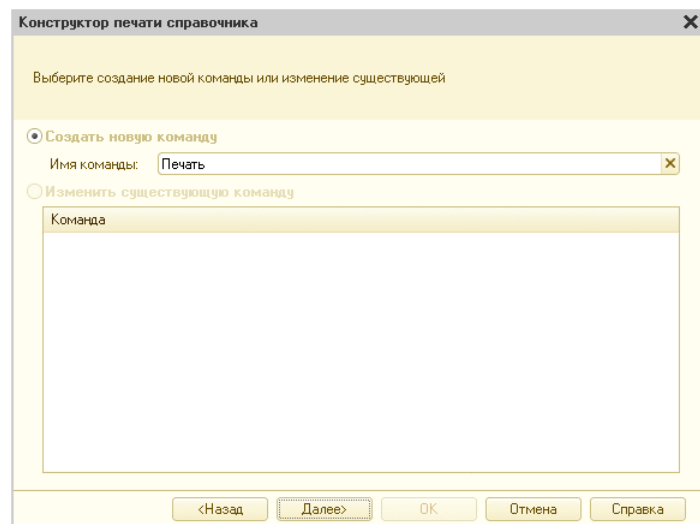


Рис. 397. Создание команды печати

Если выбрано изменение существующей команды, то в модуле команды объекта будут замещены тексты обработчика [ОбработкаКоманды\(\)](#), серверного метода с именем

команды (в данном случае – [Печать](#) ()), метода модуля менеджера объекта с именем команды и макета.

После нажатия кнопки [Далее >](#) конструктор переходит к выбору реквизитов шапки. С помощью кнопок переноса нужно сформировать список реквизитов.

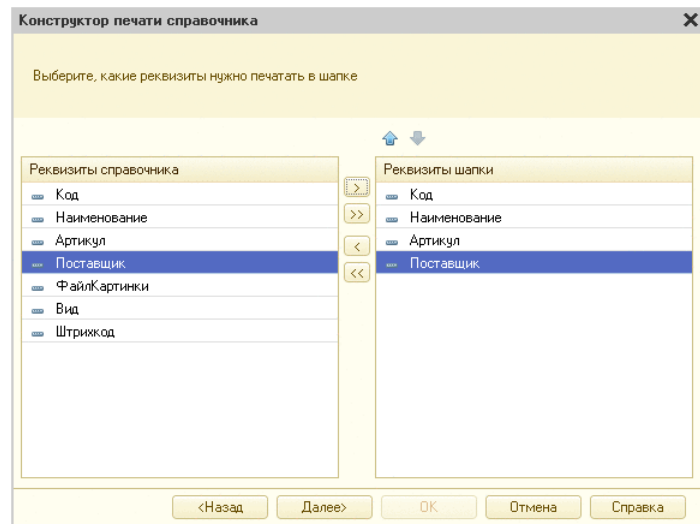


Рис. 398. Выбор реквизитов шапки

Если объект конфигурации имеет табличные части, то после нажатия кнопки [Далее >](#) конструктор переходит к выбору реквизитов очередной табличной части. Список печати формируется аналогично списку реквизитов шапки.

Если объект конфигурации имеет табличные части, то после нажатия кнопки [Далее >](#) конструктор переходит к выбору реквизитов подвала.

Затем следует указать, к какой группе будет отнесена созданная команда, и указать некоторые параметры сформированного табличного документа.

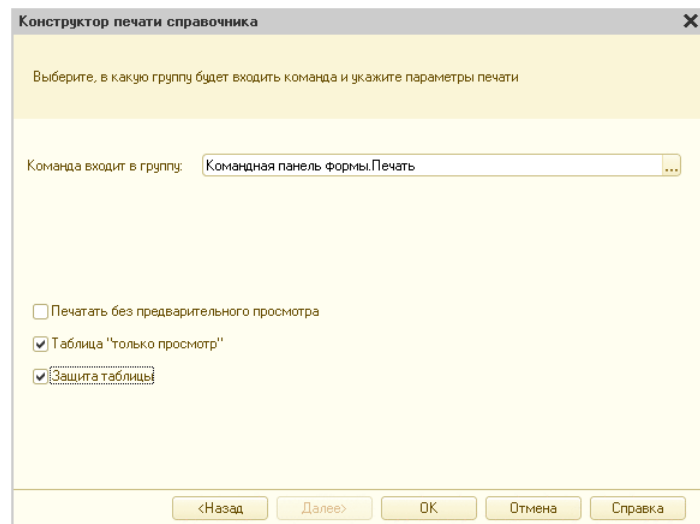


Рис. 399. Настройка конструктора печати

Нажатие кнопки **ОК** завершает работу конструктора печати.

В результате работы конструктора:

- У выбранного объекта создается макет [Макет](#) с печатной формой (имя формы соответствует имени созданной команды).
- У выбранного объекта создается команда, которая подготавливает табличный документ и вызывает серверную процедуру (с именем команды), заполняющую переданный табличный документ.
- В модуле менеджера выбранного объекта генерируется процедура, параметры которой позволяют печатать сразу несколько объектов (список). Созданная команда получает права, аналогичные праву [Просмотр](#) объекта, для которого вызывается конструктор.

27.12. Конструктор ввода на основании

Конструктор ввода на основании облегчает задачу разработки процедуры, с помощью которой будет формироваться новый объект.

Конструктор ввода на основании применяется:

- для справочников,
- документов,
- планов видов характеристик,
- планов счетов,
- планов видов расчета,
- планов обмена,
- бизнес-процессов,
- задач.

Для запуска конструктора в окне [Конфигурация](#) нужно указать требуемый объект. В контекстном меню выбрать пункт [Конструкторы – Конструктор ввода на основании](#). Если

для выбранного объекта конфигурации определен режим ввода на основании, то указанный пункт контекстного меню будет доступен.

При запуске открывается окно **Конструктор ввода на основании**.

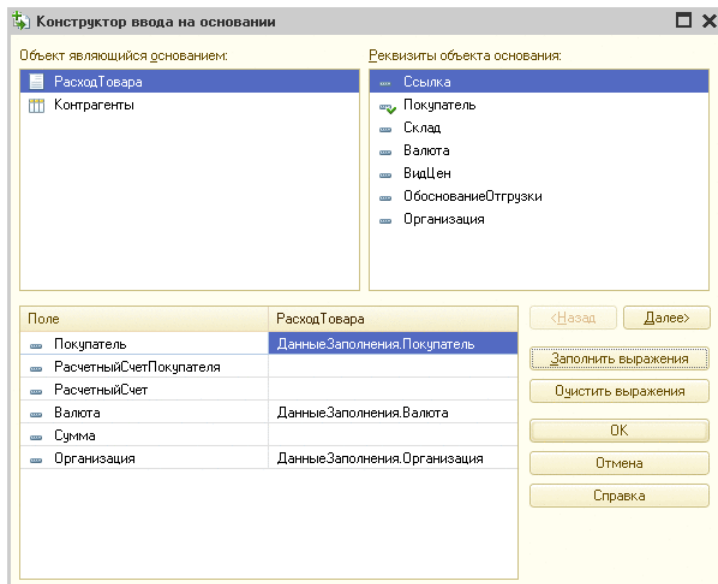


Рис. 400. Конструктор ввода на основании

В верхней части содержится список объектов-оснований; список реквизитов объекта-основания, выбранного в первом списке.

В нижней части расположен список реквизитов результирующего объекта.

В список реквизитов объекта нужно поместить формулы, определяющие, как заполнять реквизиты документа по выбранным реквизитам объекта-основания.

Эти формулы можно создавать «вручную» следующим образом. Формула определяется в поле ввода **Формула** заполнения реквизита для реквизита объекта, выделенного в списке. Можно вручную набрать ее в этом поле (а также вручную редактировать в нем созданную ранее формулу). Правильность написания формул конструктор не проверяет.

Можно также двойным щелчком в списке реквизитов объекта-основания помещать данные соответствующего реквизита в поле **Формула** заполнения реквизита, причем сразу в правильной записи. Конструктор не проверяет соответствие типов выбранных реквизитов.

Возможна и комбинация двух описанных способов: данные из списка реквизитов объектов-оснований при помещении в поле ввода замещают не всю имеющуюся там информацию, а только выделенные символы.

По кнопке **Автозаполнение после запроса** и подтверждения программа сама создаст формулы для заполнения по реквизитам объекта-основания. Ранее заполненные формулы при автоматическом заполнении не изменяются. Подбор среди реквизитов объекта-основания соответствий реквизитам документа осуществляется с учетом имен реквизитов, их идентификаторов и типов.

По кнопке **Очистить формулы** после запроса и подтверждения очищаются все формулы, созданные как автоматически, так и вручную.

Формулы, созданные для объекта-основания, выбранного в списке объектов-оснований, запоминаются при переходе к другому объекту-основанию этого списка. Поэтому можно, не прекращая работы конструктора, создать процедуры ввода на основании для нескольких документов-оснований. Можно также возвращаться к редактированию процедуры для какого-либо объекта-основания, снова выбирая его в списке.

Для прекращения работы конструктора служат кнопки **OK** и **Отмена** (соответственно с сохранением и без сохранения сделанных изменений).

Кнопка **Обновить** позволяет отразить в модуле формы сделанные изменения без выхода из окна конструктора.

В результате работы конструктора в модуле объекта создается процедура **ОбработкаЗаполнения()**. В начале процедуры конструктор помещает предупреждение: **Данный фрагмент построен конструктором. При повторном использовании конструктора внесенные вручную изменения будут потеряны!**

При повторном использовании конструктора все формулы, сформированные ранее для какого-либо документа-основания, будут появляться в списке реквизитов документа при выделении соответствующего документа-основания в списке документов-оснований. Конструктор предьявит для просмотра и редактирования формулы, сформированные при предыдущем его запуске, как с помощью кнопки **Заполнить** формулы автоматически, так и вручную. Более того, он учтет и формулы, вручную внесенные в модуль документа, если они помещены в формируемую им процедуру. Правильность этих формул конструктор не проверяет. Например, если для какого-то реквизита документа указано заполнение двумя различными реквизитами документа-основания, конструктор для показа выберет один из них (по алфавиту). Именно он останется в модуле после обновления результатов работы конструктора. Вторая строка, соответствующая тому же реквизиту, будет удалена.

27.13. Конструктор форм объектов конфигурации

Для каждого объекта конфигурации, в составе которого могут находиться подчиненные объекты типа **Форма**, при добавлении новой формы запускается **Конструктор формы**. Это специальный мастер, с помощью которого производится выбор типа формы и размещаются реквизиты объекта конфигурации. Несмотря на существенные различия типов объектов, для которых производится создание формы, конструкторы имеют много общего. Поэтому рассмотрим работу конструктора форм на примере конструктора формы документа.

При создании новой формы объекта на экран выводится конструктор формы.

Рис. 401. Конструктор формы

Группа элементов управления **Выберите тип формы** предназначена для выбора типа формы (состав определяется видом выбранного объекта конфигурации, для которого создается форма). Для объекта конфигурации **Документ** это следующие формы:

- форма документа;
- форма списка документа;
- форма выбора документа;
- произвольная форма (пустая форма).

Число форм по каждому типу форм неограниченно. Если у объекта несколько форм одного типа, то одну из них можно выбрать в качестве основной. Если при вызове формы объекта не указано явно, какую именно форму следует вызвать, на экран будет выводиться основная форма. Для указания основной формы при создании нужно установить флажок **Назначить форму основной**. В дальнейшем выбор основной формы можно изменить в окне редактирования объекта (см. [здесь](#)), на закладке **Формы**.

Выбор типа **Произвольная форма** не приводит к созданию основного реквизита. В этом случае форма имеет типовое поведение. Выбор иного типа формы влечет создание основного реквизита и определяет отличительное от типового поведение формы. Данные отличия описываются объектами, называемыми расширением формы (см. описание соответствующих объектов справки по встроенному языку).

Выбор типа формы определяет тип основного реквизита и, следовательно, поведение всей формы, а также возможный состав команд панели инструментов.

Нужно указать имя формы, синоним и комментарий.

Состав команд командных панелей определяется источником действий и типом основного реквизита формы.

Если нажать кнопку **Готово**, то конструктор самостоятельно разместит элементы управления в форме и откроет форму для редактирования.

Если необходимо изменить состав реквизитов формы, следует нажать кнопку **Далее >**. Состав реквизитов определяется основным реквизитом формы.

Рис. 402. Выбор реквизитов формы

На этом шаге конструктора формы следует выбрать реквизиты, которые необходимо разместить в форме. Выбор осуществляется в колонке **Реквизиты** установкой пометки слева от наименования элемента.

ВНИМАНИЕ! Конструктор формы не включает в список те реквизиты, чей тип не имеет визуального представления (например, **ХранилищеЗначения**).

27.13.1. Особенности конструктора форм констант

Каждая константа может иметь собственную форму редактирования. Для создания такой формы необходимо выполнить команду **Создать форму констант** контекстного меню нужной константы.

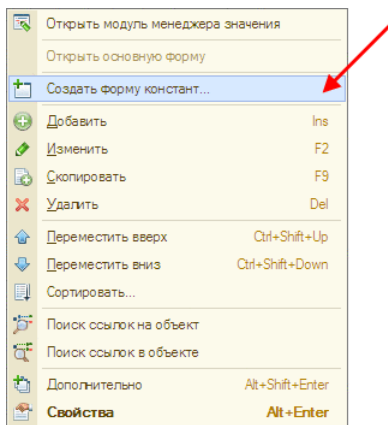


Рис. 403. Создание формы редактирования констант

После выполнения данной команды будет открыт конструктор общей формы, где в качестве данных будут перечислены все константы системы и текущая константа будет единственным выбранным реквизитом (по умолчанию).

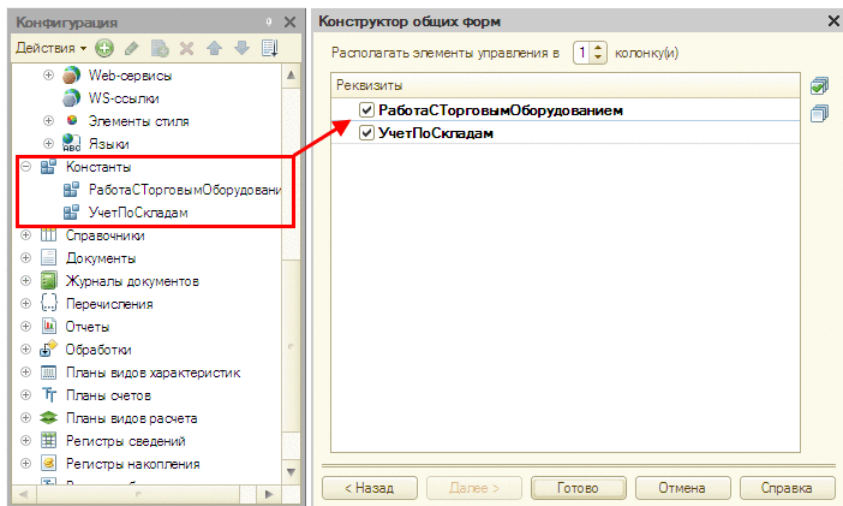


Рис. 404. Выбор редактируемых констант

Такая же форма будет автоматически создана системой, если у константы не будет заполнено свойство [Основная форма](#), но будет установлено свойство [Использовать стандартные команды](#).

Если необходимо создать форму редактирования нескольких констант, то следует создать общую форму типа [Форма констант](#), а затем на странице выбора реквизитов указать константы, которые необходимо редактировать.

Формы констант будут входить в командный интерфейс тех подсистем, куда входит сама константа (если у константы установлено свойство [Использовать стандартные команды](#)) и общая форма редактирования константы (если у общей формы установлено свойство [Использовать стандартные команды](#)).

Если для константы, у которой установлено свойство [Использовать стандартные команды](#), создается форма редактирования, то конструктор общей формы сбрасывает свойство [Использовать стандартные команды](#) для формы. Таким образом, формы редактирования константы будут попадать в командные интерфейсы только тех подсистем, куда входит сама константа.

27.14. Конструктор макета

Конструктор макета используется для создания макетов объектов конфигурации и общих макетов.

Окно конструктора выводится на экран при выполнении пункта [Действия – Добавить](#), если выбрана ветвь подчиненного объекта [Макеты](#) или [Общие макеты](#).

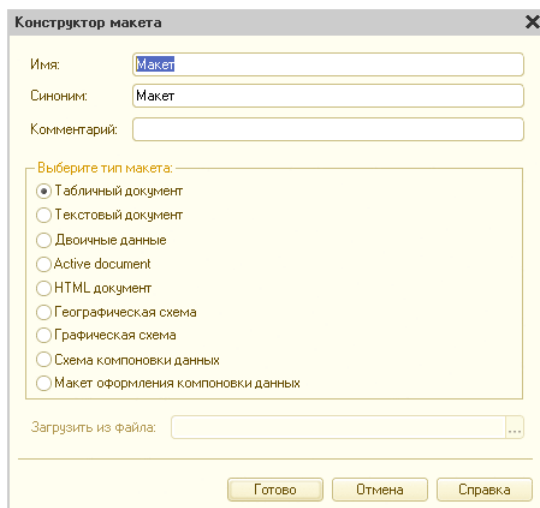


Рис. 405. Конструктор макета

Нужно указать имя, синоним и комментарий и выбрать тип макета.

Тип [Табличный документ](#) предполагает использование стандартной технологии создания и использования макетов табличных документов системы «1С:Предприятие».

Тип макета **Текстовый документ** предполагает использование специально подготовленных в качестве макетов текстовых документов. Для текстового документа свойство **Расширение** имеет значение **Текстовый макет**.

Выбор типа **Двоичные данные** подразумевает, что разработчик конфигурации знает, как работать с объектом данного типа.

Тип макета **Active document** позволяет использовать технологию **OLE Active document**. Если нажать кнопку **Готово**, то конструктор производит поиск доступных типов **Active document** и предлагает выбрать один из найденных.

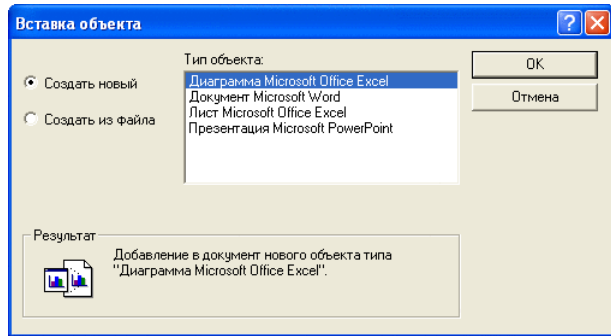


Рис. 406. Вставка Active document

Можно выбрать готовый **Active document**, служащий основой для создания макета (загрузить из файла прототип макета). Если выбран файл, чей тип не попадает в список поддерживаемых **Active document**, то конструктор выводит предупреждение.

Указанный документ будет храниться в конфигурации. Работа с макетом должна быть организована с использованием свойств и методов, предоставляемых выбранной технологической платформой.

Если выбран тип **HTML-документ**, то открывается редактор HTML-макета. В данном режиме поддерживаются все возможности HTML-редактора. В дополнение к этому редактор HTML-макета позволяет использовать картинки библиотеки картинок, из файла, из файла-коллекции или из «внутренних» картинок. Внутренние картинки – это уже выбранные из файла картинки, они хранятся «внутри» макета.

Чтобы поместить картинку в макет, нужно выбрать пункт **Элементы – Картинка...**

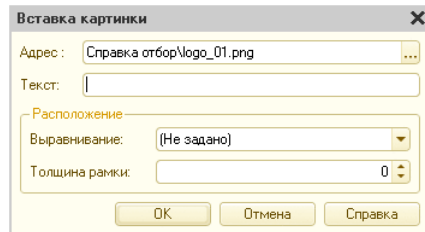


Рис. 407. Вставка картинки

Для выбора картинки нажать кнопку выбора. В открывшемся окне выбрать нужную картинку.

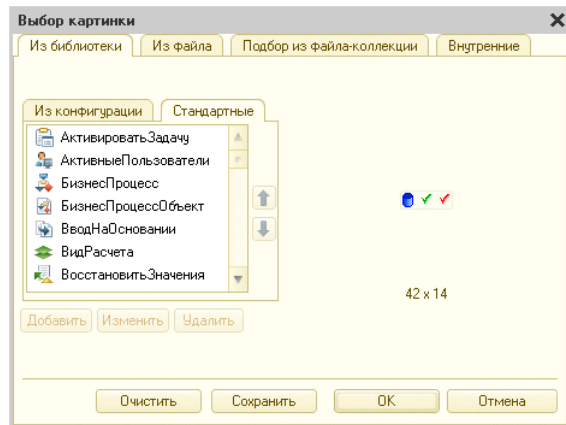


Рис. 408. Выбор картинки

Выбор типа **Географическая схема** позволяет использовать подготовленные редактором географических схем документы для создания макета.

Выбор типа **Графическая схема** позволяет использовать графические схемы, подготовленные в редакторе графических схем, или загрузить схему из файла и использовать ее в качестве макета.

Выбор типа **Схема компоновки данных** открывает конструктор схемы компоновки данных (см. [здесь](#)).

Выбор типа **Макет оформления компоновки данных** открывает окно конструктора макета оформления (см. [здесь](#)).

27.15. Конструктор форматной строки

Для написания выражений, использующих представление числа, даты и логического выражения, можно использовать конструктор форматной строки.

Чтобы вызвать конструктор, нужно установить курсор в требуемое место текста модуля и выбрать пункт **Текст – Конструктор форматной строки**. Для новой строки редактор выводит сообщение: **Форматная строка не найдена. Создать новую форматную строку?** После нажатия кнопки **Да** на экран выводится окно конструктора.

Также конструктор можно вызвать из свойств **Формат** и **Формат редактирования** элемента формы. Для этого нужно нажать кнопку выбора соответствующего свойства.

Закладки окна соответствуют типу данных, форматная строка которых формируется в окне.

Элементы управления, с помощью которых выбирается формат представления, размещены на трех закладках по типам данных:

- число,

- дата,
- логические значения.

Поле **Язык (Страна)** (параметр **л**) определяет представление информации в соответствии с национальными представлениями. Значения, заданные на следующих закладках, являются более приоритетными по отношению к указанию страны.

Для числа нужно выбрать закладку **Число**.

Рис. 409. Конструктор форматной строки для числа

В поле **Длина** (параметр **чц**) указывается общее число десятичных знаков (целой и дробной частей). В поле **Точность** (параметр **чдц**) указывается число десятичных знаков дробной части.

В поле **Сдвиг** (параметр **чс**) указывается сдвиг разрядов. Если положительный, то производится деление числа на 10 в соответствующей степени, если отрицательный – умножение.

В поле **Разделитель дробной части** (параметр **чрд**) задается символ разделителя.

В поле **Разделитель групп** (параметр **чрг**) задается символ разделителя групп. Если в качестве разделителя использовать пустую строку, то разделителем будет символ неразрывного пробела.

В поле **Группировка** (параметр **чг**) выбирается вариант группировки цифр в целой части числа.

В поле **Представление нуля** (параметр **чн**) выбирается вариант представления нулевого значения. Параметр можно использовать в форматной строке, для поля ввода он не используется.

В поле **Представление отрицательных чисел** (параметр **чо**) выбирается вариант представления отрицательных чисел.

Если флажок **Выводить лидирующие нули** (параметр **чвн**) установлен, то лидирующие нули выводятся.

СОВЕТ. В нижней части окна конструктора расположена область, в которую выводится результат действия форматной строки (группа **Пример**) и сама форматная строка.

Для настройки даты нужно выбрать закладку **Дата**.

Рис. 410. Конструктор форматной строки для даты

В поле **Формат даты** (параметр **ДФ**) выбирается вариант представления даты.

В поле **Локальный формат даты** (параметр **ДЛФ**) выбирается локальный вариант представления даты. Для поля ввода не допускается выбор значения параметра **дд**.

В поле **Представление пустой даты** (параметр **ДП**) выбирается вариант представления пустой даты. Параметр можно использовать в форматной строке, для поля ввода он не используется.

СОВЕТ. В нижней части окна конструктора расположена область, в которую выводится результат действия форматной строки (группа **Пример**) и сама форматная строка.

Для данных типа **Булево** нужно выбрать закладку **Логическое значение**.

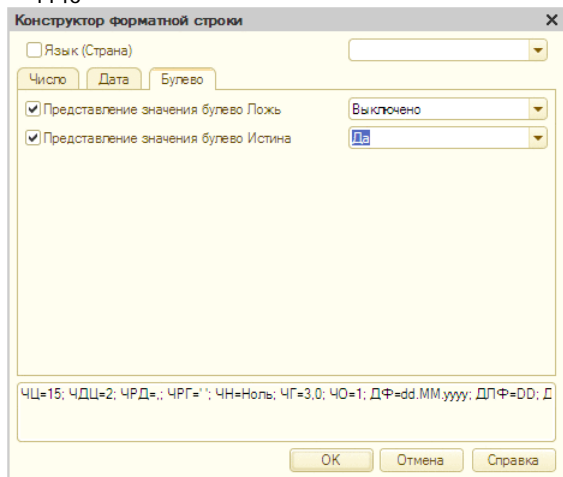


Рис. 411. Конструктор форматной строки для логического значения

Если какое-то значение не выбрано, то оно выбирается из настроек, устанавливаемых по умолчанию.

Подробнее о параметрах форматной строки написано в описании параметра [Форматная строка](#) метода [Формат\(\)](#) в описании встроенного языка.

В результате использования конструктора форматной строки в текст (или поле ввода) будет вставлена сформированная строка (при нажатии кнопки **OK**).

СОВЕТ. В редакторе текстов можно установить курсор внутри форматной строки и выбрать пункт меню [Текст – Конструктор форматной строки](#) для изменения существующей форматной строки.

27.16. Конструктор строк на разных языках

Конструктор позволяет отредактировать строки на всех языках конфигурации и дополнительно тех, чьи коды содержатся в редактируемом наборе, однако конфигурация не содержит языков с таким кодом.

Результатом работы конструктора будет строка вида:

```
<код_языка1> = <Строка_1>; <код_языка2> = <Строка_2>;...
```

[Копировать в буфер обмена](#)

Конструктор удобно использовать при создании параметра [Исходная строка](#) функции глобального контекста [НСтр\(\)](#).

Для вызова конструктора следует установить курсор в нужное место модуля и выбрать пункт [Текст – Конструктор строк на разных языках](#). Редактор анализирует текст модуля в области курсора и, когда найдена конструкция, открывает окно конструктора. Если конструкция не найдена, то конструктор сообщает об этом.

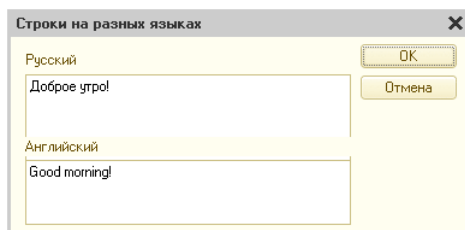


Рис. 412. Конструктор строк на разных языках

В окне конструктора будет столько полей для ввода текста, сколько языков определено для конфигурации.

Нужно ввести тексты и нажать кнопку **OK**. Результатом работы конструктора будет строка:

```
ru = 'Доброе утро!'; en = 'Good Morning!'
```

[Копировать в буфер обмена](#)

27.17. Редактор табличных документов

Для создания различных печатных форм, а также форм, предназначенных для представления и ввода информации с использованием табличных документов, в системе «1С:Предприятие» используется специализированный редактор табличных документов.

Таблицу сочетаний клавиш для редактора табличных документов можно найти в справке при использовании программы.

27.17.1. Что такое табличный документ в системе «1С:Предприятие»

Хотя в программе реализована возможность использования табличного документа для непосредственного ввода, обработки и отображения данных различных типов, как в «обычных» электронных таблицах (использование табличных документов, размещенных в форме), табличные документы в системе «1С:Предприятие» используются в основном для представления уже обработанной информации, в частности, описания печатной формы отчета. Обработка информации и помещение ее в нужные места табличного документа для большинства объектов конфигурации выполняются программными модулями на языке системы «1С:Предприятие».

В системе программ «1С:Предприятие» табличный редактор применяется для работы с отдельными табличными документами и макетами печатных форм.

Сформированный табличный документ можно сохранить в файл на диске в различных форматах. Если документ сохранен в формате MXL, то в дальнейшем этот документ можно просмотреть с помощью программы работы с файлами, скачать которую можно по этому адресу: <http://v8.1c.ru/metod/fileworkshop.htm>.

Макет хранится внутри конфигурации. Макеты бывают общими (располагаются в ветви [Общие – Макеты](#); например, макеты стандартных платежных документов, печатающихся из различных документов), а также могут относиться к определенному объекту конфигурации (например, карточка основного средства). Объект конфигурации может иметь несколько различных макетов печатных форм.

В форме также можно расположить табличный документ. Для этого необходимо использовать элемент управления [Поле табличного документа](#). В этом режиме в табличный документ можно вставить другие элементы управления.

Макеты и табличные документы с точки зрения приемов работы практически полностью совпадают.

Табличные документы (отдельные файлы и макеты) можно сравнивать и объединять.

27.17.2. Макет

Создание макета производится конструктором макетов (см. [здесь](#)) и конструктором запроса с обработкой результата (см. [здесь](#)).

27.17.2.1. Общие принципы проектирования макета

Проектирование макета заключается в «рисовании» составных частей, кирпичиков – именованных областей, из которых затем будет «собрана» готовая выходная форма – отчет. Так как практически все деловые документы имеют «прямоугольную» структуру, удобнее всего создавать макеты таких документов в редакторе, способном манипулировать прямоугольными элементами.

Именно таким редактором и является табличный редактор, входящий в систему «1С:Предприятие». В процессе создания макета можно вводить в ячейки табличного документа разнообразный текст; задавать параметры форматирования (ячейке в целом); изменять высоту строк и ширину колонок; включать в макет рисованные элементы – линии и прямоугольники, а также другие графические объекты: картинки, OLE-объекты и диаграммы, различные элементы управления; определять оформление как всего табличного документа, так и отдельных ячеек или групп ячеек.

При создании макета отчета руководствуются следующим.

Практически каждый отчет состоит из так называемой шапки (заголовка) отчета, в которой указывается наименование отчета, исходные параметры построения. Для формирования таких данных создают именованную область, которую чаще всего называют **Шапка**. Если в отчете присутствует табличная часть, то в шапку обычно включают наименование колонок.

Для вывода строк создают именованные области (в показанном на [рис. 413](#) примере это **Строка**), отвечающие за вывод различной информации в табличную часть. Число строк табличной части обычно невозможно узнать при подготовке макета, но структурно информация в табличной части повторяется, поэтому при построении отчета используют одни и те же области, описывающие отдельную строку. Ячейкам, предназначенным для вывода конкретной информации, ставят в соответствие параметры (в приведенном примере это **Счет**, **Наименование**, **Цена**, **Сумма** и др.). При выводе очередной строки этим параметрам присваивают содержимое каждой выводимой строки, а затем уже включают сформированную область в состав отчета. Так происходит до тех пор, пока не будет выведена вся информация табличной части.

Обычно отчет завершается выводом итоговых данных и реквизитами ответственных лиц. Эти данные обычно размещают в области, именуемой **Подвал**.

В окончательном виде макет представляет собой совокупность прямоугольных областей, каждая из которых служит для выдачи какой-то части готового отчета: область для выдачи заголовочной части (наименования, даты и т. п.); область для выдачи шапки табличной части и так далее.

Рис. 413. Пример макета

Сам процесс построения отчета происходит следующим образом:

- Вначале отчет строится как пустой табличный документ. Необходимо иметь в виду, что макет не связан напрямую с готовым отчетом. Скорее макет представляет собой некий конструктор, набор областей, из которых в процессе работы программного модуля складывается готовый отчет.
- В процессе обработки алгоритма отчета из макета в нужном порядке извлекаются именованные области, транслируются (вместо параметров подставляются их значения) и копируются в табличный документ готового отчета. Средства языка формирования отчетов позволяют наращивать отчет как вниз, так и вправо.

27.17.2.2. Свойства ячеек макета

Категория свойств «Макет»

Категория свойств **Макет** показывается, если в свойствах табличного документа установлено свойство **Макет**.

Заполнение – пункты свойства **Заполнение** устанавливают, какого рода информация введена в ячейку. Свойство показывается, если в категории **Значения** не установлено свойство **Содержит значение**. Пункты не изменяют внешний вид ячейки, а используются только в процессе обработки шаблона, при формировании готового табличного документа.

Пункты этого списка имеют следующий смысл:

Формат данных	Пояснение
Текст	Информация в ячейке является текстом и при формировании табличного документа будет перенесена из макета в готовый табличный документ без изменений
Параметр	Информация в ячейке представляет собой параметр, имя которого указывается в свойстве Параметр
Шаблон	Информация в ячейке представляет собой текст с включенными в него именами параметров, заключенных в квадратные скобки. При формировании табличного документа переменные будут вычислены и включены в текст. Место, отводимое в тексте для вывода значений параметров, определяется длиной этих значений. Пример шаблона: Директор: [Директор]

Информация в ячейке готового табличного документа преобразуется в тип **Строка**.

Параметр – имя параметра для вывода содержимого ячейки. Свойство показывается, если в категории **Значения** установлено свойство **Содержит значение** или когда в свойстве **Заполнение** категории **Макет** выбрано значение **Параметр**.

Параметр расшифровки – указывается имя параметра, по которому программа производит обработку расшифровки значения, находящегося в ячейке. Имеет смысл использовать, когда результирующий табличный документ размещен в форме (элемент управления **Поле табличного документа**).

Расход товара

Номер 000000015
 Дата 09.11.2007 14:12:14
 Покупатель Магазин "Продукты"
 Склад Средний

Товар	Цена	Количество	Сумма
Торт	200	25	5 000

Расшифровка

Расход товара

Номер 000000015
 Дата 09.11.2007 14:12:14
 Покупатель Магазин "Продукты"
 Склад Средний

Товар	Цена	Количество	Сумма
Торт	200	25	5 000

Гиперссылка

Рис. 414. Виды курсоров в табличном документе

Когда готовый табличный документ открыт в режиме **Только просмотр**, при помещении указателя мыши над ячейкой, содержащей заполненное свойство **Параметр** **расшифровки**, указатель мыши может принимать форму как на [рис. 414](#) (курсор вида **Гиперссылка** будет появляться тогда, когда для ячейки-ссылки установлено свойство **Гиперссылка**). Это значит, что возможна детализация (расшифровка) данных табличного документа. Теперь, если дважды щелкнуть левой кнопкой мыши на этой ячейке (или сделать ее активной и нажать клавишу **Enter**, а для гиперссылки просто щелкнуть мышью), значение поля будет выдано на экран:

- Значения типа **Строка**, **Число**, **Дата** и **Перечисление** будут выданы для просмотра.
- Если значение имеет тип **Документ**, соответствующий документ будет открыт для просмотра и редактирования.
- Если значение является элементом справочника, этот элемент будет открыт для просмотра и редактирования в диалоге. Если при настройке свойств справочника было задано редактирование в списке, будет открыта форма списка справочника, а указатель в табличном поле будет установлен на нужный элемент справочника.

Эта обработка расшифровки называется стандартной и не требует никаких дополнительных настроек.

Если требуется, чтобы обработка расшифровки производилась особым образом, то необходимо в свойствах поля табличного документа, в категории **События** для события **Обработка расшифровки** указать имя процедуры, осуществляющей обработку события, возникающего при выборе ячейки с расшифровкой. В модуле формы в теле этой процедуры нужно средствами встроенного языка описать обработку расшифровки.

Приведем пример. Для отчета создана форма и макет **МакетПечати**. Макет содержит именованные области **Заголовок**, **Шапка**, **ТоварыШапка** и **Товары**. Область **Товары** содержит ячейку, в параметрах расшифровки которой указано имя параметра расшифровки **Расшифровка**. В форме размещен элемент формы типа **Таблица**, связанный с реквизитом формы **ТабДок** типа **ТабличныйДокумент**. В категории свойств **События** этого элемента для свойства **Обработка расшифровки** указывается на процедуру обработки выбора ячейки **РезультатОбработкиРасшифровки()**, расположенной в модуле отчета.

Формирование табличного документа производится по следующей примерной схеме:

```
ТабДок.ОтображатьСетку = Ложь;
ТабДок.ОтображатьЗаголовки = Ложь;
ТабДок.Защита = Истина;
ТабДок.ТолькоПросмотр = Истина;
Макет = Документы.РасходТовара.ПолучитьМакет("МакетПечати");
// Заголовок
Область = Макет.ПолучитьОбласть("Заголовок");
ТабДок.Вывести(Область);
// Шапка
Шапка = Макет.ПолучитьОбласть("Шапка");
Шапка.Параметры.Заполнить(ЭтотОбъект);
ТабДок.Вывести(Шапка);
// Товары
Область = Макет.ПолучитьОбласть("ТоварыШапка");
ТабДок.Вывести(Область);
ОбластьТовары = Макет.ПолучитьОбласть("Товары");
Для Каждого ТекСтрокаТовары Из Товары Цикл
  ОбластьТовары.Параметры.Заполнить(ТекСтрокаТовары);
  ТабДок.Вывести(ОбластьТовары);
КонецЦикла;
```

Копировать в буфер обмена

Пример реализации процедуры обработки выбора расшифровки:

```
Процедура РезультатОбработкиРасшифровки(Эл, Расшифровка, СтандартнаяОбработка)
  СтандартнаяОбработка = Ложь;
  ТипРасшифровки = ТипЗнч(Расшифровка);
  Если ТипРасшифровки = Тип("СправочникСсылка.Номенклатура") Тогда
    ИмяФормыОтчета = "ФормаОтчета1";
  Иначе
    ИмяФормыОтчета = "ФормаОтчета2";
  КонецЕсли;
  ФормаОтчета = ОбработкаОбъект.ПолучитьФорму(ИмяФормыОтчета);
  ФормаОтчета.ВыполнитьОтчет(Расшифровка, ПериодС, ПериодПо);
  ФормаОтчета.Открыть();
КонецПроцедуры
```

Копировать в буфер обмена

ФормаОтчета1 и **ФормаОтчета2** – имена форм, разработанных специально для детализированных отчетов (обычно такие формы содержат поле табличного документа, в которое выводится результат).

Использование расшифровки – определяется область использования расшифровки. При выборе ячейки расшифровка возможна только для указанной ячейки; при выборе строки расшифровка действует в каждой ячейке текущей строки. Если выбрано **Не использовать** и указана расшифровка, расшифровка не действует: указатель мыши не изменяет форму (возможные виды курсоров (см. [рис. 414](#)) и выбор в ячейке не отрабатывается.

Категория свойств «Значения»

Содержит значение – если установлено, то ячейка содержит значение. Установка свойства влияет на состав свойств других категорий.

Тип значения – тип значения ячейки. Для табличного документа перечень содержит типы **Число**, **Строка**, **Дата** и **Булево**. Для элемента управления формы **Поле табличного документа** помимо примитивных типов перечень дополнительно содержит типы, определенные для текущей конфигурации, – документы, справочники, перечисления и т. д. Кроме того, можно установить тип данных **Произвольный**. В таком случае он может быть определен уже во время заполнения формы с помощью средств встроенного языка.

Элемент управления – выбирается элемент управления для редактирования содержимого ячейки. Список возможных значений зависит от выбранного типа содержимого. Например, при выборе в типе данных значения **Число** в качестве редактора можно выбрать **Поле ввода** или **Флажок**. Для выбранного типа редактора в список объектов

(Ячейки, Табличный документ) будет добавлена строка, содержащая тип редактора [Поле ввода](#) или [Флажок](#). Состав свойств палитры свойств данного объекта определяется типом выбранного элемента управления.

Формат – задается форматная строка, которая будет использоваться при выводе значения. Если не производить настройку формата изображения, то формат будет выбран из региональных настроек информационной базы.

Формат редактирования – задается форматная строка, которая будет использоваться для отображения значения при его редактировании. Если не производить настройку формата изображения, то формат будет выбран из региональных настроек информационной базы.

Формат можно ввести вручную или с помощью конструктора форматной строки (см. [здесь](#)).

ПРИМЕЧАНИЕ. При редактировании табличного документа в тонком клиенте недоступны следующие свойства ячейки: [Содержит значение](#), [Тип значения](#), [Элемент управления](#), [Формат редактирования](#).

27.17.3. Табличный документ в режиме ввода данных

Другой способ построения отчета заключается в том, что пользователь вводит данные в предназначенные для этого ячейки табличного документа, размещенного в элементе формы.

В процессе ввода эти данные обрабатываются процедурами, написанными на встроенном языке, располагаемыми в модуле формы. Результаты расчетов могут использоваться при вычислении других ячеек табличного документа.

После ввода данных можно распечатать отчет и сохранить его для последующей работы.

Общая схема работы с табличным документом в режиме ввода данных выглядит следующим образом:

- формируется макет типа [Табличный документ](#), подготовленный особым образом, в который будет выполняться ввод данных;
- формируется форма, в которой будет расположено поле вида [Табличный документ](#), и в нем будет организован ввод данных;
- формируются обработчики элемента формы вида [Табличный документ](#), которые будут обрабатывать данные, введенные пользователем (при необходимости).

Рассмотрим перечисленные этапы подробнее на примере формирования бланка доставки товара из документа [ДоставкаТоваров](#). Желаемый бланк приведен на [рис. 415](#). В этом бланке поля, выделенные подчеркиванием, должны вводиться пользователем. Причем поле [Дата доставки](#) будет автоматически рассчитываться как поле [Дата оформления](#), увеличенное на [Срок доставки](#). [Дата оформления](#) является датой документа.

Бланк доставки товаров

Адрес доставки: _____

Срок доставки _____
Дата доставки _____

Дата оформления _____
Подпись _____

Рис. 415. Бланк доставки

27.17.3.1. Подготовка табличного документа

Для того чтобы реализовать табличное поле в режиме ввода данных, необходимо создать макет типа [Табличный документ](#) (с именем [БланкДоставки](#)). Затем нужно открыть сформированный макет и установить свойство [Макет](#) в значение [Ложь](#).

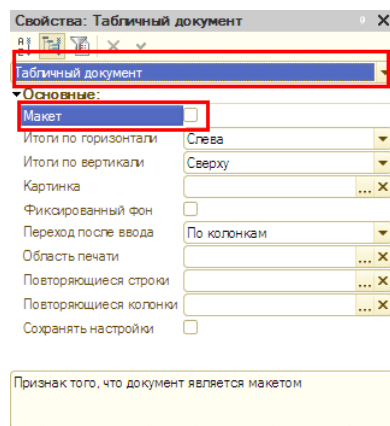


Рис. 416. Свойства табличного документа

Далее нужно сформировать необходимую структуру документа, который будет формироваться.

Сформируем необходимые поля:

- **Адрес доставки.** Нужно объединить несколько ячеек и затем определить следующие свойства:
 - **Имя** – [АдресДоставки](#);
 - **Защита** – [Ложь](#);
 - **Содержит значение** – [Истина](#);
 - **Тип значения** – [Строка](#);
 - **Элемент управления** – [Поле ввода](#).
- **Срок доставки.** Для ячейки определить следующие свойства:
 - **Имя** – [СрокДоставки](#);
 - **Защита** – [Ложь](#);
 - **Содержит значение** – [Истина](#);

- Тип значения – Число, длина 3, точность – 0, неотрицательное;
- Элемент управления – Поле ввода.
- Дата доставки. Для ячейки определить следующие свойства:
 - Имя – ДатаДоставки;
 - Защита – Ложь;
 - Содержит значение – Истина;
 - Тип значения – Дата, состав даты – Дата;
 - Элемент управления – Поле ввода.
- Дата оформления. Для ячейки определить следующие свойства:
 - Имя – ДатаОформления;
 - Защита – Истина;
 - Содержит значение – Истина;
 - Тип значения – Дата, состав даты – Дата;
 - Элемент управления – Поле ввода;
 - Форма – ДЛФ=DD.

В результате должен получиться табличный документ, представленный на [рис. 417](#).

	1	2	3	4	5	6	7	8
1								
2	Бланк доставки товаров							
3								
4	Адрес доставки:							
5	АдресДоставки							
6								
7	Срок доставки:	СрокДоставки						
8	Дата доставки:	ДатаДоставки						
9								
10	Дата оформления:	ДатаОформлени						
11	Подпись:							
12								

Рис. 417. Готовый макет

Макет представлен с включенным режимом [Отображать именованные ячейки](#) (можно включить с помощью команды меню [Таблица – Имена – Отображать именованные ячейки](#)).

ПРИМЕЧАНИЕ. Свойства, которые могут быть заданы для поля ввода в макете, не сохраняются и не используются системой при формировании полей ввода в табличном документе в режиме ввода данных.

27.17.3.2. Подготовка формы к вводу данных

После того как подготовлен макет табличного документа, следует сформировать форму, в которой будет выполняться ввод данных, и реализовать ее вызов из документа.

В документе нужно создать команду формы [ОформитьДоставку](#) со следующим программным текстом:

```
&НаКлиенте
Процедура ОформитьДоставку(Команда)
    ПараметрыДоставки = Новый Структура("ДатаДокумента", Объект.Дата);
    ОткрытьФормуМодально("Документ.ДоставкаТоваров.Форма.ОформлениеДоставки", ПараметрыДоставки);
КонецПроцедуры
```

[Копировать в буфер обмена](#)

Команду следует разместить на форме.

Теперь необходимо создать форму, в которой будет происходить ввод данных.

Для этого следует в документе [ДоставкаТоваров](#) создать произвольную форму [ОформлениеДоставки](#). В ней нужно создать следующие элементы:

- параметр формы [ДатаДокумента](#) типа [Дата](#);
- реквизит формы [ДатаДокумента](#) типа [Дата](#);
- реквизит формы [ТабличныйДокумент](#) типа [ТабличныйДокумент](#);
- элемент формы [ТабличныйДокумент](#), связанный с реквизитом формы [ТабличныйДокумент](#).

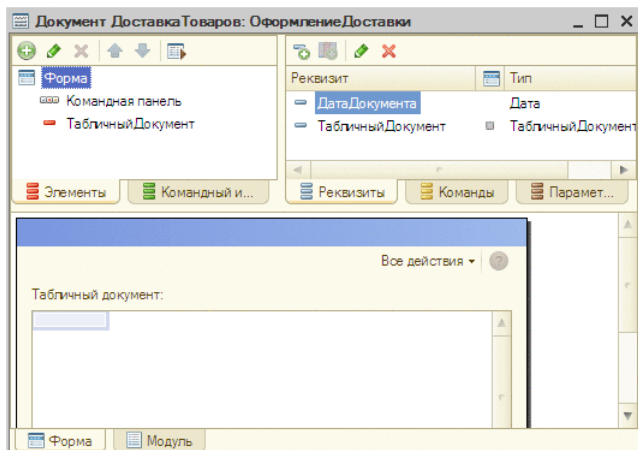


Рис. 418. Форма «ОформлениеДоставки»

Затем необходимо реализовать в форме следующие обработчики:

```

&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
// Запомним дату формирования документа
ДатаДокумента = Параметры.ДатаДокумента;
// Установим подготовленный макет
ТабличныйДокумент = Документы.ДоставкаТоваров.ПолучитьМакет("БланкДоставки");
// Установим дату создания документа в поле табличного документа
ТабличныйДокумент.Область("ДатаОформления").Значение = ДатаДокумента;
КонецПроцедуры
&НаКлиенте
Процедура ПриОткрытии(Отказ)
// Активируем область табличного документа
Элементы.ТабличныйДокумент.ТекущаяОбласть = ТабличныйДокумент.Область("АдресДоставки");
КонецПроцедуры

```

В результате получается следующая форма, заполненная данными:

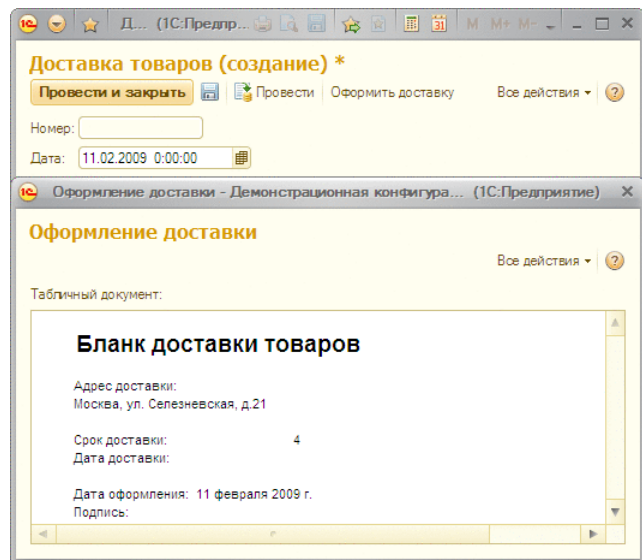


Рис. 419. Бланк доставки

27.17.3.3. Реализация реакции на изменение ячеек табличного документа

В получившейся форме отсутствует автоматический пересчет полей **Срок доставки** и **Дата доставки** при их изменении.

Для того чтобы такой пересчет стал возможным, следует реализовать обработку соответствующего события для реквизита формы, связанного с табличным документом.

Нужно добавить обработчик **ПриИзмененииСодержимогоОбласти** для элемента формы **ТабличныйДокумент**. В обработчике должен располагаться следующий программный текст:

```

&НаКлиенте
Процедура ТабличныйДокументПриИзмененииСодержимогоОбласти(Элемент, Область)
СекундВСутках = 24 * 60 * 60;
Если Область.Имя = "ДатаДоставки" Тогда
ТабличныйДокумент.Область("СрокДоставки").Значение = (ТабличныйДокумент.Область("ДатаДоставки").Значение - НачалоДня(ДатаДокумента)) / СекундВСутках;
ИначеЕсли Область.Имя = "СрокДоставки" Тогда
ТабличныйДокумент.Область("ДатаДоставки").Значение = НачалоДня(ДатаДокумента) + ТабличныйДокумент.Область("СрокДоставки").Значение * СекундВСутках;
КонецЕсли
КонецПроцедуры

```

Теперь табличный документ обрел желаемую интерактивность.

27.18. Редактор карты маршрута

Карта маршрута – это схематическое изображение последовательности выполнения действий, предусмотренных бизнес-процессом. Она является частным случаем графической схемы.

Основное отличие карты маршрута от графической схемы – особая обработка специальных элементов схемы (точка старта, завершения, действия, условия и т. д.).

Карта маршрута является одновременно и инструкцией системе по выполнению последовательности действий бизнес-процесса, и иллюстрацией для пользователя структуры этих действий, а также средством отображения текущего состояния бизнес-процесса (описание метода **ПолучитьКартуМаршрута()** справки по встроенному языку).

Карта маршрута представляет собой прямоугольную область экрана, которая в самом общем случае содержит различные элементы карты маршрута, например, точку старта, точку завершения, декорации, соединительные линии и т. д.

27.18.1. Редактирование карты маршрута

Для редактирования карты маршрута в окне редактирования бизнес-процесса, на закладке **Прочее** нужно нажать кнопку **Карта маршрута**.

Процесс редактирования карты маршрута заключается в размещении на карте маршрута элементов карты различных типов, редактировании их свойств и соединении их друг с другом соединительными линиями.

Для редактирования карты маршрута используется мышь или клавиатура (некоторые операции выполняются только с помощью мыши – например, соединение элементов карты маршрута с помощью соединительных линий). При появлении редактируемой карты маршрута на экране становятся доступными кнопки панели инструментов **Вставка элементов карты маршрута** и пункт меню верхнего уровня **Карта маршрута**.

Установка свойств карты маршрута производится с помощью палитры свойств (см. [здесь](#)).

27.18.2. Элементы карты маршрута

27.18.2.1. Рекомендации по оформлению

Наилучшим подходом для рисования карт маршрута является вертикальная ориентация карты. Карта маршрута предполагает вертикальное расположение (сверху вниз). Например, при добавлении элемента карты маршрута (далее в данном разделе «элемента») **Точка действия** он сразу создается с исходящей соединительной линией, направленной вниз.

При создании надписей желательно использовать одинаковые шрифты. Использование другого шрифта является дополнительным средством привлечения внимания (например, заголовков декорации, выделяющей группу элементов).

Редактор карты маршрута позволяет размещать в карте маршрута различные элементы, задавать их размеры и выравнивать границы и т. д.

Чтобы облегчить размещение элементов, редактор карты маршрута предоставляет различные сервисные средства. Это использование разметочной сетки, а также выполнение различных действий над группой элементов (выравнивание, распределение в карте маршрута, установка размеров и т. д.).

В качестве дополнительных возможностей можно использовать индивидуальное оформление элементов. С помощью свойств категории [Оформление](#) производится настройка цвета текста и фона поля, шрифта текста, выбирается вид и цвет рамки, использование картинки, подсказки и другие приемы оформления. Состав свойств зависит от типа элемента.

27.18.2.2. Порядок элементов

В отличие от обычной графической схемы, в карте маршрута даже после изменения порядка определенные типы элементов все равно сохраняют порядок, характерный для данного типа. А именно:

- Декорации всегда находятся на заднем плане (внизу) (т. е. менять порядок декораций можно только относительно друг друга – относительно других элементов они всегда будут ниже).
- Следом за декорациями (выше их) идут соединительные линии.
- На самом веру – все остальные элементы (визуализирующие точки бизнес-процесса). Таким образом, элементы, составляющие логику карты маршрута, всегда лежат выше декораций и декоративных соединительных линий по порядку отрисовки.

27.18.2.3. Работа с соединительными линиями

Смысл карты маршрута состоит в описании последовательности действий бизнес-процесса – она задается именно связью элементов с помощью элемента [Соединительная линия](#). Соединительные линии нельзя вставлять просто так – они всегда присоединены к каким-либо точкам бизнес-процесса и не могут существовать сами по себе. Соединительные линии нельзя удалить, если это не предусмотрено точкой бизнес-процесса (например, точка разделения и точка выбора варианта).

Связывать друг с другом можно все элементы, визуализирующие точки бизнес-процесса, т. е. все элементы, кроме элементов [Декорация](#), [Соединительная линия](#) и [Декоративная линия](#). По умолчанию большинство элементов, визуализирующие точки бизнес-процесса, вставляются в карту маршрута с одной исходящей линией, которую нельзя ни отсоединить, ни удалить. Ее можно только передвинуть на другой порт (**порт** – область элемента, куда может быть присоединена линия – обычно это середина стороны прямоугольника, занимаемого элементом; на схеме незанятый порт обозначается синим крестиком).

Для работы с элементом карты маршрута [Соединительная линия](#) нужно выбрать ее нажатием мыши на любом сегменте (отрезке) соединительной линии либо перейти к ней с помощью клавиши **Tab** (**Shift + Tab**).

Если конец линии не присоединен ни к какому элементу, то прямоугольник на конце линии имеет красный цвет, а стрелка имеет контур, т. е. не закрашена внутри. Для присоединения нужно мышью захватить конец соединительной линии (серый или красный прямоугольник) и, потащив в область порта какого-либо элемента, там отпустить. После этого соединительная линия автоматически перестроится. Можно также переприсоединить соединительную линию (присоединив ее конец к другому элементу). При этом действует следующее ограничение: нельзя напрямую зациклить один элемент на втором (**точка1 " точка2 " точка2 " точка1**), если ни один из этих элементов не является элементом карты маршрута [Точка условия](#) или [Точка выбора варианта](#).

При вставке элемента в карту маршрута автоматически происходит попытка присоединить незанятые порты этого элемента к близлежащим неприсоединенным линиям. При перемещении или изменении размера элемента автоматически происходит попытка присоединить незанятые порты этого элемента к близлежащим свободным (неприсоединенным) линиям, а также исходящие из этого элемента линии, чей конец не присоединен к незанятым портам других элементов, находящимся в непосредственной близости.

Элемент [Декоративная линия](#) предназначен для соединения декораций и точек карты маршрута. В карте маршрута может быть размещено произвольное число декоративных линий.

27.18.2.4. Проверка корректности карты маршрута

При выборе пункта [Карта маршрута – Проверить](#), а также автоматически при сохранении карты маршрута происходит проверка корректности карты маршрута. При этом проверяются следующие некорректные ситуации:

- закливание;
- наличие неприсоединенных линий;
- наличие точек маршрута, не являющихся точкой старта и при этом не имеющих ни одной входящей линии;
- карта маршрута, не имеющая ни одной точки старта;
- наличие точек маршрута, не имеющих пути в точку завершения;
- карта маршрута, в которой не все линии, вошедшие в точку слияния, вышли из соответствующей ему точки разделения;
- карта маршрута, в которой распаралеленные ветки (исходящие из точки разделения) входят в одни и те же точки маршрута (до точки слияния);
- карта маршрута с циклами, которые не содержат ни одной точки вида [Точка действия](#), [Точка вложенного бизнес-процесса](#) или [Точка обработки](#) (цикл – замкнутая уникальная последовательность точек маршрута);
- карта маршрута, в которой какая-либо точка маршрута вида [Точка условия](#) или [Точка выбора варианта](#) не имеет обработчика события (соответственно [ПроверкаУсловия\(\)](#) либо [ВыборВарианта\(\)](#)).

27.18.3. Элементы карты маршрута

В разделе описываются все типы элементов карты маршрута в алфавитном порядке представления элементов.

Поведение элементов настраивается в палитре свойств путем установки и выбора значений. Некоторые свойства элементов присущи всем или большинству типов элементов. Индивидуальные свойства приводятся в справке по встроенному языку для каждого вида элементов.

27.18.3.1. Общие свойства элементов карты маршрута

Существует ряд свойств, которые присущи всем или большинству типов элементов. Обычно такие свойства имеют одинаковое назначение и редактируются одинаковым образом. Ниже будут описаны такие свойства, а в разделах, посвященных отдельным типам элементов, будет изложен порядок редактирования уникальных свойств элементов.

Приводятся только свойства, отличающие элементы карты от соответствующих элементов графической схемы.

Для удобства описания свойства будут сгруппированы по категориям палитры свойств.

Некоторые свойства, описываемые ниже, для некоторых элементов могут отсутствовать.

27.18.3.1.1. Категория свойств «Основные»

Наименование задачи – наименование задачи, которая будет формироваться в точках вида [Точка действия](#) или [Точка вложенного бизнес-процесса](#).

27.18.3.1.2. Категория свойств «События»

Свойства этой категории определяют поведение точек бизнес-процесса, представленных элементами, при определенных действиях, например, при интерактивной активации,

при создании задач, при проверке условия (для элементов вида [Точка условия](#)) и т. д.

При работе с элементами большинство действий может инициировать запуск связанных с этими действиями процедур. В теле каждой процедуры средствами встроенного языка описывается обработка события.

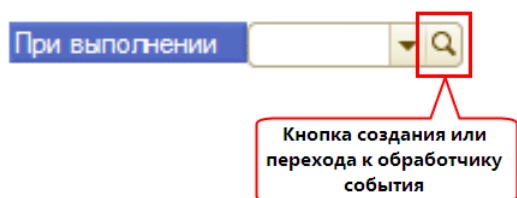


Рис. 420. Создание обработчика события

Инициализация создания процедуры, отвечающей за отработку события, производится нажатием специальной кнопки палитры свойств, расположенной справа от реквизита с наименованием события в категории свойств [События](#) (см. [рис. 420](#)).

Процедуры создаются в модуле бизнес-процесса, свойством которого является редактируемая карта маршрута.

В режиме работы с бизнес-процессом при наступлении события управление выполнением программы передается процедуре, связанной с этим событием.

Список событий карты маршрута или элементов, а также условия их наступления приведены в справке по встроенному языку, в описании соответствующего объекта.

27.18.3.2. Соединительная линия

Элемент карты маршрута [Соединительная линия](#) предназначен для соединения в карте маршрута элементов, визуализирующих точки бизнес-процесса (например, элемент [Точка старта](#), элемент [Точка действия](#), элемент [Точка условия](#) и т. д.). Элемент [Соединительная линия](#) автоматически перестраивается при изменении положения элементов. Возможности вмешиваться в алгоритм построения нет.

Элемент [Соединительная линия](#) вставляется автоматически при вставке других элементов карты маршрута и не может существовать сам по себе. Вставка дополнительных соединительных линий предусмотрена только в элементах карты маршрута вида [Точка разделения](#) и [Точка выбора варианта](#).

При построении линии система руководствуется правилом выбора кратчайшего пути, состоящего из вертикальных и горизонтальных отрезков линий и не пересекающего другие точки карты.

27.18.3.3. Точка действия

Элемент карты маршрута вида [Точка действия](#) отображает основную точку бизнес-процесса, по которой выдаются и выполняются задачи.

27.18.3.3.1. Категория свойств «Адресация»

Пояснение – строка, дополнительно характеризующая адресацию точки действия. Применяется, когда атрибуты адресации проставляются из встроенного языка, а не задаются заранее на этапе проектирования карты маршрута.

Групповая – если установлено в значение *Истина*, то задачи на данной точке бизнес-процесса будут выданы каждому члену группы (отдела). В противном случае выдается одна задача на всю группу, например, «Отдел продаж», выполняет ее один человек (первый взявший ее).

Реквизиты адресации – этих свойств столько, сколько реквизитов адресации у задачи, указанной в свойстве [Задача](#) данного бизнес-процесса. В палитре свойств можно выбрать значение из predeterminedных данных, тип которых задан в реквизите адресации задачи (например, из справочника [Отделы](#) или [Исполнители](#)).

27.18.3.4. Точка разделения

Элемент карты маршрута [Точка разделения](#) отображает точку бизнес-процесса, в которой поток исполнения разделяется на несколько параллельных веток, идущих одновременно. По умолчанию размещается в карте маршрута с тремя исходящими соединительными линиями.

Для добавления выходящей линии в контекстном меню нужно выбрать пункт [Добавить линию](#). Для удаления – выделить линию, выбрать пункт [Удалить](#). При этом нельзя удалить единственную оставшуюся исходящую линию.

27.18.3.5. Точка условия

Элемент карты маршрута [Точка условия](#) отображает точку бизнес-процесса, из которой есть два выхода, отражающие результат выполнения логического условия.

По умолчанию элемент [Точка условия](#) вставляется в карту маршрута с двумя исходящими слева и справа соединительными линиями. Справа располагается ветка, по которой идет процесс при возврате значения *Истина* в обработчике [ПроверкаУсловия](#) (). Ветки условия можно поменять местами. Для этого нужно выбрать линию, исходящую из элемента [Точка условия](#) и, захватив мышью прямоугольник у начала линии, перенести его на противоположную сторону элемента [Точка условия](#).

27.18.3.6. Точка завершения

Элемент карты маршрута [Точка завершения](#) отображает точку бизнес-процесса, в которой завершается бизнес-процесс. В карте маршрута может быть несколько элементов этого вида.

27.18.3.7. Точка старта

Элемент [Точка старта](#) отображает точку бизнес-процесса, с которой начинается выполнение бизнес-процесса. Для бизнес-процесса, имеющего несколько точек вида [Точка старта](#), при запуске должна быть указана нужная точка старта. В элемент [Точка старта](#) не могут входить соединительные линии.

27.18.3.8. Точка слияния

Элемент карты маршрута [Точка слияния](#) отображает точку бизнес-процесса, в которую сходятся параллельные пути исполнения, начавшиеся в точке разделения. Пока исполнение задач по всем параллельным путям не придет в точку слияния, переход к следующей за слиянием точке не будет выполнен. Одной точке слияния всегда соответствует одна точка разделения (но не наоборот, т. к. могут быть точки разделения без точек слияния). Не требуется явно указывать, какому элементу [Точка разделения](#) соответствует элемент [Точка слияния](#). Это будет определено автоматически.

27.18.3.9. Точка вложенного бизнес-процесса

Элемент карты маршрута [Точка вложенного бизнес-процесса](#) отображает точку бизнес-процесса, в которой запускается на исполнение вложенный бизнес-процесс. Исполнение основного (родительского) процесса возобновляется только после завершения вложенного процесса.

27.18.3.9.1. Категория свойств «Данные»

Бизнес-процесс – ссылка на вложенный бизнес-процесс.

27.18.3.10. Точка обработки

Элемент карты маршрута [Точка обработки](#) отображает точку бизнес-процесса, выполняемую в автоматическом режиме и не имеющую адресата.

27.18.3.11. Точка выбора варианта

Элемент карты маршрута [Точка выбора варианта](#) отображает точку бизнес-процесса, имеющую несколько выходов (вариантов), из которых, в зависимости от значения возвращаемого параметра [Вариант](#) обработчика [ОбработкаВыбораВарианта](#) (), выбирается только один.

Карта маршрута не имеет собственного модуля. Обработчики событий помещаются в модуль объекта бизнес-процесса.

27.19. Редактор картинок

Конфигуратор предоставляет средство редактирования картинок и коллекций картинок.

Картинки хранятся в конфигурации в ветви [Общие – Общие картинки](#) или в файлах на диске.

Для создания новой картинки в конфигурации в ветви [Общие – Общие картинки](#) нужно выполнить команду [Действия – Добавить](#), а далее открыть созданную пустую картинку для редактирования.

Для редактирования картинки в конфигурации в ветви [Общие – Общие картинки](#) следует выбрать картинку и выполнить команду меню [Действия – Изменить](#), далее в открывшемся диалоге нажать кнопку [Редактировать](#). Если картинка хранится в формате, отличном от формата [PNG](#), то при открытии будет предложено конвертировать картинку в формат [PNG](#). При отказе от конвертации редактировать картинку невозможно.

Картинки формата [WMF](#) или [EMF](#) редактировать невозможно.

Чтобы создать новую картинку, расположенную в файле на диске, нужно выполнить команду [Файл – Новый](#) и в окне выбора типа редактора выбрать [Картинка](#). Далее отредактировать пустую картинку и сохранить ее командой [Файл – Сохранить](#) или [Файл – Сохранить как...](#)

Для редактирования картинки, расположенной в файле на диске, следует выполнить команду меню [Файл – Открыть](#) и в списке файлов выбрать нужный файл.

Редактор работает в двух режимах: редактирование картинки и редактирование коллекции картинок. Различие между картинкой и коллекцией картинок носит условный характер. Под коллекцией картинок понимается картинка, состоящая из картинок-элементов, имеющих одинаковый размер. Любую картинку можно представить как коллекцию картинок. Коллекция картинок позволяет хранить и редактировать картинки, чье использование однотипно, например, пиктограммы, картинки кнопок, иконки и т. д. По умолчанию редактор открывается в режиме редактирования картинок.

Приемы редактирования картинки не зависят от выбора вида картинки (см. [здесь](#)). Особенности работы с коллекцией картинок см. [здесь](#).

27.19.1. Редактирование картинки

После выбора (или создания) картинки ее можно редактировать с помощью редактора картинок.

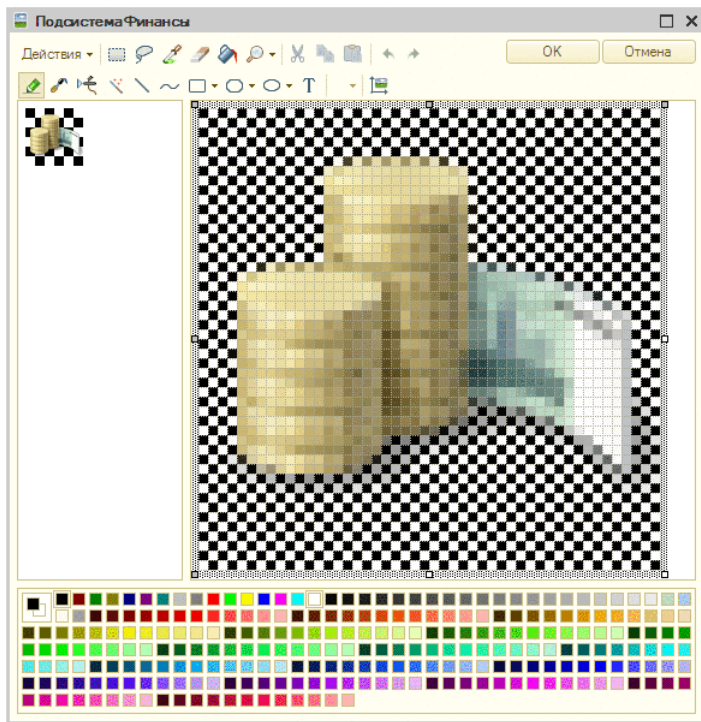


Рис. 421. Редактор картинок

Окно редактора состоит из панели инструментов, двух полей для редактирования картинки (левое представляет картинку в натуральную величину и предназначено для просмотра результата редактирования, а правое поле содержит увеличенное изображение картинки и предназначено для редактирования) и палитры цветов для выбора цвета.

Редактировать картинку можно в любом поле. Редактирование осуществляется мышью. Предварительно нужно выбрать инструмент или фигуру для рисования (карандаш, кисть, аэрограф, линию, прямоугольник, эллипс), в палитре цветов следует выбрать цвет для инструмента.

Текущее положение указателя мыши и размер области, а также масштаб показываются в панели состояния.

Картинка представляет собой прямоугольную область, состоящую из набора точек (пикселей), размер которой можно изменить с помощью мыши, потянув маркер нижней или правой границы или правого нижнего угла. Размеры картинки также можно изменить в окне [Параметры картинки](#) (см. ниже).

Редактирование сводится к указанию определенного цвета каждой точки картинки. Количество цветов определяется разрешением картинки. Чем больше разрешение, тем больше цветов можно использовать.

ВНИМАНИЕ! Использование разрешения 24 бита на пиксель при больших размерах картинок приводит к увеличению размеров конфигурации.

Выбор цвета производится в палитре цветов отдельно для каждой кнопки мыши. Выбранный цвет показывается двойной рамкой.

Состав цветов можно менять. Для этого нужно дважды щелкнуть мышью цвет, который требуется изменить. В открывшемся окне выбора цвета выбрать нужный из стилей или создать новый цвет.

Размер картинки можно менять. Для этого достаточно с помощью указателя мыши потянуть маркер стороны или угла области картинки.

Сетка (пиксельная) предназначена для облегчения редактирования картинки. Сетка показывается прерывистой линией. Для настройки показа сетки нужно выбрать пункт [Действия – Сетка](#). На экран выводится диалог.

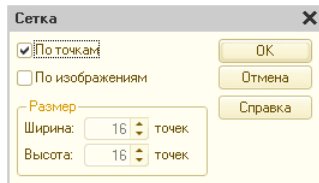


Рис. 422. Установка параметров сетки

Если флажок **По точкам** установлен, то показывается пиксельная сетка.

Если установлен флажок **По изображениям**, то подразумевается, что картинка представляет собой коллекцию картинок одинакового размера (о редактировании коллекции (см. раздел ниже). В этом случае становятся доступными поля для указания размеров элемента коллекции. Помимо пиксельной сетки в поле редактирования картинки выводится сетка коллекции в виде тонких сплошных линий. При этом режим редактирования не изменяется.

При установленном флажке **По изображениям** изменение размера картинки кратно размеру ячейки. Если флажок не установлен, то размер изменяется с точностью до пикселя.

Для рисования используют различные инструменты и набор фигур. Набор и порядок использования инструментов аналогичны стандартному набору инструментов, используемых в программе **Paint**, входящей в **Microsoft Windows**. Таблицу сочетаний клавиш для редактора картинок (см. в справке при использовании программы).

Но есть и некоторые отличия. По кнопке **Масштаб** изображение поля редактирования может масштабироваться в пределах от 1:1 до 20:1 с шестью степенями выбора. Нажатие кнопки приводит к выбору следующей степени масштаба. Когда достигнут масштаб 20:1, очередное нажатие кнопки приводит к выбору масштаба 1:1. Нужный масштаб можно выбрать сразу, нажав справа от кнопки масштаба кнопку выбора (с маленьким треугольником вершиной вниз).

Если нажать кнопку **Масштаб** и перевести указатель мыши на любую из областей рисунка, то появляется рамка. Эта рамка обозначает область изображения, которая будет показана при нажатии на клавиши мыши.

Изменить масштаб также можно, используя мышь с колесом прокрутки с нажатой клавишей **Ctrl**.

Кнопки рисования фигур (прямоугольник, прямоугольник с закругленными краями, эллипс) также имеют кнопки выбора, нажав которые можно выбрать вид фигуры (простой, обведенный закрашенный, закрашенный).

Кнопка **Параметры картинки** вызывает на экран диалог:

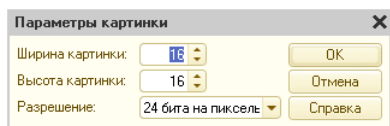


Рис. 423. Параметры картинки

В диалоге задается размер картинки и разрешение (определяет максимальное число используемых цветов). Для картинок, у которых разрешение установлено в 1, 4 или 8 бит на пиксель, невозможно использовать альфа-канал. Разрешается использование только одного прозрачного цвета. Использовать альфа-канал можно в случае, если разрешение больше 8 бит на пиксель.

ПРИМЕЧАНИЕ. Не следует использовать в конфигурациях альфа-канал у картинок, чей размер превышает 40 000 точек (например, картинка 200x200 точек). Для таких картинок не поддерживается корректное отображение в веб-клиенте, который работает в веб-браузере **Microsoft Internet Explorer 6.0**. Это правило не относится к картинкам-коллекциям, размер элементов которых меньше указанного ограничения.

Редактор картинок допускает использование стандартных команд работы с буфером обмена. Для вставки рисунка нужно использовать **Ctrl + V**, для копирования – **Ctrl + C**, для копирования с удалением – **Ctrl + X**. Если размер картинки превышает текущий размер, то редактор предлагает изменить его.

В палитре инструментов есть многоцелевая кнопка, которая меняет свое назначение в зависимости от выбранного инструмента или фигуры. Она расположена во втором ряду справа, перед кнопкой **Параметры картинки**. При выборе линии, кривой, прямоугольника и эллипса данная кнопка показывает толщину используемой линии, с помощью которой рисуется фигура, в пикселях. Нажатие кнопки приводит к последовательной смене толщины (пять ступеней). При нажатии кнопки выбора открывается выпадающее меню, в котором можно выбрать нужную толщину линии рисования фигуры. Для аэрографа с помощью данной кнопки определяется размер пятна, для кисти – размер и форма, для ластика – размер. При выборе режима выделения или ввода текста кнопка предоставляет возможность задать режим прозрачности.

Для ввода текста нужно нажать кнопку **Текст**. На экран выводится диалог:

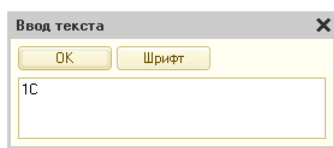


Рис. 424. Диалог ввода текста

В многострочном поле ввода вводится текст. По кнопке **Шрифт** выбирается требуемый шрифт (начертание, размер и другие характеристики). Выбор шрифта осуществляется из системных шрифтов или из стилей, определенных в конфигурации.

С помощью кнопки **Замена цветов** просто осуществляется замена выбранного указателем мыши цвета (указывается пиксель) на цвет, установленный для данной кнопки мыши. Таким образом можно быстро перекрашивать рисунки, используя сразу два цвета на каждую кнопку мыши.

27.19.2. Коллекции картинок

Коллекция картинок – это картинка, состоящая из отдельных элементов (картинок) одинакового размера. Элементы образуют прямоугольную матрицу, каждая ячейка которой представляет самостоятельную картинку.

Коллекция картинок предназначена для упрощения выбора нужной картинки в элементы управления, заголовки колонок и т. д. Использование коллекции гарантирует выбор картинок одинакового размера для однотипного использования.

С помощью редактора картинок можно создавать и редактировать коллекции картинок. Для перехода в режим редактирования коллекции нужно выбрать пункт **Режим коллекции**. При этом картинка, показанная в поле просмотра, разбивается на ячейки. Для редактирования картинка-элемента следует дважды щелкнуть ячейку мышью – картинка-элемент показывается в поле редактирования. Приемы редактирования описаны в разделе выше.

В панели инструментов добавляются две кнопки – **Добавить колонку** и **Добавить строку**. При нажатии той или иной кнопки в картинку добавляется новая колонка или строка.

Изменить размер элемента можно только в диалоге **Параметры картинки** (маркеры изменения размера недоступны).

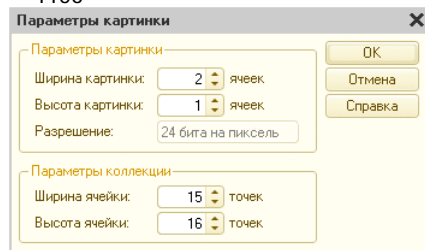


Рис. 425. Параметры коллекции

В группе элементов **Параметры картинки** задаются размеры картинки в элементах (ячейках). В группе элементов **Параметры коллекции** задаются размеры ячейки в пикселях (точках).

Коллекцию картинок можно также редактировать как обычную картинку. При этом рекомендуется предварительно настроить показ сетки по изображениям (см. [здесь](#)).

27.20. Редактор HTML-документа

Данный редактор предоставляет основные возможности по редактированию HTML-документов. Также с помощью этого редактора редактируется справочная информация для объектов конфигурации.

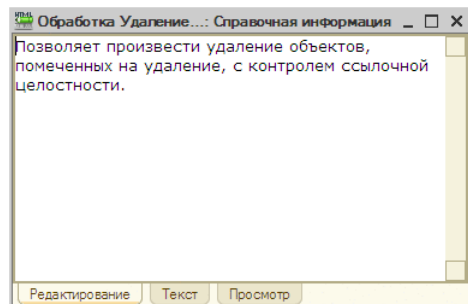


Рис. 426. Редактор HTML-документа

Редактор состоит из трех закладок:

- **Редактирование** – с помощью этой закладки имеется возможность редактировать документ в визуальном режиме.
- **Текст** – с помощью этой закладки имеется возможность редактировать документ непосредственно с помощью языка разметки HTML. При этом возможности редактора совпадают с возможностями редактора текстов (см. [здесь](#)).
- **Просмотр** – на этой закладке можно посмотреть, как будет выглядеть созданный документ, однако недоступно непосредственное изменение документа.

27.20.1. Визуальное редактирование

27.20.1.1. Форматирование текста

Текст HTML-документа можно форматировать, т. е. устанавливать его стиль (шрифт, размер, оформление, отступы, выравнивание и т. д.).

При копировании текста из других текстовых редакторов, например, Microsoft Word, сохраняется исходное форматирование текста.

27.20.1.2. Вставка и работа с таблицей

В HTML-документ можно вставлять таблицы. Для вставки таблицы нужно выбрать пункт **Таблица – Вставить таблицу**. В открывшемся диалоге указать количество строк и колонок.

Если в дальнейшем в процессе редактирования выяснится, что указанного количества строк или колонок не хватает, их можно добавить.

Для работы с таблицей (вставка, удаление строк, колонок и ячеек) предназначены команды, выполняемые при выборе пунктов меню **Таблица** главного меню конфигуратора.

Пункт меню «Таблица»	Действие
Вставить строку	Сверху от текущей строки добавляется новая строка
Удалить строку	Удаляется текущая строка
Вставить колонку	Справа от текущей колонки добавляется новая колонка
Удалить колонку	Удаляется текущая колонка
Вставить ячейку	В таблицу слева от текущей ячейки добавляется ячейка. При этом все ячейки справа сдвигаются
Удалить ячейку	Удаляет текущую ячейку. Все ячейки справа сдвигаются влево
Объединить ячейки	Объединяет две ячейки: текущую и ячейку, расположенную справа от текущей. При этом введенная информация в этих ячейках также объединяется. Если требуется объединить больше чем две ячейки, нужно выполнить эту операцию несколько раз
Разбить ячейку	Разделяет текущую ячейку на две ячейки

Таблицы могут быть вложенными. Для создания вложенной таблицы необходимо создать новую таблицу в ячейке уже существующей таблицы.

27.20.1.3. Вставка картинки

В HTML-документ можно вставлять картинки.

Для вставки картинки нужно выбрать пункт **Элементы – Картинка**. В открывшемся диалоге указать адрес картинки (путь и имя файла картинки) или выбрать файл в стандартном диалоге выбора файла. В поле **Текст** ввести поясняющий текст, который будет появляться в режиме просмотра в виде всплывающей подсказки при помещении указателя мыши над картинкой.

В группе реквизитов **Расположение** необходимо указать, если требуется, выравнивание картинки (как располагается картинка в документе).

Значение выравнивания	Действия
Не задано	Картинка располагается в тексте

Лево	Картинка располагается в новой строке, прижата к левой границе документа
Право	Картинка располагается в новой строке, прижата к правой границе документа
Поверх текста	Картинка располагается в тексте с выравниванием по центру картинки
Низ	Картинка располагается в тексте с выравниванием текста по нижней границе картинки
Центр	Картинка располагается в тексте с выравниванием текста по центру картинки
Верх	Картинка располагается в тексте с выравниванием текста по верхней границе картинки

Если требуется поместить картинку в рамку, в реквизите диалога [Рамка](#) нужно указать толщину рамки. Если толщина не указана или равна 0, то рамка не рисуется. Для вставки картинки нажать кнопку **ОК**. Указанные в диалоге реквизиты можно изменить в палитре свойств картинки.

Вставленную картинку можно переместить. Для этого следует указать картинку и перетащить ее мышью в нужное место документа. При этом свойство картинки [Положение](#) не меняется, т. е. если установлено не [Лево](#) и не [Право](#), то картинку можно расположить в тексте, в противном случае картинка будет располагаться в новой строке.

Картинку можно копировать. Копирование осуществляется стандартным образом (с помощью буфера обмена или с помощью мыши при нажатой клавише **Ctrl**).

Картинку можно расположить в любой части HTML-документа (текст, таблица и надпись).

27.20.1.4. Надпись

[Надпись](#) – специальный объект HTML-документа. Он представляет собой область, в которую можно поместить текст, таблицу, картинку или другую надпись. При просмотре надписи вложенные объекты непрерывно прокручиваются справа налево.

Для вставки линии нужно выбрать пункт [Элементы – Надпись](#).

Размер надписи можно изменить. Для этого следует выделить надпись и с помощью мыши переместить маркер надписи.

Надписи можно располагать в тексте и в таблице.

27.20.1.5. Гиперссылка

Для создания гиперссылки в HTML-документе можно создавать ссылки.

Нужно выделить объект (текст или картинку), выбрать пункт [Элементы – Ссылка](#). На экран выводится диалог формирования ссылки.

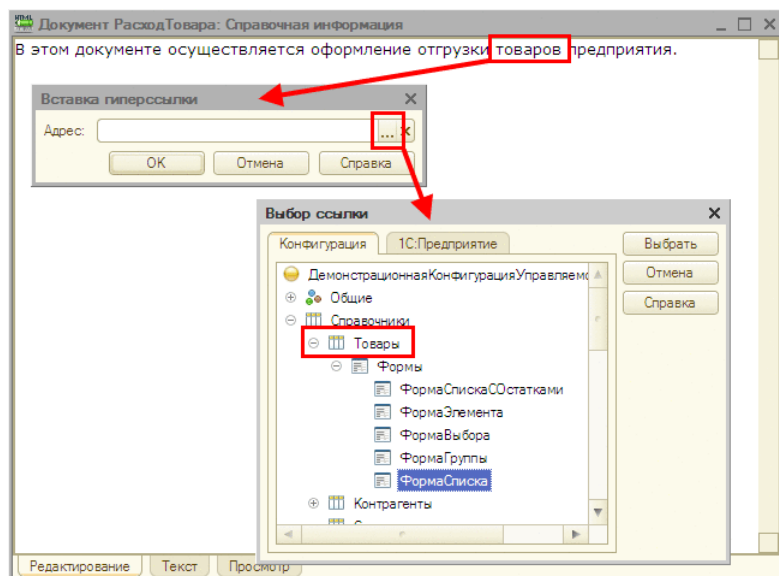


Рис. 427. Вставка гиперссылки

В реквизите [Адрес](#) нужно указать:

- Адрес гиперссылки.
- Ссылку на справочную информацию по объекту конфигурации.
- Ссылку на системный раздел справки. Если режим редактирования конфигурации установлен в значение [Управляемое приложение](#), то будут отображены только разделы справки, относящиеся к управляемому приложению (а также колонка, описывающая доступность раздела в веб-клиенте). Если режим редактирования конфигурации установлен в значение [Управляемое приложение и обычное приложение](#), то будут отображены все разделы справки с указанием доступности раздела в управляемом приложении, веб-клиенте и обычном приложении.
- Имя закладки (предваренное символом «#»).

Для создания ссылки следует нажать кнопку **ОК**.

Выделенный объект форматируется в соответствии со стилем гиперссылки. Обычно это синий цвет текста с подчеркиванием. Картинка помещается в рамку синего цвета.

27.20.1.6. Закладки

Для создания внутренней ссылки в HTML-документе можно создавать закладки. Для этого следует выделить объект (текст или картинку) и выбрать пункт [Элементы – Закладка](#). На экран выводится диалог ввода имени закладки. Выделенный объект при этом не форматируется.

Закладки используются для организации внутренней ссылки.

27.20.1.7. Линия

С помощью линий производится зрительное разделение документа.

27.20.2. Редактирование в HTML-формате

Возможности редактора HTML-документов, предоставляемые на закладке [Редактирование](#), ограничены. На закладке [Текст](#) разработчик может отредактировать документ в формате HTML-кода.

Следует помнить, что ручное изменение кодировки с помощью атрибута `charset` тега `META` не оказывают влияние на поведение редактора. При сохранении HTML-документ будет записан в кодировке utf-8 с неизменным тегом `charset`, а при чтении тег `charset` будет установлен в значение utf-8.

Для просмотра результата редактирования HTML-документа нужно перейти на закладку [Просмотр](#) и проверить работу объектов документа.

27.21. Локализация конфигураций

Под локализацией конфигураций понимается формирование строковых значений, появляющихся в программе в режиме 1С:Предприятие, на языках, указанных в ветви [Общие – Языки](#) дерева объектов конфигурации. Это может быть наименование пунктов в интерфейсе, наименованиях (синонимах) объектов, справочной информации, текстах модулей и т. д.

Наиболее сложным при выполнении данной работы является поиск мест, в которых необходимо ввести текст на требуемом языке. Поэтому данным режимом удобно пользоваться даже тогда, когда определен только один язык.

Для начала поиска нужно выбрать пункт [Правка – Редактирование текстов интерфейса](#).

На экран выводится диалог:

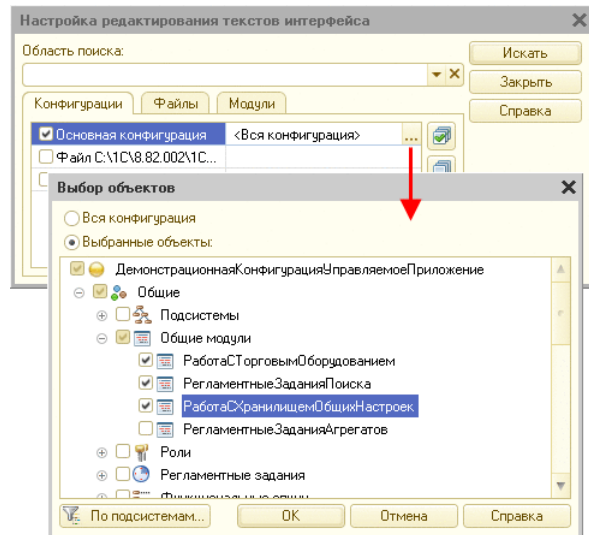


Рис. 428. Редактирование текстов интерфейса

На закладке [Конфигурации](#) выбираются объекты конфигурации, в которых требуется осуществить редактирование текстов интерфейса.

В список конфигураций будут включены все открытые на текущий момент окна конфигураций (помимо основной это может быть конфигурация базы данных, конфигурации, расположенные в файлах, конфигурации хранилища и поставки).

В поле выбора можно выбрать строку [Вся конфигурация](#), и в этом случае будет сформирован полный список объектов конфигурации, которые содержат интерфейсные свойства. Поиск можно осуществлять только в определенных объектах, если в поле выбора выбрать этот объект.

На закладке [Файлы](#) можно выбрать текстовые и табличные документы, внешние обработки, размещенные в файлах.

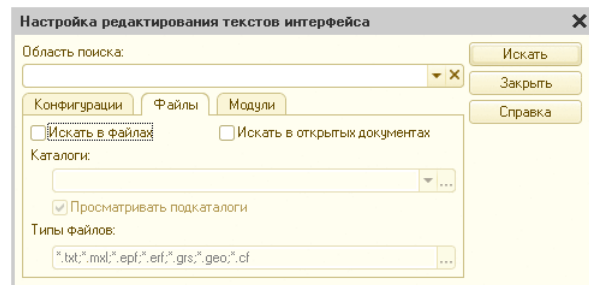


Рис. 429. Настройка поиска в файлах

На закладке [Модули](#) нужно установить флажок [Искать в функциях "НССтр" \("NStr"\) в модулях](#), если требуется определить места использования оператора `НССтр()` в модулях.

Для запоминания области поиска (список объектов конфигураций, файлов и открытые документы) в реквизите [Область поиска](#) следует указать наименование текущих установок. При повторном открытии окна поиска в списке областей достаточно выбрать нужную и выполнить поиск.

После того как выбраны нужные объекты, следует нажать кнопку [Искать](#). На экран выводится окно:

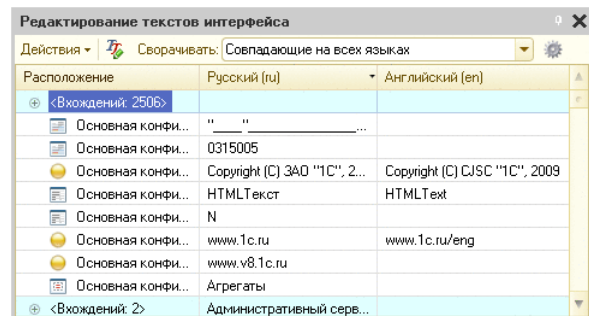


Рис. 430. Окно с результатами поиска

Окно содержит табличное поле, в первой колонке которого выводится расположение найденного текста. Другие колонки обычно соответствуют указанным языкам. В табличном поле может быть больше колонок, чем определено языков. Дополнительные колонки появляются в том случае, когда в процессе конфигурирования создавались новые объекты [Язык](#), которые затем либо удалялись, либо у них менялся код языка (при удалении или смене кода языка объектов данного типа конфигурировщик не сбрасывает текст интерфейсов, введенных для этих языков).

Содержимое табличного поля отсортировано по одной из колонок. Для смены сортировки достаточно щелкнуть заголовок колонки. При повторном щелчке заголовка производится смена направления сортировки.

Текст можно также изменить прямо в ячейке. Для этого нужно выбрать ячейку и нажать клавишу **Enter**. Поле ввода переводится в режим редактирования. Следует ввести нужный текст и снова нажать клавишу **Enter**. Измененный текст показывается красным цветом.

Для быстрого доступа к тексту интерфейса определенного объекта достаточно в колонке **Расположение** дважды щелкнуть мышью нужную строку. На экран выводится форма, в которой данный текст используется. В палитре свойств можно также произвести просмотр и замену найденного текста. Если текст изменялся вне окна редактирования, то табличное поле можно обновить.

Над табличным полем расположены элементы управления, предназначенные для выполнения различных действий и настройки окна.

В поле выбора **Сворачивать** производится выбор режима сворачивания текстов на разных языках. Если выбрано значение **Нет**, то сворачивание не производится. Если выбрано **Совпадающие на языке сортировки**, то все элементы, имеющие одинаковый текст в колонке, по которой производится сортировка, сворачиваются. При этом в первой колонке показывается значок группы (+), щелкнув который можно развернуть группу. В первой колонке будет показан текст **<Вхождений N>** (где **N** – число вхождений). Если в других колонках по этой группе будут различные значения текста, то в ячейке этих колонок будет надпись **<Различные значения>**.

Если выбрано **Совпадающие на всех языках**, то все элементы, имеющие одинаковый текст, сворачиваются.

Для свернутых строк можно производить групповую замену текста. Для этого в ячейке нужного языка достаточно ввести его так, как если бы это была одна строка. Введенный текст заменяет текст сразу во всех строках, входящих в группу.

С помощью пункта **Действия – Новый поиск** открывается окно настройки для изменения разделов конфигурации, в которых будет произведен поиск.

С помощью пункта **Действия – Копировать тексты** производится полное копирование текстов одного языка в другой язык. Эта операция рекомендуется тогда, когда используемые слова и фразы на различных языках в основном совпадают.

Пункт **Действия – Очистить тексты** производит полную очистку текстов указанного языка.

Пункт **Действия – Заполнить тексты** производит оперативный перевод синонимов, заголовков, подсказок и текстов интерфейсов с использованием файла соответствия. Он представляет собой табличный документ, состоящий из нескольких колонок (по числу используемых языков).

В первой строке каждой колонки должен быть написан код языка (например, **ru** или **lv**). В остальных строках следуют образцы соответствий. Никаких требований к упорядочиванию (сортировке) строк не предъявляется. Строки могут дублироваться.

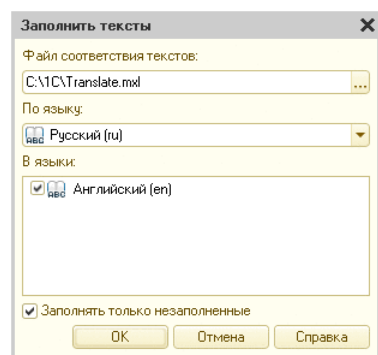


Рис. 431. Заполнение текстов

После выбора файла соответствия нужно указать основной язык (**По языку**) и языки, для которых требуется выполнить перевод текста (**В языки**). Установка флажка **Заполнять только незаполненные** приводит к блокировке перезаписи уже локализованных строк.

Выбор ключевого языка и языков для заполнения осуществляется из числа языков, которые были обнаружены в процессе поиска интерфейсных текстов. Требуется, чтобы все выбранные для заполнения языки были определены в файле соответствий (в нем должны быть колонки с заголовками, соответствующими кодам этих языков).

ПРИМЕЧАНИЕ. Если при открытии окна **Редактирование текстов интерфейса** обнаружены строки только на одном языке, команда **Заполнить тексты** будет недоступна.

Нажатие кнопки **OK** производит заполнение текстов интерфейсов. При этом выполняется заполнение текстов с проверкой соответствия не только по тому языку, на основании которого выполняется загрузка, но и по всем языкам, которые присутствуют в файле соответствия текстов.

Это позволяет, например, поставить в соответствие одинаковым строкам на одном языке разные строки на другом. Для этого можно в двуязычной конфигурации завести еще один вспомогательный язык для комментирования текстов интерфейса и по-разному заполнять такой комментарий для омонимов основного языка. Например, для слова «Счет» на русском языке можно написать комментарии **«бухгалтерский счет»** и **«документ счет»**. Это даст возможность при заполнении текстов из файла соответствия текстов сопоставить с первым из этих слов английское **«Account»**, а со вторым – **«Invoice»**.

Пункт **Действия – Экспорт в табличный документ** производит выгрузку содержимого табличного поля в табличный документ. Выполнение команды осуществляет вывод только уникальных строк.

С помощью пункта **Действия – Настройка** осуществляется настройка показа табличного поля и установка режима открытия редакторов при групповой замене.

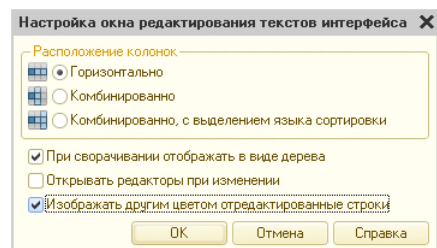


Рис. 432. Настройка окна редактирования текстов интерфейса

В группе **Расположение колонок** производится выбор способа расположения колонок:

- **Горизонтально** – это так, как показано на [рис. 430](#).
- **Комбинированно** – располагает колонки одна под другой.
- **Комбинированно, с выделением языка сортировки** – размещает колонки следующим образом: справа от колонки **Расположение** будет колонка с языком, по которому производится сортировка, а правее будут располагаться одна под другой колонки с другими языками. При щелчке мышью в области заголовка колонки выполняется сортировка по этой колонке, и она располагается справа от колонки **Расположение**. Колонка, по которой ранее выполнялась сортировка, занимает место выбранной колонки. Если языков всего два, то колонки просто меняются местами.

Если флажок **При сворачивании отображать в виде дерева** установлен, то доступ к свернутым строкам возможен, а свернутые группы показываются в виде дерева.

Рекомендуется не устанавливать флажок [Открывать редакторы при групповой замене](#), если число вхождений достаточно большое.

Если флажок [Изображать другим цветом отредактированные строки](#) установлен, то измененный текст строк будет показан другим цветом.

27.22. Выполнение централизованной проверки конфигурации

Для выполнения проверки конфигурации нужно выбрать пункт [Конфигурация – Проверка конфигурации](#). На экран выводится окно:

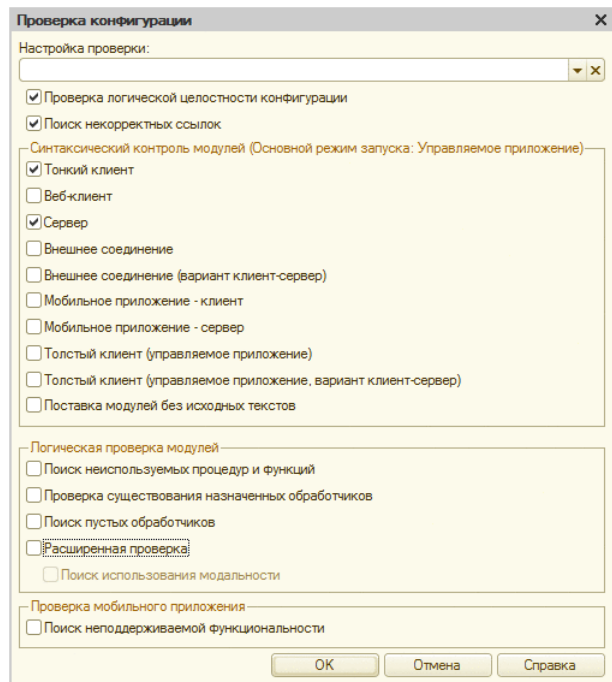


Рис. 433. Проверка конфигурации

Рассмотрим подробнее, как выполняется проверка конфигурации. Любой прикладной код в системе «1С:Предприятие» выполняется в некотором окружении. Это окружение характеризуется некоторым набором определенных инструкций препроцессора (см. [здесь](#)) и набором доступных объектов. Соответственно, при выполнении проверки конфигуризатор определяет инструкции препроцессора для каждого отмеченного пункта и выполняет проверку. Следует отметить, что если проверяется клиентское приложение в клиент-серверном варианте, то на стороне клиента проверяется весь прикладной код, а на стороне сервера – только те общие модули, которые доступны для вызова из клиентского приложения.

Так как различных режимов работы получается достаточно много, то проверки разделены на две группы:

- Группа проверок, доступная для принудительной и автоматической проверки при редактировании конфигурации. Эти режимы доступны для выбора на закладке [Модули – Проверка](#) диалога настройки параметров конфигуризатора (см. [здесь](#)).
- Весь список проверок, который можно вызвать при централизованной проверке конфигурации (описываемый режим).

Далее будет описано, какие инструкции препроцессора будут определены при выборе каждого режима проверки, описаны особенности режима (если они есть), а также описаны режимы проверки, не имеющие отношения к определению окружения проверяемого исходного кода.

Общие параметры

Проверка логической целостности конфигурации

Стандартная проверка, обычно выполняемая перед обновлением базы данных. При выполнении данной проверки выполняется удаление некорректной информации, которая могла оказаться в конфигурации после редактирования предыдущими версиями системы.

Поиск некорректных ссылок

Поиск ссылок на удаленные объекты. Выполняется по всей конфигурации, включая формы, макеты, интерфейсы и т. д. Также выполняется поиск некорректных ссылок на предопределенные данные для планов счетов (если для плана счетов задано свойство [Виды субконто](#)) и планов видов расчетов.

Синтаксический контроль модулей

Тонкий клиент

Проверка компиляции модулей в режиме проверки среды тонкого клиента в режиме управляемого приложения, выполняемого в файловом варианте.

Определены инструкции препроцессора: [Клиент](#), [ТонкийКлиент](#).

Веб-клиент

Проверка компиляции модулей в режиме проверки среды веб-клиента в режиме управляемого приложения, выполняемого в файловом варианте.

Определены инструкции препроцессора: [Клиент](#), [ВебКлиент](#).

Сервер

Проверка компиляции модулей в режиме проверки среды сервера «1С:Предприятия».

Определены инструкции препроцессора: [Сервер](#).

Внешнее соединение

Проверка компиляции модулей в режиме проверки среды внешнего соединения, выполняемого в файловом варианте.

Определены инструкции препроцессора: [ВнешнееСоединение](#), [Сервер](#).

Внешнее соединение (вариант клиент-сервер)

Проверка компиляции модулей в режиме проверки среды внешнего соединения, выполняемого в варианте клиент-сервер.

Определены инструкции препроцессора: [ВнешнееСоединение](#).

Мобильный приложение – клиент

Проверка компиляции модулей в режиме проверки среды клиентского приложения, выполняемого на мобильной платформе.

Определены инструкции препроцессора: [Клиент](#), [НаКлиенте](#), [МобильноеПриложениеКлиент](#).

Мобильное приложение – сервер

Проверка компиляции модулей в режиме проверки среды клиентского приложения, выполняемого на мобильной платформе.

Определены инструкции препроцессора: [Сервер](#), [НаСервере](#), [МобильноеПриложениеСервер](#).

Толстый клиент (управляемое приложение)

Проверка компиляции модулей в режиме проверки среды управляемого клиента, выполняемого в файловом варианте.

Если свойство конфигурации [Использовать обычные формы в управляемом приложении](#) имеет значение [Ложь](#), то модули обычных форм не проверяются при выборе этого теста.

Определены инструкции препроцессора: [Клиент](#), [ТолстыйКлиентУправляемоеПриложение](#), [Сервер](#).

Толстый клиент (управляемое приложение, вариант клиент-сервер)

Проверка компиляции модулей в режиме проверки среды управляемого клиента, выполняемого в варианте клиент-сервер.

Если свойство конфигурации [Использовать обычные формы в управляемом приложении](#) имеет значение [Ложь](#), то модули обычных форм не проверяются при выборе этого теста.

Определены инструкции препроцессора: [Клиент](#), [ТолстыйКлиентУправляемоеПриложение](#).

Поставка модулей без исходных текстов

Если в настройках поставки конфигурации для некоторых модулей указана поставка без исходных текстов, проверяется возможность генерации образов этих модулей.

Логическая проверка модулей

Поиск неиспользуемых процедур и функций

Поиск локальных (не экспортируемых) процедур и функций, на которые отсутствуют ссылки. В том числе осуществляется поиск неиспользуемых обработчиков событий.

Процедура или функция модуля формы будет считаться используемой, если ее имя встречается в виде ["ИмяПроцедуры"](#) (включая кавычки), т. е. является параметром какого-либо метода, например, назначение обработчика какого-либо события.

Проверка существования назначенных обработчиков

Проверка существования обработчиков событий интерфейсов, форм, элементов управления, элементов карт маршрута.

Поиск пустых обработчиков

Поиск назначенных обработчиков событий, в которых не выполняется никаких действий. Существование таких обработчиков может привести к падению производительности системы.

Расширенная проверка

Для ограниченного набора типов выполняется проверка обращений к методам и свойствам объектов «через точку». Также выполняется проверка корректности указания параметров, заданных строкой, для ограниченного набора методов.

Поиск использования модальности

Данный флажок становится доступным только в том случае, если включена расширенная проверка конфигурации. В этом режиме конфигуратор выполняет поиск в модулях методов, открывающих модальные диалоговые окна. Свойство конфигурации [Режим использования модальности](#) не влияет на работу механизма проверки. При включении данного флажка ошибки будут выдаваться и в том случае, если свойство [Режим использования модальности](#) установлено в значение [Использовать](#) или [Использовать с предупреждением](#).

Проверка мобильного приложения

Поиск неподдерживаемой функциональности

Проверка в этом режиме показывает:

- Наличие в прикладном решении объектов конфигурации, которые не реализованы для мобильной платформы;
- Наличие в прикладном решении планов обмена, у которых установлено свойство [Распределенная информационная база](#);
- Использование объектов конфигурации, которые не поддерживаются мобильной платформой:
 - В свойствах [Тип](#) реквизитов объектов конфигурации, констант, параметров сеанса;
 - В свойстве [Тип параметра команды](#) объекта [Команда](#);
 - В свойстве [Тип](#) реквизитов и колонок реквизита формы.
- Наличие неуправляемых форм;
- Наличие в формах элементов, которые не поддерживаются мобильной платформой. Эта проверка не делается для форм, у которых свойство [Назначения использования](#) не предполагает использование на мобильной платформе.
- Проверка, что начальная страница состоит не более чем из одной формы.

Выборенную совокупность настроек можно сохранить для дальнейшего использования. Для этого в поле [Настройка проверки](#) нужно указать имя настройки. Для использования прежней настройки достаточно выбрать имя настройки.

Все сообщения об ошибках выдаются в окно сообщений. Для прерывания проверки конфигурации следует использовать комбинацию клавиш [Ctrl + Break](#).

ПРИМЕЧАНИЕ. В начальной стадии проверки прерывание может быть обработано с задержкой.

При проверке конфигурации, подключенной к хранилищу, во избежание ошибок, связанных с информацией о метаданных, рекомендуется захватить корневой объект конфигурации.

Если параметр [Редактирование конфигурации для режимов запуска](#) (см. [здесь](#)) имеет значение [Управляемое приложение и обычное приложение](#), то в диалоге добавляются следующие параметры:

- Толстый клиент (обычное приложение);
- Толстый клиент (обычное приложение, вариант клиент-сервер).

При этом диалог приобретает следующий вид:

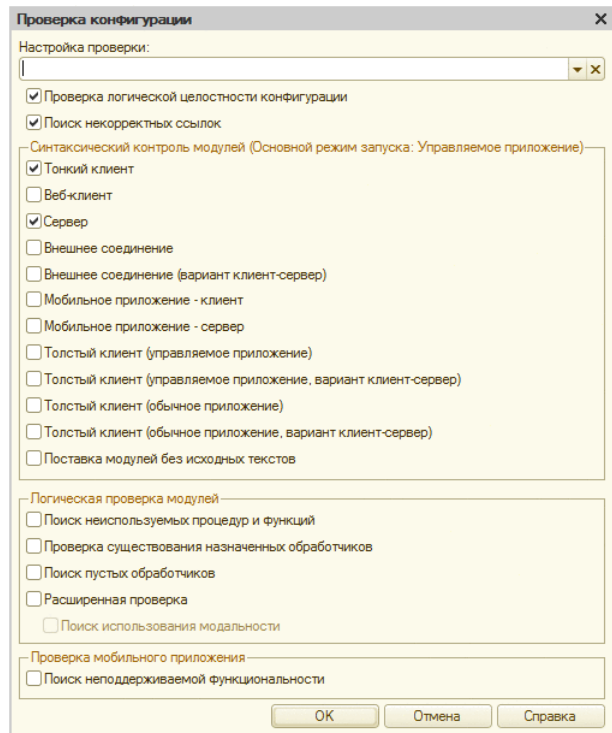


Рис. 434. Проверка конфигурации

Далее приводится описание этих тестов.

Толстый клиент (обычное приложение)

Проверка компиляции модулей в режиме проверки среды клиентского приложения, выполняемого в файловом варианте.

Если свойство конфигурации **Использовать управляемые формы в обычном приложении** имеет значение *Ложь*, то модули управляемых форм и модули команд не проверяются при выборе этого теста. Модуль формы, указанной в свойстве **Основная форма констант**, проверяется при выборе этого теста независимо от значения свойства **Использовать управляемые формы в обычном приложении**.

Определены инструкции препроцессора: *Клиент, ТолстыйКлиентОбычноеПриложение, Сервер*.

Толстый клиент (обычное приложение, вариант клиент-сервер)

Проверка компиляции модулей в режиме проверки среды клиентского приложения, выполняемого в варианте клиент-сервер.

Если свойство конфигурации **Использовать управляемые формы в обычном приложении** имеет значение *Ложь*, то модули управляемых форм и модули команд не проверяются при выборе этого теста. Модуль формы, указанной в свойстве **Основная форма констант**, проверяется при выборе этого теста независимо от значения свойства **Использовать управляемые формы в обычном приложении**.

Определены инструкции препроцессора: *Клиент, ТолстыйКлиентОбычноеПриложение*.

27.23. Механизм анализа эргономики работы с решениями, созданными на платформе «1С:Предприятие»

Запуск клиентского приложения с ключом командной строки **/logui** создаст на компьютере пользователя в папке файлов приложений **%APPDATA%\1C\1Cv82\<Уникальный идентификатор>** файл **logui.txt**, в котором будут регистрироваться все интерактивные действия пользователя (нажатия клавиш, щелчки мыши). Например:

[Копировать в буфер обмена](#)

```
"19.06.2008 16:22:49", "Event FormActivate", "Name Документ.ПриходТовара.Форма.ФормаСписка", "t=0"
"19.06.2008 16:22:49", "Event FormActivate", "Name Документ.ПриходТовара.Форма.ФормаСписка", "t=0"
"19.06.2008 16:22:49", "Event Lclick", "Form РегистрНакопления.Взаиморасчеты.Форма.ТекущиеВзаиморасчеты", "Type TableBox", "t=0", "beg"
```

Запись интерактивных действий выполняется для толстого и тонкого клиентов и не выполняется для веб-клиента.

Подробнее о формате файла **logui.txt** можно посмотреть в книге «1С:Предприятие 8.3. Руководство администратора».

Глава 28. Отладка и тестирование прикладных решений

Отладка – этап разработки прикладного решения, в процессе которого происходит обнаружение, локализация и устранение ошибок в прикладном решении.

В процессе отладки можно пользоваться отладчиком (см. [здесь](#)), использовать результаты замера производительности (см. [здесь](#)), использовать механизмы отображения вызовов сервера (см. [здесь](#)), имитировать задержки при вызове сервера (см. [здесь](#)). Также существует возможность выполнить сценарий автоматизированного тестирования (см. [здесь](#)).

В данной главе приводится описание инструментов платформы, которые помогают в процессе отладки прикладного решения и его тестирования.

28.1. Отладчик

28.1.1. Общая информация

Отладчик – инструмент, облегчающий разработку и отладку программных модулей системы «1С:Предприятие». Отладчик предоставляет следующие возможности:

- возможность отладки выполнения модулей, как в файловом, так и в клиент-серверном варианте, фоновых заданий;
- пошаговое выполнение модуля;
- расстановка точек останова;
- прерывание и продолжение выполнения модуля;
- возможность отладки нескольких модулей одновременно;
- вычисление выражений для анализа состояния переменных;
- просмотр стека вызовов процедур и функций;
- возможность остановки по возникновению ошибки;
- возможность редактирования модуля в процессе отладки.

В отладчике используется понятие предмета отладки. **Предмет отладки** – это контекст встроенного языка, характеризуемый совокупностью параметров:

- имя пользователя, от имени которого исполняется код на встроенном языке;
- тип предмета отладки;
- сетевое имя компьютера, на котором исполняется код на встроенном языке;
- номер используемого сеанса;
- номер IP-порта, через который отладчик управляет работой предмета отладки.

ПРИМЕЧАНИЕ. Если в клиент-серверном варианте и при наличии нескольких рабочих процессов в кластере, в процессе отладки кода веб-клиента или сервера сеанс был переназначен на другой рабочий процесс, то отладка становится невозможной. Подробнее о сеансах можно прочитать в книге «1С:Предприятие 8.3. Клиент-серверный вариант. Руководство администратора».

К типам предметов отладки относятся:

- **тонкий клиент** – код на встроенном языке, исполняемый в тонком клиенте;
- **веб-клиент** – код на встроенном языке, исполняемый в веб-клиенте;
- **сервер** – код на встроенном языке, исполняемый на сервере;
- **сервер (файловый вариант)** – код на встроенном языке, исполняемый на сервере в файловом варианте;
- **СОМ-соединение** – код на встроенном языке, исполняемый через внешнее соединение;

- **Web-сервис** – код на встроенном языке, в котором обрабатываются вызовы методов Web-сервисов;
- **фоновое задание** – код на встроенном языке, в котором обрабатываются фоновые задания;
- **толстый клиент** – код на встроенном языке, исполняемый в толстом клиенте.

28.1.2. Использование отладчика

28.1.2.1. Общая схема

Чтобы иметь возможность отлаживать код на встроенном языке, нужно обеспечить запуск приложения, в котором исполняется код, в отладочном режиме.

ВНИМАНИЕ! Для работы режима отладки необходимо, чтобы на компьютере была включена поддержка сетевого протокола TCP/IP. Включить поддержку протокола TCP/IP можно в настройках соответствующего подключения к сети.

Если режим 1С:Предприятие не запущен, то для начала отладки нужно выбрать пункт [Отладка – Начать отладку](#). Конфигуратор запускает режим 1С:Предприятие в отладочном режиме.

Если в настройках конфигуратора установлен режим разрешения отладки или указано, что отладка будет начата при запуске (окно настроек открывается с помощью команды [Сервис – Параметры](#) закладка [Запуск 1С:Предприятия](#), см. [здесь](#)), то для начала отладки также можно использовать режим запуска, выполняемый командой [Сервис – 1С:Предприятие](#). Если требуется выполнить отладку кода, выполняемого определенным пользователем, то в форме настроек можно указывать пользователя, от лица которого запускается отладочный режим.

Отладчик и предметы отладки при поиске друг друга используют IP-адрес [127.0.0.1](#). Для устойчивой работы службы транспорта данных необходимо определить соответствие адреса [127.0.0.1](#) символическому имени [localhost](#). Для этого можно в файл [C:\Windows\system32\drivers\etc\hosts](#) вписать соответствие:

```
127.0.0.1 localhost
```

[Копировать в буфер обмена](#)

DNS-сервер используется, либо когда в диалоге [Предметы отладки](#) имя компьютера, на котором нужно искать предметы отладки, указано не в виде IP-адреса, либо когда в xml-файлах настройки отладки COM-соединений и Web-сервисов отладчик указан не в виде IP-адреса. Т. е. URL отладчика используется для COM-соединений и Web-сервисов именно в том виде, как она записан в файле. Если там записано имя компьютера, а не его IP-адрес, то по-прежнему будут выполняться обращения к DNS-серверу.

Для обеспечения возможности отладки необходимо разрешить использование портов в диапазоне [1560—1591](#) (по умолчанию) на компьютере, где установлен конфигуратор, и на компьютерах, на которых расположены используемые предметы отладки. Если для отладки выделяется другой диапазон портов с помощью файла [debugcfg.xml](#) (см. [здесь](#)), разрешить использование необходимо для нового диапазона портов. Кроме того, этот же диапазон портов следует указать на всех компьютерах, где расположены предметы отладки (кроме серверных предметов отладки, которые используют для работы диапазон портов, заданный при настройке сервера «1С:Предприятия»).

Если на компьютере, где установлен конфигуратор, запрещено использование портов, необходимых отладчиком, то отладка на этом компьютере будет недоступна. Также может наблюдаться следующее:

- увеличение времени запуска клиентского приложения;
- замедление рабочего процесса сервера «1С:Предприятия» во время работы, если сервер работает в режиме отладки (ключ [-debug](#)).

28.1.2.2. Настройка приложения для работы в отладочном режиме

28.1.2.2.1. Отладка клиентского приложения

Для установки отладочного режима можно использовать следующие варианты запуска:

- В режиме Конфигуратор в форме настроек (меню [Сервис – Параметры](#)), на закладке [Запуск 1С:Предприятия – Дополнительно](#) установить флажок [Устанавливать режим разрешения отладки](#), далее выполнить подключение предмета отладки. Также можно установить флажок [Начинать отладку при запуске](#); в этом случае при запуске системы «1С:Предприятие» подключение будет выполнено автоматически.
- Открыть информационную базу в режиме 1С:Предприятие с ключом командной строки [/debug](#) (отладочный режим).
- Если запущено клиентское приложение, то в форме настроек (открыть с помощью меню [Сервис – Параметры](#)) установить отладочный режим (установить флажок [Отладка разрешена](#)). Следует иметь в виду, что после применения настроек снять флажок нельзя.

- Если режим 1С:Предприятие уже запущен и требуется установить возможность отладки только для следующего запуска, следует в форме настроек (меню [Сервис – Параметры](#)), на закладке [Системные](#) установить флажок [Устанавливать режим разрешения отладки при запуске](#).

28.1.2.2.2. Отладка кода на сервере

В отладочном режиме загрузка объектов конфигурации производится по мере необходимости, а не при начале работы системы, как в обычном режиме работы сервера. Это ускоряет процесс запуска «1С:Предприятия» при изменении конфигурации, то есть ускоряет процесс разработки.

Также следует учитывать, что в отладочном режиме производительность системы будет ниже, чем при обычной работе. Поэтому не рекомендуется использовать отладочный режим работы сервера для реальной работы пользователей.

Расположение и параметры конфигурационного файла [srv1cv83](#), а также о параметрах командной строки запуска сервера «1С:Предприятия» можно посмотреть в книге «1С:Предприятие 8.3. Клиент-серверный вариант. Руководство администратора».

Сервер как приложение

Если сервер «1С:Предприятия» работает в режиме приложения (в любой ОС), то следует остановить сервер и перезапустить его так, чтобы в числе параметров командной строки был параметр [-debug](#).

[Копировать в буфер обмена](#)

```
Ragent -debug <остальные параметры командной строки>
```

Сервер как служба ОС Windows

Если сервер работает как служба ОС Windows, то необходимо выполнить следующие операции:

1. Остановить сервер «1С:Предприятия».

[Копировать в буфер обмена](#)

```
Ragent -stop
```

2. Выполнить повторную регистрацию агента сервера в качестве службы Windows таким образом, чтобы в числе параметров команды [ragent](#) был параметр [-debug](#).

[Копировать в буфер обмена](#)

```
Ragent -instsrv -debug <остальные параметры командной строки>
```

3. Запустить сервер «1С:Предприятия».

[Копировать в буфер обмена](#)

```
Ragent -start
```

Сервер как «демон» ОС Linux

Если в отладочный режим необходимо перевести сервер «1С:Предприятия» работающего в режиме «демона» в ОС Linux, то необходимо выполнить следующие операции:

1. Остановить сервер «1С:Предприятия».

[Копировать в буфер обмена](#)

```
/etc/init.d/srv1cv83 stop
```

2. В конфигурационном файле [srv1cv83](#) установить значение параметра [SRV1CV8_DEBUG](#) в значение [1](#).

[Копировать в буфер обмена](#)

```
SRV1CV8_DEBUG=1
```

3. Сохранить конфигурационный файл.

4. Запустить сервер «1С:Предприятия».

[Копировать в буфер обмена](#)

```
/etc/init.d/srv1cv83 start
```

28.1.2.2.3. Отладка внешнего соединения

Для указания внешнему соединению необходимости запуска в отладочном режиме используются настройки, размещенные в xml-файле [comcntrcfg.xml](#), который расположен в каталоге конфигурационных файлов системы «1С:Предприятие». Если файл не найден, приложение открывается в обычном режиме.

```
<config xmlns="http://v8.1c.ru/v8/comcntrcfg">
  <debugconfig debug="true" debuggerURL="tcp://localhost:1560"/>
</config>
```

Подробнее о файле `comcntrcfg.xml` можно посмотреть в книге «1С:Предприятие 8.3. Руководство администратора».

Если для элемента `debug` указан атрибут `url`, то к моменту обращения к внешнему соединению по указанному URL должен быть доступен конфигуратор. В противном случае исполнение внешнего соединения будет приостановлено и система будет ожидать доступность конфигулятора. При этом для продолжения работы в конфигураторе следует подключить необходимый предмет отладки.

28.1.2.2.4. Отладка Web-сервиса, HTTP-сервиса и доступа через стандартный интерфейс OData

Для указания сервису необходимости запуска в отладочном режиме (только для файлового варианта информационной базы) используются настройки, размещенные в файле `default.vrd`, который должен располагаться в каталоге виртуального приложения. В этом файле необходимо указать элемент `debug`, отсутствие которого означает невозможность отладки Web-сервиса.

Пример элемента `debug` из файла `default.vrd`:

```
<debug enable="true" url="tcp://localhost"/>
```

Подробнее о файле `default.vrd` можно посмотреть в книге «1С:Предприятие 8.3. Руководство администратора».

Если для элемента `debug` указан атрибут `url`, то в момент обращения к сервису, по указанному URL должен быть доступен конфигуратор. В противном случае исполнение сервиса будет приостановлено и система будет ожидать доступность конфигулятора. При этом для продолжения работы в конфигураторе следует подключить необходимый предмет отладки.

Для необходимости отладки сервиса в клиент-серверном варианте информационной базы следует включить режим отладки на сервере с помощью ключа командной строки `debug`. Подробнее о ключах командной строки запуска сервера «1С:Предприятия» можно посмотреть в книге «1С:Предприятие 8.3. Клиент-серверный вариант. Руководство администратора».

ПРИМЕЧАНИЕ. Для отладки HTTP-сервиса и работы со стандартным интерфейсом OData не рекомендуется использовать веб-браузер Microsoft Internet Explorer 9.0, т. к. в данном веб-браузере имеются ошибки кодирования URL.

28.1.2.2.5. Отладка веб-клиента

Для установки отладочного режима можно использовать следующие варианты запуска:

- В режиме Конфигуратор в форме настроек (меню [Сервис – Параметры](#)), на закладке [Запуск 1С:Предприятия – Дополнительно](#) установить флажок [Устанавливать режим разрешения отладки](#), далее выполнить подключение предмета отладки. Также можно установить флажок [Начинать отладку при запуске](#). В этом случае при запуске системы «1С:Предприятие» подключение будет выполнено автоматически.
- Открыть информационную базу в режиме 1С:Предприятие с ключом командной строки `debug` (отладочный режим):

```
http://localhost/demo?debug
```

- Открыть информационную базу в режиме 1С:Предприятие с ключами командной строки `debug` и `debuggerurl="ip-адрес"`:

```
http://localhost/demo?debug&debuggerurl="127.0.0.1"
```

- [ip-адрес](#) – это адрес отладчика. В случае такого запуска отладчик, расположенный по адресу [127.0.0.1](#), выполнит автоматическое подключение предметов отладки (отладчик должен быть запущен по указанному адресу).

Если необходимо включить отладку серверной части файлового варианта информационной базы, то это можно сделать двумя способами:

- Запустить отладку непосредственно из конфигулятора, как указано выше. В этом случае режим отладки для серверной части файлового варианта, расположенной на компьютере с веб-клиентом, включится автоматически.
- Включить режим отладки с помощью файла `default.vrd`, который должен располагаться в каталоге виртуального

приложения. В этом файле необходимо указать элемент **debug**, отсутствие которого означает невозможность отладки Web-сервиса.

Пример элемента *debug* из файла *default.vrd*:

[Копировать в буфер обмена](#)

```
<debug enable="true" url="tcp://192.168.0.30"/>
```

Также следует учитывать, что в отладочном режиме производительность системы будет ниже, чем при обычной работе. Поэтому не рекомендуется использовать отладочный режим работы серверной части файлового варианта работы для реальной работы пользователей.

Подробнее о файле **default.vrd** можно посмотреть в книге «1С:Предприятие 8.3. Руководство администратора».

Во время отладки веб-клиента нужно помнить следующие особенности:

- не поддерживается возможность интерактивного включения режима отладки;
- не поддерживается автоматический поиск предметов отладки на удаленных компьютерах при отладке веб-клиента для файлового варианта информационной базы;
- не поддерживается активизация конфигуратора из веб-клиента в процессе отладки;
- не поддерживается отладка внешних отчетов и обработок;
- в случае принудительного прекращения работы веб-клиента из отладчика, прекращается работа того процесса веб-браузера, который был запущен при старте веб-клиента из конфигуратора.

ВНИМАНИЕ! Для отладки веб-клиента (в том числе и клиентского программного кода) в клиент-серверном варианте необходимо, чтобы сервер «1С:Предприятия» был запущен в отладочном режиме (ключ **debug**).

ПРИМЕЧАНИЕ. Не рекомендуется использовать отладку веб-клиента на низкоскоростных каналах связи.

Подробнее о клиент-серверном варианте работы можно посмотреть в книге «1С:Предприятие 8.3. Руководство администратора».

28.1.2.2.6. Отладка внешней обработки (отчета)

Для отладки внешнего отчета (обработки) необходимо соблюдение двух условий:

1. Внешняя обработка (отчет) открыты в клиентском приложении из конкретного места файловой системы;
2. Конфигуратор, который обслуживает предмет отладки с отлаживаемой внешней обработкой (отчетом), имеет доступ именно к тому файлу внешней обработки (отчета), который открыт клиентским приложением.

Если эти условия не выполняются – отладка внешней обработки (отчета) будет невозможна. В частности, не поддерживается отладка внешних обработок (отчетов), расположенных в информационной базе. В этом случае рекомендуется извлечь внешнюю обработку (отчет) из информационной базы на время отладки.

28.1.2.3. Подключение предметов отладки

Для выполнения отладки модуля нужно, чтобы предмет отладки был подключен. Для управления подключением нужно выбрать пункт **Отладка – Подключение**. На экран выводится окно для выбора предмета отладки. В список попадают только те предметы отладки, для которых выполняются следующие условия:

- Строка подключения к информационной базе для конфигуратора и предмета отладки полностью совпадает в части параметров, описывающих расположение информационной базы (параметр **File** для файлового варианта и параметры **Srvr** и **Ref** для клиент-серверного варианта), **включая регистр текста**.

Примеры несовпадающих строк соединения:

[Копировать в буфер обмена](#)

```
// сетевой диск R: подключен к папке \\Server\database,
// при этом следующие адреса не одинаковы с точки зрения отладчика:
File=\\Server\database\mybase;
File=R:\mybase;
// компьютер db_server имеет ip-адрес 192.168.1.1,
// при этом следующие адреса не одинаковы с точки зрения отладчика:
Srvr=db_server;Ref=my_db;
Srvr=192.168.1.1;Ref=my_db;
```

- В приложении должна быть включена возможность отладки (для сервера – флажок **debug**, для клиентского

приложения – соответствующий параметр командной строки, или флажок диалога настройки клиентского приложения, или корректно настроенный конфигурационный файл).

Предметы отладки доступны для подключения на протяжении интервала времени существования сеанса, в рамках которого работает конкретное фоновое задание, Web-сервис или внешнее соединение. Это интервал может длиться очень незначительно время, и в этом случае можно воспользоваться настройкой автоматического подключения предметов отладки.

ПРИМЕЧАНИЕ. При подключении предмета отладки в списке пользователей информационной базы должны существовать пользователи, от лица которых выполнена авторизация как в конфигураторе, так и в предмете отладки.

Обычно список содержит одну строку с указанием на запущенную в режиме 1С:Предприятие конфигурацию. Если запущено несколько приложений системы «1С:Предприятие» с данной конфигурацией, то список может содержать несколько строк.

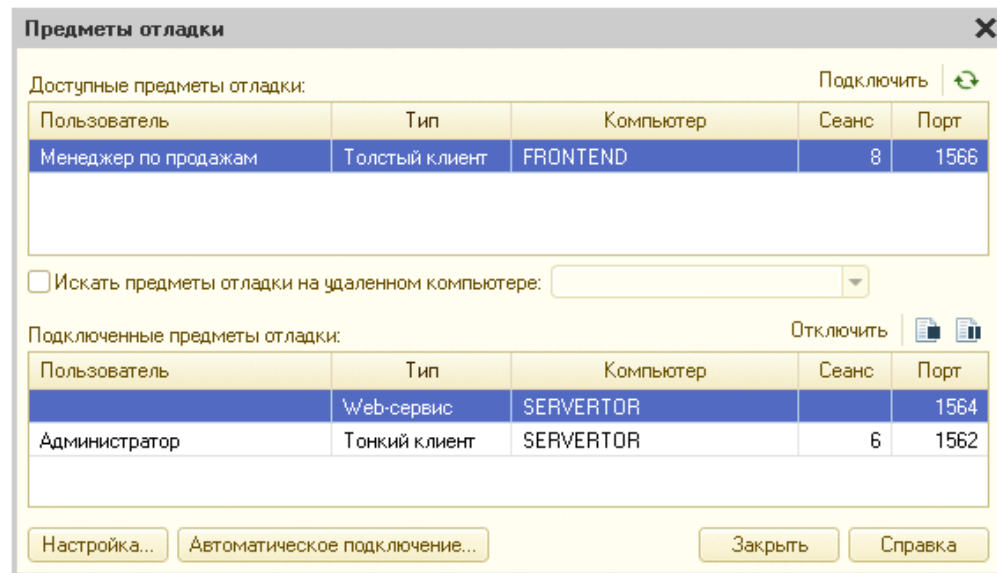


Рис. 435. Подключение предметов отладки

Если флажок [Искать предметы отладки на удаленном компьютере](#) установлен, то в поле, расположенном справа от флажка, следует ввести имя компьютера (или его сетевой адрес) или выбрать имя из ранее вводимых имен. При этом в список доступных предметов отладки будут добавлены предметы отладки, найденные на удаленном компьютере. Список подключенных предметов отладки будет содержать те предметы отладки, которые уже подключены к отладчику.

Нажатие кнопки [Подключить](#) подключает к отладчику выбранный предмет отладки. В окне подключения это отображается переносом предмета отладки из списка доступных в список подключенных предметов отладки.

Для исключения предмета отладки нужно указать его в списке подключенных и нажать кнопку [Отключить](#). В окне подключения это отображается переносом предмета отладки из списка подключенных в список доступных предметов, и к нему можно повторно подключиться. При этом точки останова, установленные в отключенных предметах отладки, не будут «срабатывать» при прохождении выполнения через них.

Для закрытия предмета отладки следует нажать кнопку [Завершить](#), для останова в месте выполнения – кнопку [Остановить](#).

Чтобы открыть диалог настройки диапазона, следует нажать кнопку [Настройки](#). Диапазон определяет границы, в рамках которых отладчик ищет предметы отладки на текущем или указанном компьютере.

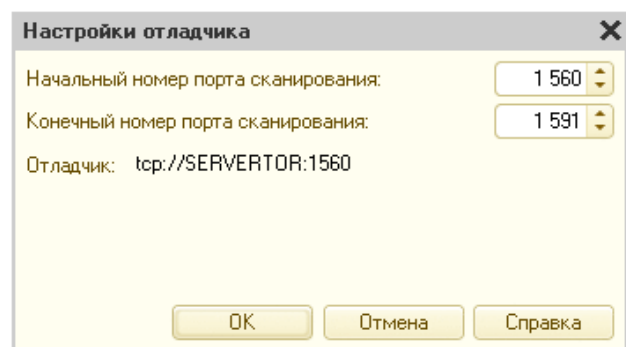


Рис. 436. Настройка отладчика

В поле [Отладчик](#) диалога содержатся настройки текущего отладчика. Их можно использовать, например, в командной строке при запуске клиентского приложения в качестве параметра ключа командной строки `/debuggerurl` или в xml-файле с настройками отладки для внешнего соединения или Web-сервиса.

Для автоматического подключения предметов отладки на сервере системы «1С:Предприятие», работающем в отладочном режиме, можно воспользоваться диалогом [Автоматическое подключение](#) и отметить в нем соответствующие типы предметов отладки.

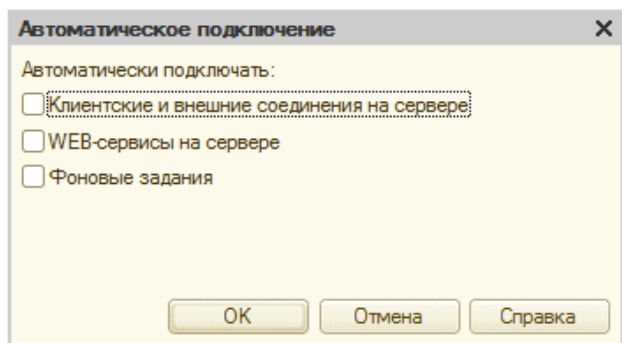


Рис. 437. Настройка автоматического подключения

Автоматическое подключение фоновых заданий, выполняющегося клиентским приложением или расширением веб-сервера, данным диалогом не регулируется. Для отладки фоновых заданий необходимо одновременное выполнение нескольких условий:

- Для тонкого или толстого клиента:
 - Разрешить отладку в клиентском приложении;
 - Разрешить автоматическое подключение к фоновым заданиям в диалоге настройки автоматического подключения.
- Для веб-сервера:
 - В файле публикации ([default.vrd](#)) разрешить отладку для отлаживаемой информационной базы;
 - Разрешить автоматическое подключение к фоновым заданиям в диалоге настройки автоматического подключения.

28.1.2.4. Дополнительная настройка диапазона портов

Если все порты для подключения в стандартном диапазоне заняты, существует возможность указать дополнительный диапазон. Этот диапазон настраивается в файле [debugcfg.xml](#), расположенном в каталоге конфигурационных файлов компьютера, на котором расположен конфигуратор, отлаживаемое клиентское приложение или выполняется внешнее соединение. Если диапазон портов изменен на компьютере с отлаживаемым приложением, то эти же порты следует установить и для конфигулятора, из которого выполняется отладка (если отлаживаемое приложение и конфигуратор расположены на разных компьютерах). Если файл не найден, то для отладки используются порты из стандартного диапазона (**1560—1591**). Предметы отладки на сервере используют те же порты, что и процессы сервера: [rmngr](#) и [rphost](#). Указания дополнительных диапазонов портов для предметов отладки на сервере не требуется.

Пример:

[Копировать в буфер обмена](#)

```
<config xmlns="http://v8.1c.ru/v8/debugcfg">
  <debugports range="1540:1550"/>
</config>
```

Подробнее о файле [debugcfg.xml](#) можно посмотреть в книге «1С:Предприятие 8.3. Руководство администратора».

28.1.3. Точка останова

Точка останова – это место в программном модуле, в котором исполнение программного модуля останавливается и управление передается отладчику.

В левой колонке текста модуля специальными значками показываются текущая точка останова, места останова и состояние точек останова.



Рис. 438. Виды точек останова

Точку останова можно установить в любой строке модуля, в любой момент работы с отладчиком. В том случае, если строка, на которой устанавливается точка останова, не содержит операторов (например, пустая строка), содержит неисполняемый текст (например, заголовок процедуры или функции, определение переменных) или является продолжением оператора, начатого на предыдущих строках, положение точки останова будет автоматически скорректировано. Место нахождения точки останова отмечается специальным знаком в левой колонке окна модуля. Для включенных и отключенных точек останова используются разные знаки (см. [рис. 438](#)).

Точку останова можно также установить или снять с помощью мыши. Для этого в серой области строки, в которой требуется установить точку останова, нужно дважды щелкнуть левой кнопкой мыши.

Для управления точками останова используются следующие команды меню **Отладка** главного меню конфигуратора:

Команда	Пояснение
Точка останова	Устанавливает либо стирает точку останова на той строке, на которой стоит курсор
Точка останова с условием	Устанавливает точку останова и открывает диалог для ввода условия останова – логического выражения. Если условие было введено, то открывается диалог для редактирования условия. Если в данной строке точка останова была установлена, то открывается диалог для ввода условия останова и ставится условие для точки останова с условием. Останов в указанной точке будет выполняться только в том случае, если условие останова истинно
Отключить точку останова	Включает либо отключает действие точки останова. Доступна, если в текущей строке есть точка останова
Убрать все точки останова	Стирает все ранее расставленные точки останова во всех модулях
Отключить все точки останова	Запрещает действия всех ранее расставленных точек останова во всех модулях, не удаляя их
Список точек останова	Просмотр и управление точками управления
Останавливаться по ошибке	При возникновении ошибки отладчик останавливает выполнение и переходит к строке модуля, вызвавшей ошибку

Установленные точки останова запоминаются при закрытии конфигурации. Для просмотра списка точек останова нужно открыть конфигурацию и выбрать пункт меню **Отладка – Список точек останова**.

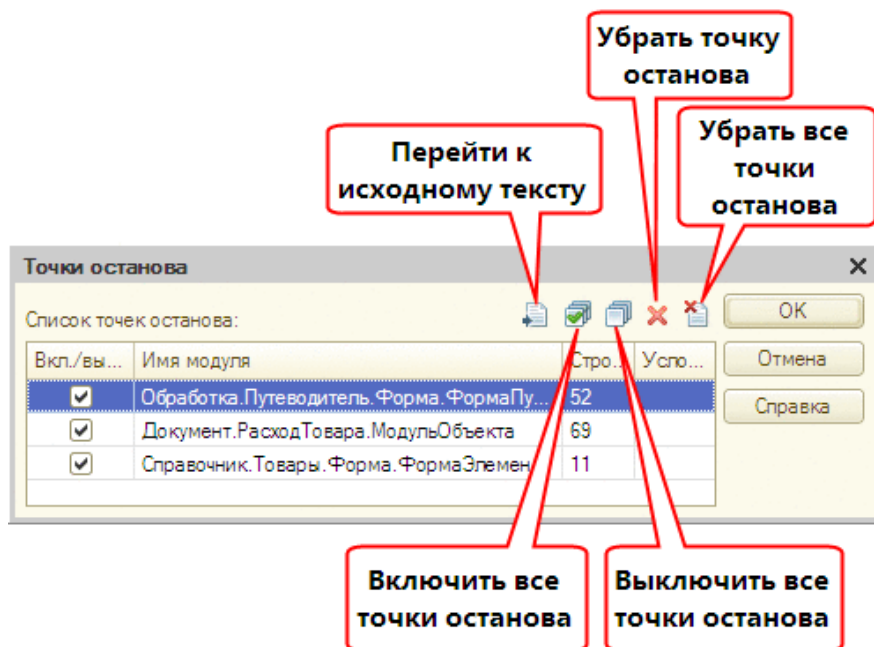


Рис. 439. Список точек останова

С помощью флажков в колонке **Вкл./выкл.** Можно управлять состоянием точек останова. Если флажок установлен, то при отладке на данной точке отладчик выполнит остановку.

Назначение остальных кнопок описано на [рис. 439](#).

28.1.4. Пошаговое выполнение

В режиме пошагового исполнения предмет отладки, выполнив очередную команду, ждет от отладчика инструкций о продолжении работы.

В момент подключения первого из предметов отладки в меню **Отладка** система добавляет пункты, с помощью которых осуществляется управление процессом отладки.

На каждом шаге исполнения модуля существует несколько вариантов продолжения. Для выбора варианта продолжения используются пункты меню **Отладка**.

Команда	Пояснение
Шагнуть в	Если следующим выполняемым оператором модуля является вызов функции или процедуры, начинается ее пошаговое выполнение, иначе отладчик переходит к следующему оператору
Шагнуть через	Если следующим выполняемым оператором модуля является вызов функции или процедуры, она выполняется целиком (не пошагово) и отладчик переходит к следующему оператору
Шагнуть из	Прервать пошаговое выполнение функции или процедуры и остановиться на первом операторе, находящемся после ее вызова
Идти до курсора	Прервать пошаговое выполнение модуля, выполнять все операторы до той строки, на которой стоит курсор
Продолжить	Прервать пошаговое выполнение модуля и продолжить свободное выполнение

При отладке прикладных решений необходимо помнить следующую особенность: если в отлаживаемом программном коде встречается оператор **Выполнить ()**, то:

- не поддерживается пошаговое исполнение программного кода, переданного в качестве параметра оператора (включая вызовы процедур и функций);
- поддерживается прерывание исполнения программного кода с помощью точек останова, которые расположены внутри процедур и функций, вызываемых из программного кода, переданного в качестве параметра оператора **Выполнить ()**.

Если выполняется отладка сразу нескольких предметов отладки, то существует ряд особенностей пошагового выполнения:

- если выполнена остановка одного предмета отладки, при начале исполнения кода останавливаются и другие;

- выполнение команды [Продолжить](#) приводит к продолжению выполнения всех предметов отладки;
- выполнение команды [Шагнуть через](#) приводит к исполнению продвижения на следующую строку во всех предметах отладки;
- выполнение команды [Шагнуть в](#) (если выполняемым оператором модуля является вызов функции или процедуры) приводит к переходу на первый оператор внутри этого вызова, для других предметов отладки всегда выполняется команда [Шагнуть через](#).

Если производится отладка клиент-серверного варианта и код последовательно выполняется на клиенте и на сервере (подключение клиентского и серверного предметов отладки выполнено), то:

- выполнение команды [Шагнуть в](#) (если выполняемым оператором модуля является вызов функции или процедуры, исполняемой на сервере) приводит к переходу на первый оператор внутри этого вызова;
- выполнение команды [Шагнуть из](#) или команды [Шагнуть через](#) для последнего исполняемого оператора (если выполняемым оператором модуля является код функции или процедуры, исполняемый на сервере и вызванной из модуля, выполняемого в клиентском приложении) приводит к переходу на следующий исполняемый оператор внутри этого вызова.

Для выбора текущего предмета отладки выводится специальная панель инструментов [Предметы отладки](#). Панель состоит из единственного поля выбора, в котором показывается текущий предмет отладки. Это поле выбора доступно только тогда, когда управление работой какого-либо из подключенных предметов отладки находится в отладчике (например, после срабатывания точки останова). При этом в список предметов отладки попадут только те предметы, управление исполнением которых сейчас также находится в отладчике, включая текущий предмет отладки.

С помощью табло и диалога [Выражение](#) можно получить значения интересующих выражений (см. [здесь](#)). [Стек вызовов](#) позволяет проследить последовательность вызова процедур и функций (см. [здесь](#)).

Если выполняется пошаговый процесс выполнения, то стек вызова, значения переменных (в табло и в окне [Выражение](#)) показываются для текущего предмета отладки. При смене предмета отладки стек вызова и значения переменных также меняются.

ВНИМАНИЕ! Если выполнено подключение клиентского и серверного предметов отладки и осуществлен переход из клиентской части в серверную, то на клиентских уровнях стека вызова любые вычисления не выполняются. Такие уровни выводятся в окне стека вызовов серым цветом.

Если необходимо продолжить выполнение модуля, то с помощью команды [Отладка – Продолжить отладку](#) нужно разрешить подключенным предметам отладки свободное выполнение модуля (до следующей точки останова). Если для отладки подключено клиентское приложение, то оно активизируется автоматически.

Чтобы прервать процесс отладки в целом (кроме фоновых заданий), нужно снять все точки останова со всех модулей и выполнить команду [Отладка – Продолжить отладку](#), если в данный момент сработала точка останова. Если необходимо прервать отладку и завершить работу подключенных предметов отладки, следует воспользоваться командой [Отладка – Завершить](#). В последнем случае не будут выполнены процедуры [ПередЗавершениемРаботыСистемы\(\)](#) и [ПриЗавершенииРаботыСистемы\(\)](#).

В процессе отладки допускается редактирование текущей конфигурации и сохранение изменений.

ВНИМАНИЕ! Хотя в процессе отладки возможно редактирование отлаживаемого модуля, отладчик не производит компилирование измененного кода – продолжается отладка кода конфигурации базы данных (на момент запуска отладчика или подключения). Для отладки изменений, внесенных в конфигурацию, необходимо выполнить обновление конфигурации базы данных.

Если в режиме 1С:Предприятие устанавливается монопольный режим, то сохранение текущей конфигурации невозможно до тех пор, пока монопольный режим не будет снят.

Если файловый вариант информационной базы находится в состоянии пошагового выполнения кода на встроенном языке или стоит на точке останова и при этом существует открытая транзакция (явная или неявная), то выполнение некоторых операций (например, попытка захвата объектов в хранилище конфигурации) может сопровождаться ошибкой: [Ошибка операции с хранилищем конфигурации](#). ([Ошибка при выполнении сбора информации для хранилища конфигурации](#)) [Конфликт блокировок при выполнении транзакции](#). [Не удалось заблокировать таблицу FILES](#). [Перед повтором операции рекомендуется завершить работу клиентского приложения](#). Для выполнения операции, во время которой произошла ошибка, следует продолжить выполнение кода на встроенном языке до момента завершения транзакции.

Таблицу сочетаний клавиш для работы с отладчиком можно получить в справке при использовании программы.

28.1.5. Отладка внешних обработок (отчетов)

Для отладки модулей внешней обработки (отчета) необходимо открыть файл внешней обработки в конфигураторе,

воспользовавшись пунктом **Файл – Открыть**.

В дальнейшем с модулями внешней обработки (отчета) в отладчике можно работать так же, как и с любым другим модулем.

28.1.6. Управление отладкой

Команда **Отладка – Перезапустить** прекращает выполнение конфигурации и производит повторный ее запуск в режиме 1С:Предприятие. Если конфигурация была модифицирована, то на экран выдается запрос о необходимости провести обновление конфигурации базы данных.

Команда **Отладка – Завершить** прекращает выполнение модуля и заканчивает работу текущего предмета отладки.

Команда **Отладка – Остановить** останавливает выполнение модуля на текущем операторе. Данная команда позволяет начать отладку модуля со следующей исполняемой строки. Эта команда полезна при анализе «зацикливания» модуля.

Команда **Отладка – Остановка по ошибке** открывает диалоговое окно настройки остановки по ошибке.

Положение флажка **Останавливаться по ошибке** определяет, будет ли приостановлена отладка, если в процессе выполнения произошла ошибка. При остановке по ошибке на экран выводится предупреждение: **Ошибка времени выполнения**, и приводится поясняющий текст о месте и причине возникновения ошибки.

Если установлен флажок **Останавливаться только на ошибках, содержащих текст**, то отладка будет остановлена в месте ошибки (если сообщение об ошибке содержит указанную в табличном поле подстроку). Если этот флажок не установлен, то остановка по ошибке будет происходить независимо от текста сообщения об ошибке. При этом указанная подстрока будет учитываться, только если флажок напротив нее установлен.

На месте возникновения ошибки устанавливается текущая точка исполнения текущего предмета отладки, а предмет отладки, в котором произошла ошибка, становится текущим. Просмотр значений выражений и переменных позволяет установить причину возникновения ошибки.

28.1.7. Окно «Выражение»

Во время пошагового выполнения имеется возможность рассчитать выражение с помощью окна **Выражение**. Это окно вызывается на экран выбором пункта **Отладка – Вычислить выражение**.

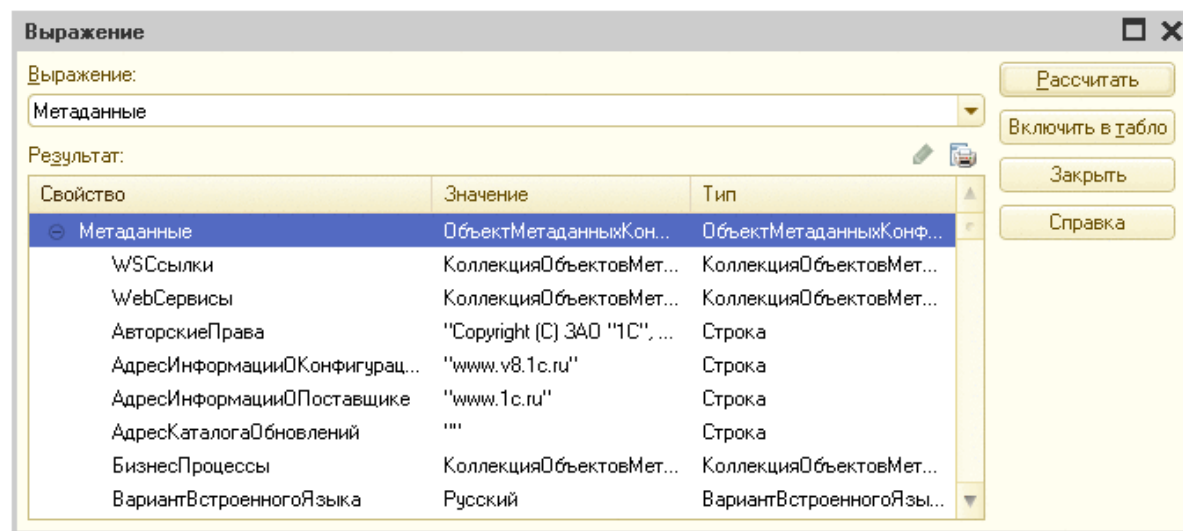


Рис. 440. Вычислить выражение

В поле **Выражение** этого окна необходимо ввести выражение на встроенном языке системы «1С:Предприятие». Выражение можно ввести «вручную» или из списка ранее введенных выражений.

Если при вызове окна **Выражение** курсор в окне модуля находился на каком-либо выражении или выражение было выделено, оно автоматически подставляется в поле **Выражение**.

Для вычисления выражения следует нажать кнопку **Рассчитать**. Результат вычисления выражения будет выдан в поле **Значение**.

Кнопка **Включить в табло** помещает введенное выражение в табло. Это позволит в дальнейшем проследить изменение результата вычисления выражения в процессе отладки модуля. При этом выражение размещается в новой строке табло. Если окна табло на экране нет, то оно открывается.

Если выражение имеет тип **Строка** или представляет собой коллекцию значений или массив, то становится доступной

кнопка панели инструментов [Показать значение](#) в отдельном окне.

Для удобства просмотра длинных строк выводится окно [Просмотр значения](#) выражения. Окно содержит элемент управления типа [Надпись](#), в которое выводится значение строки для просмотра.

Для удобства просмотра коллекции значений или массива выводится окно, содержащее табличное поле, колонки которого соответствуют именам реквизитов, а строки содержат значения. Выводится количество элементов коллекции, а в первой колонке указывается индекс каждого элемента коллекции.

Если, в свою очередь, конкретное значение также является коллекцией значений или массивом, а также строкой, то возможен просмотр этих значений в отдельном окне (как показано на [рис. 441](#)). Для этого следует выбрать нужное значение и в контекстном меню выбрать пункт [Показать значение в отдельном окне](#) или нажать клавишу **F2**.

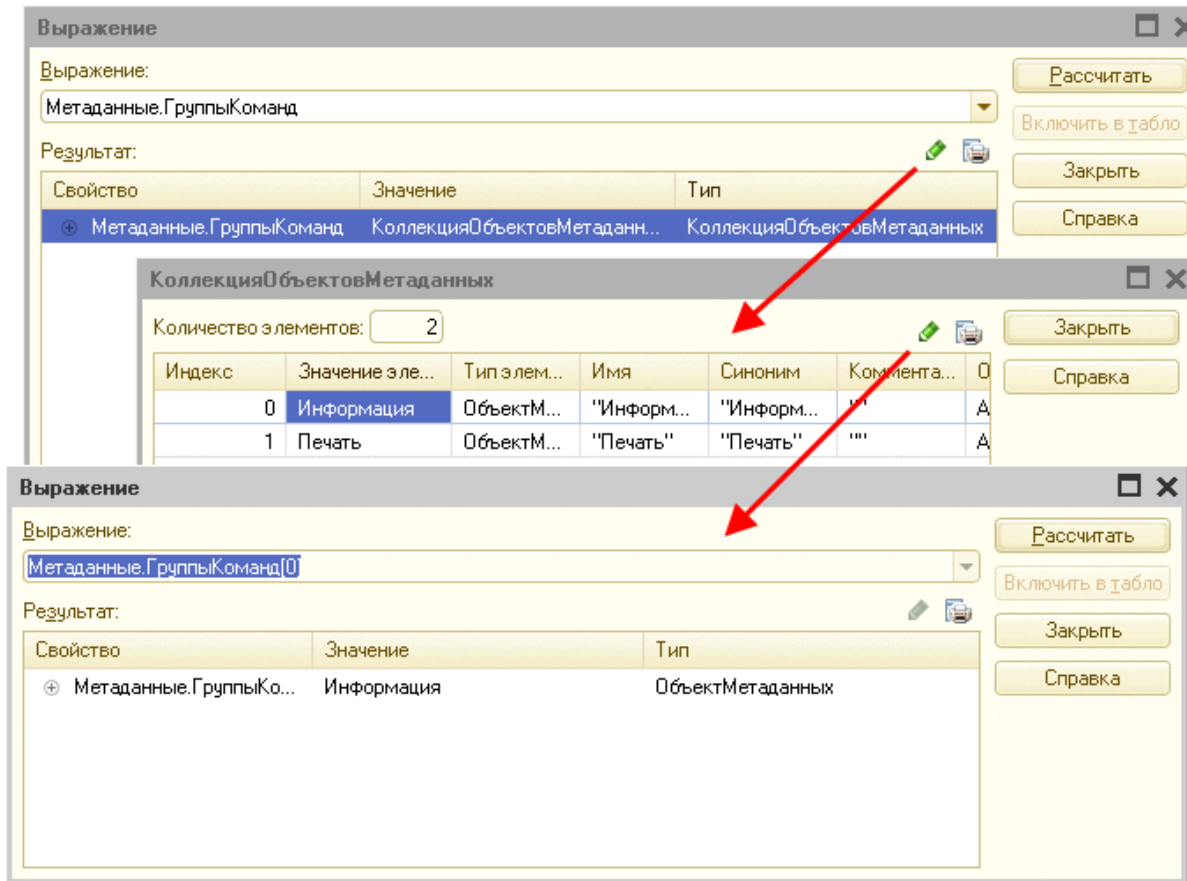


Рис. 441. Отображение значений

Для строки выводится окно, в котором значение строкового выражения выводится в многострочное поле. Это значение можно поместить в буфер обмена.

Содержимое табличного поля результата расчета можно вывести в текстовый или табличный документ по кнопке [Вывести список](#). Если вычисленное значение представлено в виде дерева, то будут выводиться данные только раскрытых строк данного дерева.

Текущее значение выражения можно также посмотреть, подведя указатель мыши к выражению. Текущее значение показывается в виде короткой подсказки рядом с переменной. Просмотр возможен, если значение имеет текстовое представление. Таким же образом можно посмотреть значение свойства. Если свойство представлено в виде `<Объект>.[<Объект>...]<Свойство>` (например, `ЭлементыФормы.КоманднаяПанель.Кнопки.Получение.Доступность`), то просмотр значения возможен при подведении указателя мыши к такому тексту. Значение показывается, если в тексте не используются круглые и квадратные скобки. Также можно посмотреть значение элемента массива, для которого явно указан его индекс (в виде числа или переменной, значение которой определено на момент просмотра). Для этого нужно выделить идентификатор массива и его индекс.

При подведении указателя мыши к отдельному объекту, или выделенному объекту, или группе объектов в тексте вида `<Объект>.[<Объект>...]<Свойство>` показывается его тип. Например, если в тексте `ЭлементыФормы.КоманднаяПанель1.Кнопки.Получение.Доступность` выделить `ЭлементыФормы.КоманднаяПанель1`, то будет показан тип `КоманднаяПанель`.

28.1.8. Табло

Табло – специальное окно, в котором отображаются результаты вычисления переменных и введенных в него формул в процессе выполнения программы в процессе отладки. Вызов табло на экран выполняется выбором пункта меню [Отладка – Табло](#), доступным при отладке.

Выражение	Значение	Тип
⊖ Метаданные.Документы	КоллекцияОбъектовМетадан...	КоллекцияОбъекто...
⊕ ОперацияПоУчетуТоваров	Операция по учету товаров	ОбъектМетаданных
⊕ Оплата	Оплата	ОбъектМетаданных
⊕ ПоступлениеДенег	Поступление денег	ОбъектМетаданных
⊕ ПриходТовара	Приход товара	ОбъектМетаданных
⊕ РасходТовара	Расход товара	ОбъектМетаданных
⊕ РучнаяОперация	Ручная операция	ОбъектМетаданных

Рис. 442. Табло

Табло представляет собой четырехстраничную форму. Каждая ее страница содержит табличное поле для ввода переменных и формул, результаты вычисления которых необходимо контролировать. Формулы могут включать арифметические выражения, выражения с использованием функций встроенного языка системы «1С:Предприятие», а также функции модуля управляемого приложения и общих модулей.

Каждая формула вводится в первую колонку табличного поля и должна находиться на отдельной строке. Результат вычисления формулы выдается в колонке **Значение**. В колонке **Тип** выводится тип выражения. Если формула введена неправильно, то вместо результата появится фраза: **Обнаружены синтаксические ошибки!**

Управление табло и результатами вычислений осуществляется с помощью команд контекстного меню.

Результат расчета может быть скопирован в буфер обмена выбором пункта **Копировать результат** контекстного меню второй колонки.

Для некоторых типов данных (см. предыдущий раздел) возможен просмотр значений в отдельном окне.

Табло также можно вывести в табличный или текстовый документ.

Если в процессе работы исходные данные, используемые в формулах, изменились, то для получения актуальных результатов расчетов необходимо выполнить обновление. Для этого в контекстном меню табло нужно выбрать пункт **Пересчитать** или **Пересчитать все**.

С точки зрения выбора страниц табло работает в двух режимах: в первом страницы выбираются с помощью закладок, располагающихся внизу, во втором – с помощью контекстного меню. Выбор режима производится установкой или снятием флажка в пункте **Закладки** контекстного меню.

Для очистки содержимого строки табло необходимо выбрать строку и нажать клавишу **Del**.

Окно табло может размещаться на экране в различных режимах. После завершения отладки окно автоматически закрывается.

28.1.9. Стек вызовов

Стек вызовов показывает последовательность вызовов процедур и функций, приведшую к строке модуля, которая отлаживается в данный момент.

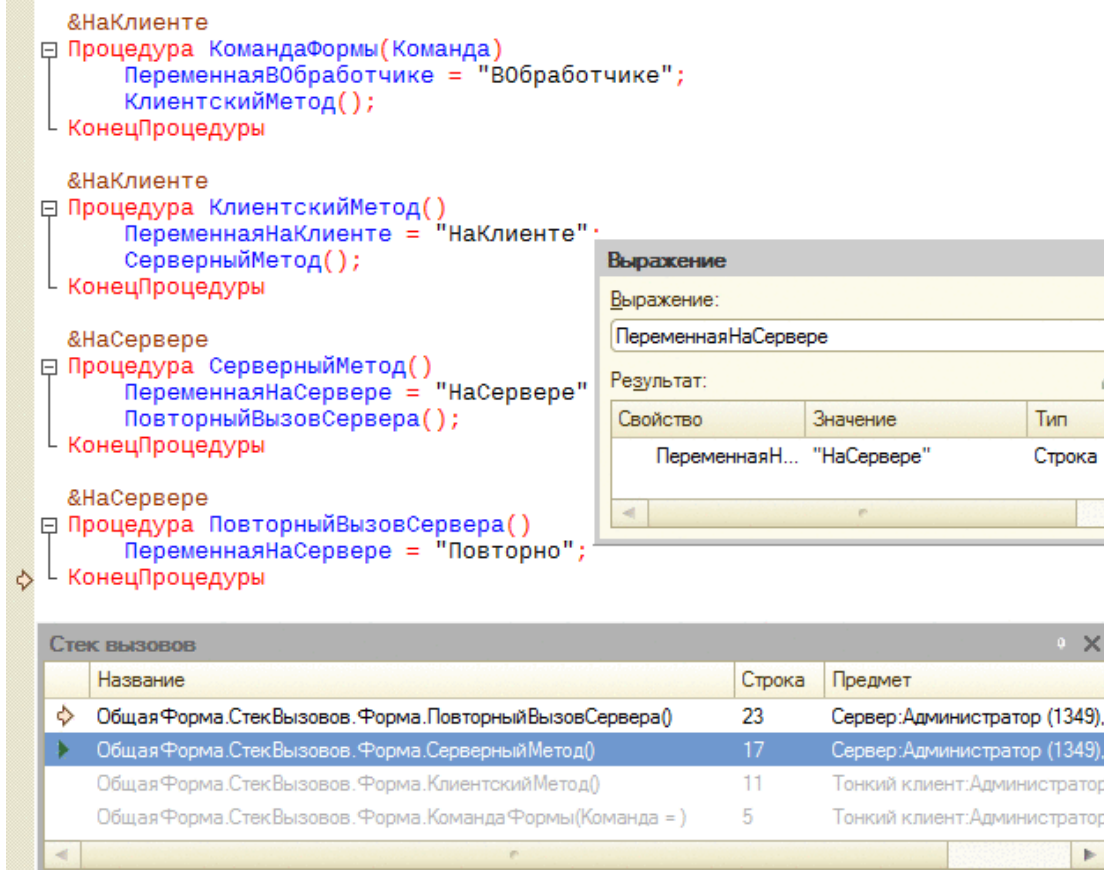


Рис. 443. Стек вызовов

Окно отображает:

- исполняемый метод – колонка **Название**;
- номер строки модуля – колонка **Строка**;
- предмет отладки, в котором расположен модуль, – колонка **Предмет**.

В первой колонке окна могут быть две пиктограммы:

- Желтая стрелка означает вершину стека вызовов.
- Зеленая стрелка (расположенная слева от строки `ОбщаяФорма.СтекВызовов.Форма.СерверныйМетод()`) указывает метод, в контексте которого будет работать вычисление переменных в окне **Выражение** и в окне **Табло**. Если вершина стека вызовов совпадает с текущим контекстом, то зеленая стрелка не отображается на экране. Для смены контекста (а также установки курсора на строку, указанную в колонке **Строка**) необходимо дважды щелкнуть левой кнопкой мыши по нужной строке окна **Стек вызовов**.

В примере, показанном на [рис. 443](#), переменная `ПеременнаяНаСервере` имеет значение `НаСервере`, т. к. отладчик переключен в контекст метода `СерверныйМетод()` (об этом свидетельствует зеленая стрелочка напротив строки `ОбщаяФорма.СтекВызовов.Форма.СерверныйМетод()`).

Если переключить контекст на вершину стека (двойной щелчок по строке с желтой стрелкой), то значение переменной `ПеременнаяНаСервере` станет равно значению `Повторно`, т. к. произойдет переключение контекста.

Некоторые строки в окне **Стек вызовов** отображаются серым цветом (последние две строки на [рис. 443](#)). Это означает, что имеется возможность переключиться на строку модуля, но нельзя вычислять выражения в контексте того метода, куда произошел переход. Таким образом, выделяются те элементы стека, которые «расположены» на клиенте, в то время как вершина стека «расположена» на сервере.

28.1.10. Особенности отладки защищенных модулей

Если при выполнении команды **Шагнуть в строку** модуля содержит вызов процедуры или функции, расположенной в защищенном модуле (см. [здесь](#)), то при первой попытке перехода к выбранной процедуре или функции защищенного модуля отладчик запрашивает пароль доступа.

Если пароль введен правильно, то отладчик открывает защищенный модуль и управление переходит на первую строку вызываемой процедуры или функции. При последующих переходах в защищенный модуль в данном сеансе работы

пароль не запрашивается.

Если отказаться от ввода пароля, то отладчик пропускает защищенный код программы (выполняя его) и переходит на первую строку кода после защищенного модуля. Аналогично выполняется команда [Шагнуть из](#).

В стеке вызовов всегда указываются наименования процедур и функций. Если защищенный модуль еще не открывался, то при первой попытке перехода к выбранной процедуре защищенного модуля отладчик запрашивает пароль доступа. При последующих переходах в защищенный модуль из стека вызовов пароль запрашиваться не будет (в текущем сеансе работы).

28.1.11. Варианты использования

Данный раздел содержит краткое описание типовых сценариев работы с отладчиком.

28.1.11.1. Запуск «1С:Предприятия» в отладочном режиме

Для того чтобы запустить «1С:Предприятие» в отладочном режиме, необходимо выполнить команду [Отладка – Начать отладку](#) или нажать клавишу **F5**. Тип клиента, который будет запущен в этом случае, определяется исходя из настроек конфигуратора (см. [здесь](#)) и основного режима запуска конфигурации (см. [здесь](#)).

28.1.11.2. Подключение к работающему «1С:Предприятию»

Для того чтобы выполнить отладку какого-либо механизма в уже работающем «1С:Предприятии», необходимо выполнить следующие действия:

- Разрешить отладку в исполняемом экземпляре «1С:Предприятия». Для этого нужно в окне настройки параметров ([Главное меню – Сервис – Параметры](#)) установить флажок [Отладка в текущем сеансе разрешена](#).

ПРИМЕЧАНИЕ. Недоступно в веб-клиенте.

- Определить номер сеанса, который будет отлаживаться. Для этого включить отображение команды [Все функции](#) (в окне настройки параметров установить флажок [Отображать команду "Все функции"](#)). Затем открыть список активных пользователей ([Главное меню – Все функции – Стандартные – Активные пользователи](#)) и запомнить значение колонки [Сеанс](#) для выделенного пользователя (текущий пользователь). Для просмотра списка активных пользователей у пользователя должно быть доступно соответствующее право доступа.
- Затем необходимо с помощью конфигуратора открыть информационную базу, которую мы собираемся отлаживать (подробнее о запуске различных режимов «1С:Предприятия» (см. книгу «1С:Предприятие 8.3. Руководство администратора»).
- Далее следует открыть окно подключения к предметам отладки (см. [здесь](#)) и подключить тот предмет отладки, у которого значение колонки [Сеанс](#) равно значению колонки [Сеанс](#) из списка активных пользователей (см. выше).

Теперь в подключенном предмете отладки начинают работать точки останова и становится возможным отлаживать программный код.

28.2. Замер производительности

Замер производительности позволяет оценить скорость работы всей конфигурации или ее части, работающей в рамках любого типа предмета отладки. Измеряется частота использования конкретных участков кода и скорость их выполнения; указывается, какой код исполнялся на сервере, а какой – на клиенте; указываются строки кода, приведшие к вызову сервера. Если имеется несколько способов решения какой-либо задачи, можно реализовать их все, после чего выбрать самый быстрый.

При этом необходимо иметь в виду, что сравнение нужно производить в одинаковых условиях. Например, если во время выполнения задачи одним из сравниваемых способов процессор компьютера был загружен еще какой-либо задачей, это может повлиять на достоверность сравнения. Возможны и другие, менее очевидные причины, по которым условия измерения окажутся различными. Поэтому при сравнении двух способов выполнения задачи, имеющих близкую производительность, желательно делать с каждым из них несколько замеров – для оценки и усреднения случайного разброса.

Для замера производительности нужно выбрать команду [Отладка – Замер производительности](#). При повторном выборе команды замер прекратится и откроется окно с его результатами. Включение и выключение замера производительности действуют на все предметы отладки, которые в настоящий момент подключены к отладчику.

28.2.1. Варианты порядка действий

Если нужно измерить производительность конфигурации, включая участок, выполняемый при старте системы, необходимо сначала выбрать команду [Отладка – Замер производительности](#), а затем запустить систему «1С:Предприятие». Время, прошедшее между стартом замера и началом работы системы, не будет учитываться в результатах замера.

Если участок, выполняемый при старте системы, включать в замер не требуется, необходимо сначала запустить систему «1С:Предприятие», подготовить ее к выполнению требуемого участка, затем перейти в конфигуризатор и включить замер.

Если в замер нужно включить участок, выполняемый при окончании работы системы «1С:Предприятие», то, независимо от того, использовался ли вариант 1 или 2 для начала замера, нужно завершить работу программы, после чего перейти в режим Конфигуратор. В этом случае прекращать замер вручную не нужно. Как только будут подведены итоги замера, его результаты появятся на экране.

Если участок, выполняемый при окончании работы системы «1С:Предприятия», включать в замер не требуется, то для появления результатов замера его нужно закончить. Например, для анализа процедуры проведения какого-либо документа нужно запустить программу, открыть документ, заполнить его, перейти в режим Конфигуратор, включить замер, перейти в режим 1С:Предприятие, провести документ, перейти в режим Конфигуратор и закончить замер.

28.2.2. Результаты замера

Результаты замера – ссылки на конкретные строки модуля с указанием частоты их выполнения и длительности – представляются в виде табличного поля, имеющего следующие колонки:

- **Модуль** – содержит название модуля.
- **Номер строки** – номер строки модуля.
- **Строка** – текст данной строки модуля.
- **Кол.** – количество вызовов данной строки за время замера.
- **Врем.** – суммарное время (сек.) выполнения данной строки за время замера.
- **%(Врем.)** – процент суммарного времени выполнения данной строки к общему времени замера (общее время замера равно сумме всех промежутков времени, в которые выполнялся код конфигурации). При этом за 100 % принимается время выполнения кода на клиенте.

Строчка	Кол.	Вр...	%В...	Клиент	Сервер	Обр. сервером
Движения.Записать();	24	0,358712	1,74			▶
РезультатУслуги = Запрос.Выполнить().В...	24	0,101783	0,38	☒	☒	
КоличествоПроведенных = КоличествоП...	468	0,096925	0,36	☒		
Обработка.Провести(ДатаПроведения, К...	469	0,088036	0,33			▶

Callouts from the image:

- Исполнение на клиенте**: points to the 'Клиент' column.
- Непосредственный вызов сервера**: points to the 'Обр. сервером' column.
- Исполнение на сервере**: points to the 'Сервер' column.
- Вложенный вызов сервера**: points to the 'Обр. сервером' column.

Рис. 444. Пиктограммы замера производительности

- **Клиент** – пиктограммой (см. [рис. 444](#)) отмечаются строки кода, выполняющиеся на клиенте;
- **Сервер** – пиктограммой (см. [рис. 444](#)) отмечаются строки кода, выполняющиеся на сервере;
- **Обр. сервер** – пиктограммами отмечаются строки кода, приводящие к вызову сервера:
 - вызов сервера происходил на уровне платформы, или непосредственно вызывались процедуры или функции, исполняемые на сервере (**Непосредственный вызов сервера** на [рис. 444](#));
 - локальный вызов процедуры или функции, исполняемой на клиенте, внутри которой вызов сервера происходил на уровне платформы, или непосредственно вызывались процедуры или функции, исполняемые на сервере (**Вложенный вызов сервера** на [рис. 444](#)).

В результатах замера производительности время выполнения каждой строки складывается из времени выполнения собственно операторов строки («чистое время») и времени вызова процедуры (функции), если такие в строке есть. С помощью флажка **Для вызовов процедур и функций включать время выполнения** можно выбирать, какое время требуется показывать: полное время (как сумму времени вызова и «чистого времени») или «чистое время» выполнения.

Модуль	Номер...	Строка	Кол.	Вре...	%(Вре...	Кл...	Се...	Обр. с...
Обработка.ПроведениеДокум...	24	Пока Выборка.Следующий(...	2 892	12,315655	45,95			
Обработка.ПроведениеДокум...	35	ПровестиНаСервере(Дата...	469	7,889772	29,43			
Обработка.ПроведениеДокум...	30	+ НСтр("ru = 'Проведено '", ...	469	2,874556	10,72			
Обработка.ПроведениеДокум...	35	Объект.Записать(РежимЗа...	83	1,119984	4,18			
Обработка.ПроведениеДокум...	7	Предупреждение(НСтр("ru ...	1	1,020438	3,81			
Обработка.ПроведениеДокум...	22	Выборка = Документы[Док...	2 810	0,585194	2,18			
Документ.РасходТовара.Моду...	128	Движения.Записать();	24	0,358712	1,34			
Документ.РасходТовара.Моду...	78	РезультатУслуги = Запрос...	24	0,101783	0,38			
Обработка.ПроведениеДокум...	36	КоличествоПроведенных = ...	468	0,096925	0,36			
Обработка.ПроведениеДокум...	52	Обработка.Провести(Дата...	469	0,088036	0,33			
Обработка.ПроведениеДокум...	51	Обработка = РеквизитФор...	469	0,067898	0,25			
Обработка.ПроведениеДокум...	27	Состояние(НСтр("ru = 'Вып...	938	0,052396	0,20			
			469	0,088036	0,33			

Кол. Врем. %(Врем.)

Для вызова процедур и функций включать время выполнения

Клиент Сервер

Рис. 445. Результат замера

Если в строке есть хотя бы один вызов процедуры (функции), то время выполнения включает время выполнения собственно операторов строки и время вызова процедуры (функции).

Если флажок установлен, то время вызова процедуры (функции) учитывается в общем времени выполнения.

Если флажок снят, в результат замера будет включено только время выполнения строк кода, но не время работы процедуры (функции), которая вызывается в данной строке. В этом случае суммарное время выполнения данной строки (в колонке **Врем.**) не будет отражать реального времени, потраченного системой на обработку данной строки. Необходимо иметь в виду, что выполнение вызванной процедуры (функции) может занимать значительное время, которое в данном случае не будет включено в результат («чистое время»).

По умолчанию флажок установлен, а его состояние запоминается между сеансами. При смене состояния изменяются наименования колонок времени.

Если флажок **Клиент** установлен, то будут показаны результаты замера выполнения кода на клиенте.

Если флажок **Сервер** установлен, то будут показаны результаты замера выполнения кода на сервере.

Если флажок **Клиент** и флажок **Сервер** установлены, то будут показаны результаты замера выполнения кода на клиенте и сервере.

Флажки показываются, если выполняется отладка серверной информационной базы. Флажки доступны, если выполняется отладка серверного предмета отладки.

Если открыто несколько окон с результатами замера производительности, то при подведении указателя мыши к колонке результатов замера показывается подсказка, содержащая URL файла, данные которого в этой колонке отображаются.

По двойному щелчку мыши в колонке с результатами замера производительности в редакторе модуля осуществляется переход к соответствующей строке файла с результатами замера производительности.

Кроме специального окна результаты замера можно видеть непосредственно в окне с исходным кодом модуля. Если в отладчике открыто окно с замером, в окнах модулей появляется колонка, показывающая количество вызовов данной строки и процент времени ее работы от общего времени.

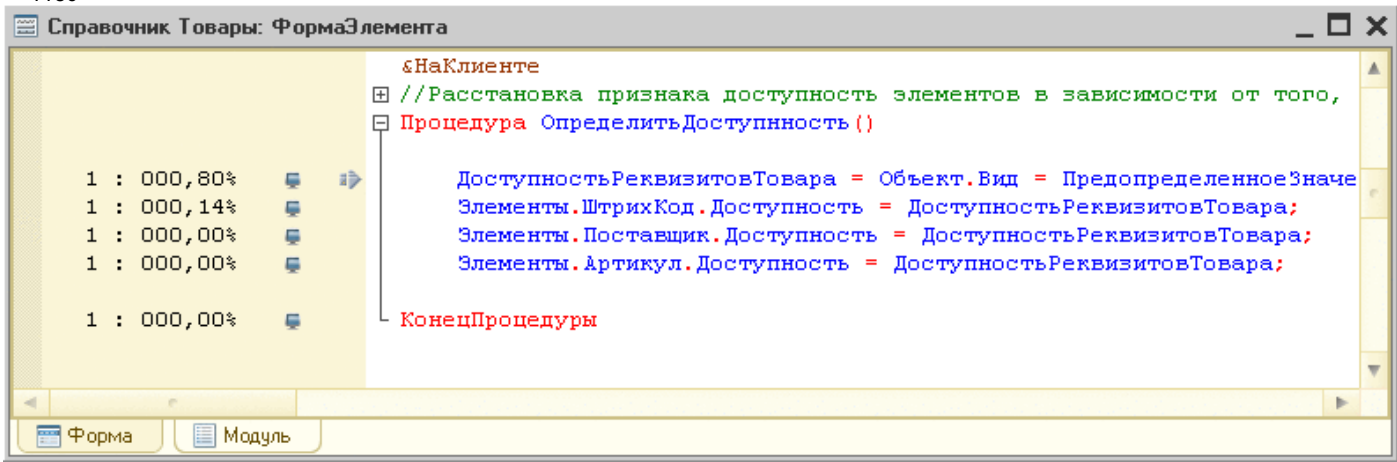


Рис. 446. Результаты замера в тексте модуля

Пиктограммами показывается выполнение кода на клиенте или сервере, а также вызовы сервера (аналогично окну с результатами замера, описание пиктограмм (см. [рис. 444](#)).

Двойным щелчком мыши по строке окна результатов замера можно переключиться на соответствующую строку в окне модуля.

Если открыто несколько замеров одновременно, то в окнах с текстами модулей появится соответствующее количество колонок.

Заккрытие окна результатов замера убирает из модулей колонки с количеством вызовов и процентом времени работы.

28.2.3. Сортировка результатов замера

Результаты замера могут быть отсортированы по любой колонке отчета.

Сортировка осуществляется щелчком мышью на заголовке одной из колонок. Щелчок на заголовках **Модуль** или **Строка** обеспечивает сортировку по номерам строк, **Кол.** – по количеству вызовов строки, **Врем.** Или **%(Врем.)** – по времени работы. Если выполнялся замер производительности выполнения кода на клиенте и на сервере, то сортировка допускается также и по этим колонкам.

28.2.4. Выборочное суммирование результатов замера

Для анализа результатов замера могут оказаться полезными суммарные характеристики. Если в окне результатов отметить несколько строк, их суммарные характеристики – суммы количества вызовов, времени работы в секундах и процентах – отображаются в нижней части окна.

Флажок **Для вызовов процедур и функций включать время выполнения** служит для выбора одного из двух методов приблизительного отслеживания уровней вложенности при суммировании. Если в данном модуле имеется и строка, вызывающая некоторую процедуру, и строки текста самой процедуры, то, конечно, не следует пометить и то, и другое: это приведет к повторному учету в сумме одного и того же времени выполнения. Если все же приходится пометить их (например, слишком много усилий пришлось бы потратить на отслеживание), то можно снять флажок, и повторного учета не будет. С другой стороны, если все вызываемые процедуры – внешние по отношению к модулю, флажок лучше установить. Тогда в общее время выполнения будет включено время отработки этих процедур, что лучше отражает реальное время работы.

28.2.5. Особенности замера производительности при работе клиент-серверной базы

При замере производительности результаты замера для кода встроенного языка, выполняемого для одного и того же номера соединения на клиенте и на сервере, будут объединяться в один общий замер производительности. При этом указывается, какие строки кода выполнялись на клиенте, а какие – на сервере, из каких строк кода осуществлялся вызов процедуры или функции, исполняемых на сервере, а также системные вызовы (самой платформы, например, оператор **Выполнить** для запроса, оператор записи объекта базы данных) на сервере.

При просмотре результатов можно указать, что показывать в замере: замер для клиента, замер для сервера, замер для клиента и сервера вместе. Выбор режима устраивается с помощью флажков в правом нижнем углу окна с результатами замера производительности.

Для серверных вызовов учитывается время выполнения серверного вызова на самом сервере. Можно рассматривать происходящее по аналогии со временем исполнения вложенных вызовов процедур и функций для клиентского приложения.

28.2.6. Сохранение результатов

Результат замера можно сохранить в файл с помощью команд [Файл – Сохранить](#) и [Файл – Сохранить как](#). В стандартном диалоге сохранения нужно выбрать каталог и указать имя файла. Файл результатов имеет расширение *.pff.

Открыть файл с замером можно командой [Файл – Открыть файл](#). Для отбора файлов с замерами следует использовать фильтр [Замер производительности \(*.pff\)](#).

28.3. Механизм имитации задержек при вызове сервера

Цель механизма – имитация работы прикладного решения в условиях существенных временных задержек, возникающих при взаимодействии с сервером.

ПРИМЕЧАНИЕ. Включить механизм можно только в тонком клиенте и в режиме управляемого приложения толстого клиента.

Механизм обеспечивает имитацию временных задержек, возникающих в следующих случаях:

- задержка при вызове сервера: указывается в секундах на каждый вызов сервера;
- задержка при отправке данных на сервер: указывается в секундах в расчете на каждые 1 кбайт (1024 байта) данных, отправляемых на сервер.
- задержка при получении данных с сервера: указывается в секундах в расчете на каждые 1 кбайт (1024 байта), принимаемых с сервера.

Включение механизма можно обеспечить с помощью соответствующей настройки параметров конфигулятора (см. [здесь](#)), а также с помощью ключа командной строки `/SimulateServerCallDelay [-CallXXXX] [-SendYYYY] [-ReceiveZZZZ]`, где:

- **Call** – параметр, указывающий величину задержки при вызове сервера в секундах, если не указан, то 1 секунда.
- **Send** – параметр, указывающий величину задержки в секундах в расчете на каждые 1 кбайт данных, отправляемых на сервер. Если не указан, то 1 секунда.
- **Receive** – параметр, указывающий величину задержки в секундах в расчете на каждые 1 кбайт данных, принятых с сервера. Если не указан, то 1 секунда.

Пример:

```
/SimulateServerCallDelay -Call2.1 -Send1.3 -Receive1.2
```

[Копировать в буфер обмена](#)

ПРИМЕЧАНИЕ. Максимальное значение устанавливаемой задержки равно 9,99 секунды. Значение, равное 0, означает, что данная задержка не применяется.

Механизм имитации задержек при вызове сервера также может быть включен в режиме 1С:Предприятие. Это можно сделать с помощью диалога [Сервис – Параметры](#). Настройка механизма имитации задержек при вызове сервера, сделанная в режиме «1С:Предприятие», действует только на протяжении текущего сеанса и не сохраняется.

28.4. Отображение вызовов сервера

Данный механизм предназначен для отладки клиент-серверной системы и оценки объема данных, передаваемых между клиентом и сервером.

Режим можно включить с помощью соответствующей настройки параметров конфигулятора (см. [здесь](#)) или ключа командной строки `/DisplayPerformance`. Отображение показателей производительности также можно включить и в режиме 1С:Предприятие с помощью флажка [Отображать показатели производительности](#) диалога [Сервис – Параметры](#). По умолчанию механизм выключен.

Если режим включен, то «1С:Предприятие» будет отображать показатели производительности в панели показателей производительности в нижней части основного окна приложения.

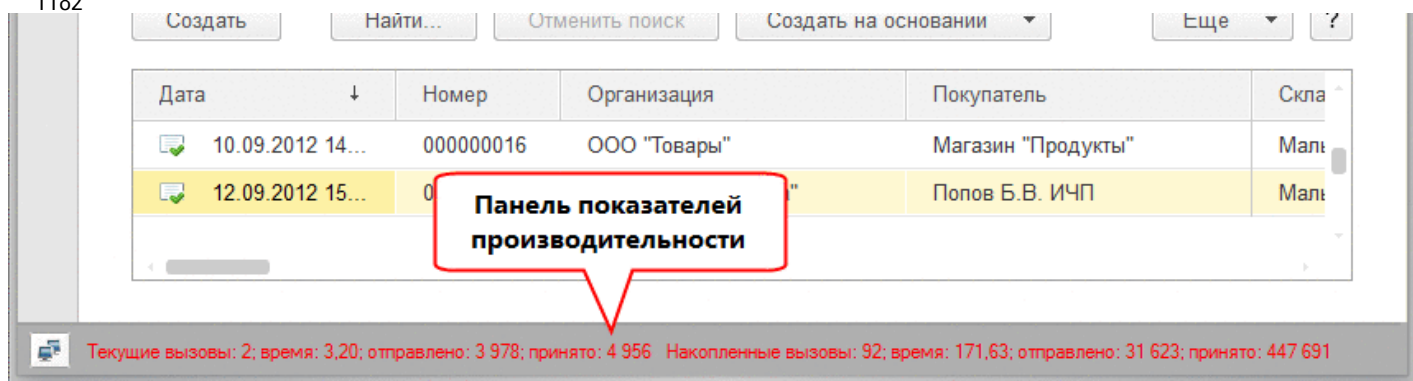


Рис. 447. Панель показателей производительности

Если включен режим имитации задержек при вызове сервера, то картинка в окне, показывающем вызовы сервера, будет иметь другой вид:

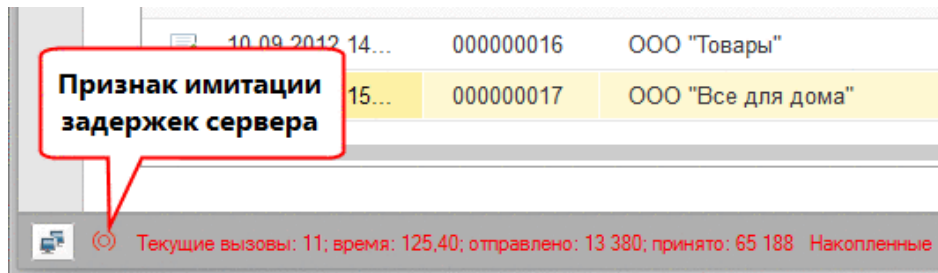


Рис. 448. Вызовы сервера в режиме имитации задержек при вызове сервера

Если прикладное решение находится в состоянии, когда основное окно приложения еще не открыто или уже закрыто, но прикладной код работает, то окно показателей производительности имеет вид:

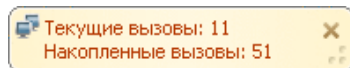


Рис. 449. Вызовы сервера

Если при этом включен режим имитации задержек сервера, то окно показателей производительности будет иметь такой вид:

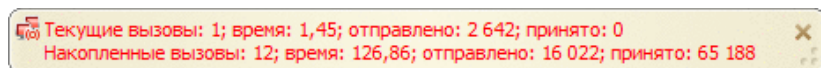


Рис. 450. Имитация задержек при вызове сервера

При изменении данных, отображаемых в окне вызовов сервера, они выводятся красным цветом.

По умолчанию окно состоит из двух счетчиков: [Текущие вызовы](#) и [Накопленные вызовы](#):

- [Текущие вызовы](#) – показывает состояние вызовов сервера с некоторого момента времени: с начала работы механизма или с последнего пользовательского действия, после которого 0,2 сек. Не было других пользовательских действий.
- [Накопленные вызовы](#) – показывает состояние вызовов сервера с начала работы механизма или с последнего его принудительного сброса через команду контекстного меню.

Настройка показателей производительности выполняется с помощью диалога [Настройка показателей производительности](#). Доступ к этому диалогу можно получить, нажав кнопку в панели показателей производительности и выбрав в меню команду [Настройка](#).

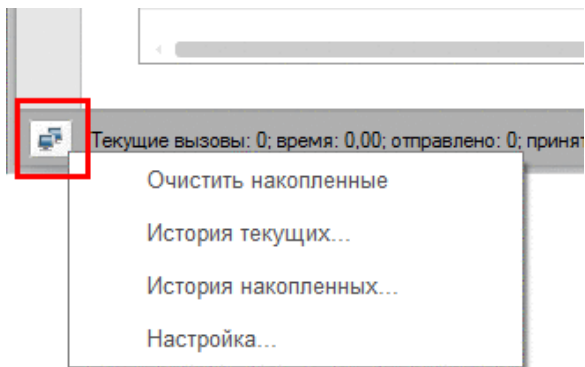


Рис. 451. Меню настройки

В диалоге настройки можно указать следующие параметры:

- группа **Периодичность** – указывает, за какой период требуется отображать значения счетчиков;
- группа **Показатели** – указывает, какие показатели в счетчиках нужно отображать.

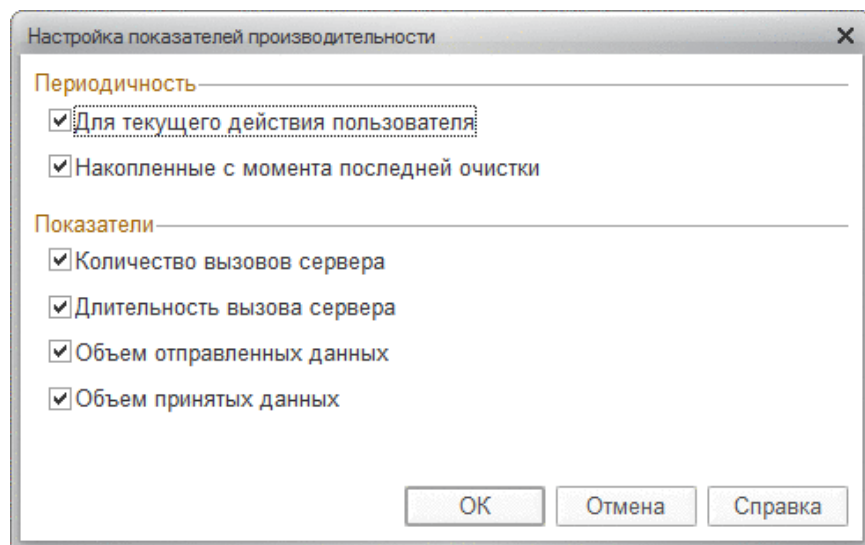


Рис. 452. Настройка показателей производительности

Также контекстное меню окна содержит следующие команды:

- **Очистить накопленные** – очищает счетчик накопленных вызовов;
- **История текущих...** – открывает окно с историей счетчика текущих вызовов;
- **История накопленных...** – открывает окно с историей счетчика накопленных вызовов.

Настройки являются общими для всех пользователей и всех информационных баз на данном компьютере. Настройка отображения окна вызовов сервера, сделанная в режиме 1С:Предприятие, действует только на протяжении текущего сеанса и не сохраняется.

Необходимо учитывать следующие особенности механизма:

- при работе в файловом варианте (при прямом соединении с информационной базой) объем данных отображается без сжатия, так как сжатие в этом случае не производится;
- при работе в клиент-серверном варианте или в файловом варианте с подключением через веб-сервер, при использовании режима сжатия трафика **SDC**, значения показателей могут зависеть от последовательности действий, так как сжатие выполняется с использованием предыдущих передаваемых данных.

В веб-клиенте данный механизм имеет ряд особенностей:

- Учитываются только синхронные запросы.
- Объем передаваемых данных отображается в символах, а не байтах. Результаты замеров можно использовать для оценочного сравнения различных вариантов вызова в веб-клиенте и не рекомендуется использовать для сравнения объема передаваемых данных между различными видами клиентов.

- Показатели производительности не отображаются до момента отображения основного окна приложения, при этом значения показателей накапливаются. В результате, после отображения основного окна приложения, можно увидеть интересующие результаты запуска веб-клиента.

- Окно с показателями имеет больший размер, чем на других клиентах, и не отображается поверх других окон.

28.5. Автоматизированное тестирование прикладных решений

28.5.1. Общая информация

Автоматизированное тестирование представляет собой имитацию интерактивных действий пользователя и проверку результатов этих действий.

ВНИМАНИЕ! Автоматизированное тестирование поддерживается только для управляемого приложения.

Сценарий автоматизированного тестирования представляет собой программу на встроенном языке системы «1С:Предприятие». В сценарии описывается последовательность выполняемых действий и проверка результата их (действий) выполнения. Для этого во встроенном языке существуют специальные объекты, позволяющие имитировать интерактивные действия в тестируемом приложении и предоставляющие доступ к логической модели интерфейса клиентского приложения и элементов управляемой формы.

В процессе тестирования участвуют две системы: **менеджер тестирования** и **клиент тестирования**. Менеджер тестирования обеспечивает:

- Установку соединения с клиентом тестирования для выполнения сценария тестирования (тестового соединения);
- Выполнение собственно сценария тестирования;
- Оценку результатов тестирования (при необходимости).

Клиент тестирования обеспечивает:

- Выполнение команд сценария тестирования, передаваемых с менеджера тестирования;
- Передачу менеджеру тестирования данных, требуемых для оценки результата тестирования.

Менеджером тестирования может выступать толстый или тонкий клиент, клиентом тестирования может быть толстый клиент, тонкий клиент или веб-клиент. Один менеджер тестирования может быть одновременно подключен к нескольким клиентам тестирования, в тоже время один клиент тестирования может быть подключен только к одному менеджеру тестирования. Для взаимодействия менеджера тестирования и клиента тестирования используется прямое TCP/IP соединение через определенный порт (который должен быть «открыт»). Для взаимодействия менеджера тестирования и клиента тестирования, который является веб-клиентом, используется сервер «1С:Предприятия» (для клиент-серверного варианта работы тестируемого приложения) или веб-сервер (для файлового варианта). Для проведения тестирования не требуется вносить какие-либо изменения в тестируемое приложение.

Сценарий тестирования может располагаться как в тестируемом прикладном решении, так и в отдельном прикладном решении. При этом схема взаимодействия менеджера тестирования и клиента тестирования не изменяется.

Следует понимать, что менеджер тестирования (и исполняющийся сценарий тестирования) не обладает непосредственным доступом к тестируемому приложению. Т.е. сценарию тестирования доступны все возможности приложения, где исполняется сам сценарий (менеджер тестирования) и доступ к логической модели интерфейса и форм тестируемого приложения.

Запуск клиентского приложения в режиме клиента тестирования не влияет на быстродействие и другие параметры клиентского приложения, однако позволяет любому менеджеру тестирования подключиться к нему, зная имя (или IP-адрес) компьютера и порт, с помощью которого происходит взаимодействие.

Также система предоставляет возможность записать интерактивные действия пользователя в файл формата **.xml**. В дальнейшем этот файл можно использовать для проведения автоматизированного тестирования. Для выполнения такой записи предназначен особый режим запуска тестируемого приложения. Управление записью интерактивных действий также доступно из встроенного языка.

28.5.2. Запуск системы тестирования

Для выполнения сценария тестирования должны быть запущены и менеджер, и клиент тестирования. Порядок запуска произволен. Важно, чтобы к моменту начала исполнения сценария тестирования клиент тестирования был запущен и доступен по адресу, используемому для подключения менеджера тестирования.

Запуск возможен как из командной строки (этот способ будет рассмотрен ниже) и так и с помощью конфигулятора. Приложение, запускаемое из конфигулятора, может выполнять одну из ролей (менеджер или клиент тестирования) в

зависимости от настроек, указанных в диалоге настроек конфигулятора, подробное описание которых см. [здесь](#).

28.5.2.1. Запуск клиента тестирования

Для запуска клиента тестирования предназначен ключ командной строки запуска клиентского приложения `TestClient`. Также можно указать номер TCP порта, с помощью которого будут взаимодействовать менеджер и клиент тестирования. Для этого служит параметр `TPort` ключа `TestClient`. Если значение для этого параметра не указано, используется значение `1538`. Указывать этот параметр нужно в том случае, если на одном компьютере выполняется несколько клиентов тестирования.

Таким образом, строка запуска клиентского приложения, которое должно выступать в качестве клиента тестирования, выглядит следующим образом:

```
lcv8c ENTERPRISE /IBName "Test Application" /TestClient -Tport 1843
```

[Копировать в буфер обмена](#)

В данном примере тонкий клиент (`lcv8c`) будет выступать клиентом тестирования для информационной базы `Test Application` и будет использовать TCP-порт `1843` для взаимодействия с менеджером тестирования.

Таким образом, для подключения к клиенту тестирования, являющемуся тонким или толстым клиентом, необходимо знать два параметра: IP-адрес (или имя компьютера) на котором запущен клиент тестирования, и номер TCP-порта, с помощью которого будет выполняться взаимодействие.

В случае запуска веб-клиента ситуация немного сложнее. Кроме указания параметра `TestClient`, необходимо указать еще один параметр командной строки запуска: `TestClientID<Идентификатор>`, который содержит уникальный идентификатор тестируемого клиентского приложения. При этом для однозначного указания клиента тестирования, в случае веб-клиента, необходимо три параметра:

- Адрес веб-сервера, на котором развернуто тестируемое приложение. Адрес информационной базы знать не обязательно!
- Номер TCP-порта, через который веб-сервер будет передавать данные между менеджером и клиентом тестирования.
- Идентификатор конкретного экземпляра тестируемого приложения, исполняющегося в веб-клиенте.

Т.е. один веб-сервер может обслуживать несколько информационных баз и несколько соединений с одной информационной базой (в том числе и с целью выполнения тестирования), необходимо указать веб-серверу, какие TCP-порты использовать для взаимодействия с менеджером тестирования. Это делается с помощью файла `testcfg.xml`, который располагается в каталоге настроек расширения веб-сервера. Подробнее о данном файле можно прочитать в книге «1С:Предприятие 8.3. Руководство администратора».

28.5.2.2. Запуск менеджера тестирования

Для того чтобы можно было запустить сценарий тестирования, следует запустить клиентское приложение в режиме менеджера тестирования. Для этого необходимо использовать ключ командной строки запуска клиентского приложения `TestManager`. Веб-клиент не может выступать в качестве менеджера тестирования.

Таким образом, строка запуска клиентского приложения, которое должно выступать в качестве клиента тестирования, выглядит следующим образом:

```
lcv8c ENTERPRISE /IBName "Test Manager" /TestManager
```

[Копировать в буфер обмена](#)

28.5.2.3. Запуск в режиме записи журнала интерактивных действий

Запуск клиентского приложения в режиме, позволяющем выполнять запись интерактивных действий, осуществляется либо с помощью ключа командной строки `UILogRecorder` либо с помощью специальной команды меню конфигулятора `Сервис – Запустить для записи журнала действий пользователя`.

Также можно указать номер TCP порта, с помощью которого менеджер тестирования сможет управлять записью журнала интерактивных действий. Для этого служит параметр `TPort` ключа `UILogRecorder`. Если значение для этого параметра не указано, используется значение `1538`. Также можно указать системе, куда записывать файл журнала. Для этого служит параметр `File` ключа `UILogRecorder`. Если этот параметр не указан, то после окончания записи клиентское приложение откроет окно текстового документа, в котором будет содержаться записанный журнал.

Таким образом, строка запуска клиентского приложения, которое должно позволять формировать журнал интерактивных действий, выглядит следующим образом:

```
lcv8c ENTERPRISE /IBName "Test Manager" /UILogRecorder
```

[Копировать в буфер обмена](#)

При таком запуске в правом верхнем углу основного окна клиентского приложения добавляется специальная командная панель, управляющая записью журнала.

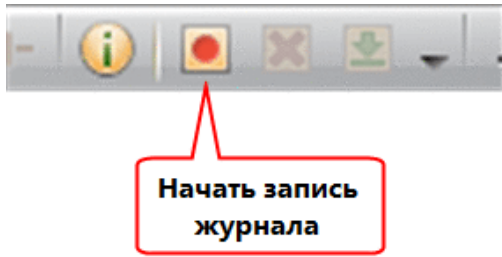


Рис. 453. Командная панель тестирования

Если нажать на единственную доступную кнопку на этой панели, то начнется запись выполняемых действий пользователя. При этом панель сменит свой вид (станут доступны все кнопки).

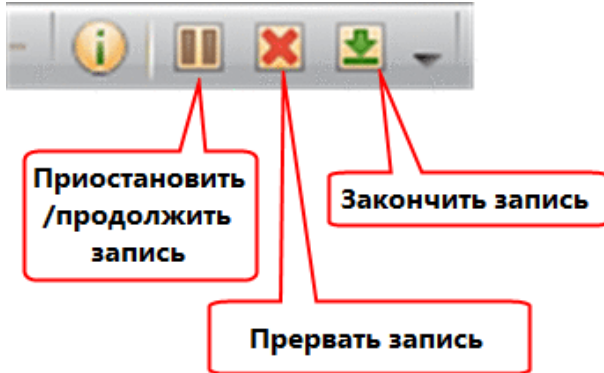


Рис. 454. Командная панель тестирования во время записи

С помощью командной панели можно завершить запись интерактивных действий, при этом будет выполнена запись списка действий в файл (если указан параметр `File`) или открыто окно текстового документа со списком действий. Также имеется возможность прервать запись журнала и приостановить и продолжить запись журнала.

28.5.3. Описание возможностей

С точки зрения сценария тестирования, тестируемое приложение представляет собой набор объектов, позволяющий имитировать интерактивные действия в тестируемом приложении и предоставляющий доступ к логической модели интерфейса клиентского приложения и элементов управляемой формы:

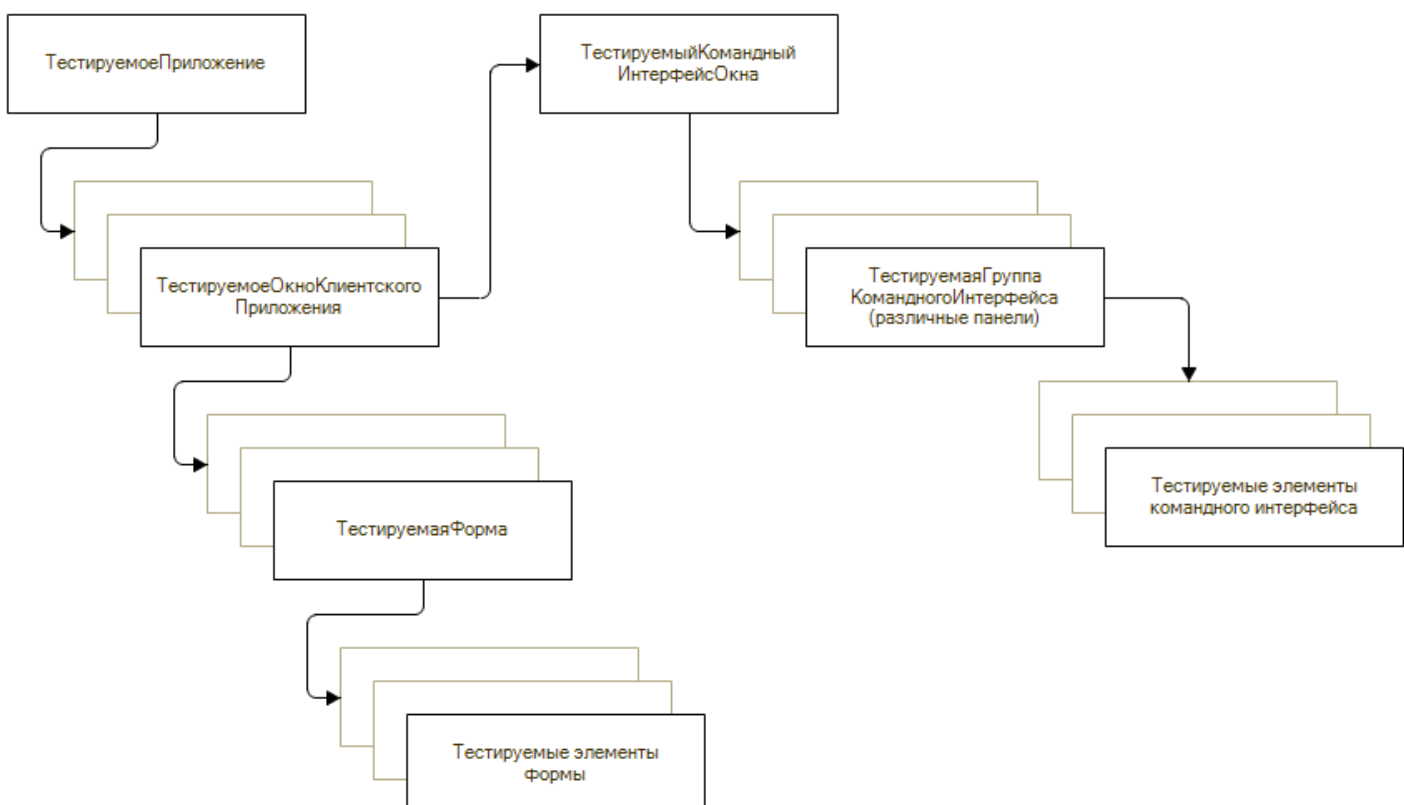


Рис. 455. Схема тестируемого приложения

Далее будут более подробно рассмотрены представленные объекты.

ТестируемоеПриложение

Предоставляет следующие возможности:

- Установить соединение с тестируемым приложением;
- Разорвать установленное соединение с тестируемым приложением;
- Получить активное окно;
- Получить список окон тестируемого приложения;
- Получить текущую информацию об ошибке;
- Получить начальную страницу тестируемого приложения;
- Найти объект с указанными параметрами в иерархии подчинённых объектов;
- Ожидать отображения объекта с заданными параметрами на экране;
- Получать и очищать накопленные показатели производительности тестируемого приложения;
- Получать командный интерфейс тестируемого приложения;
- Управлять ведением журнала интерактивных действий пользователя в тестируемом приложении.

ТестируемоеОкноКлиентскогоПриложения

Описывает окно клиентского приложения и предоставляет методы для управления окном:

- Получение коллекции подчиненных объектов окна;
- Поиск объекта с указанными параметрами в иерархии подчинённых объектов окна клиентского приложения;
- Определить, что данное тестируемое окно описывает начальную страницу (рабочий стол);
- Получить командный интерфейс тестируемого клиентского приложения;
- Работать с сообщениями пользователю;
- Закрыть тестируемую форму;
- Выполнение команды по навигационной ссылке.

ТестируемыйКомандныйИнтерфейсОкна

Описывает командный интерфейс тестируемой формы прикладного решения. Этому объекту подчинены несколько объектов *ТестируемаяГруппаКомандногоИнтерфейса*:

- Панель разделов;
- Панель навигации (кроме интерфейса *Такси*);
- Панель действий (кроме интерфейса *Такси*);
- Панель функций текущего раздела;
- Панель открытых;
- Панель инструментов;
- Панель избранного;
- Панель истории;
- Меню функций.

Для вспомогательных окон в командном интерфейсе формы присутствует только панель навигации формы. При открытии форм в закладках, объект `ТестируемыйКомандныйИнтерфейсОкна`, относящийся к главному окну приложения, не дает возможности получить информацию о подчинённых группах и элементах командного интерфейса.

ТестируемаяГруппаКомандногоИнтерфейса

Данный объект описывает группу командного интерфейса и позволяет получать коллекцию подчиненных объектов группы и выполнять поиск объекта с указанными параметрами в иерархии подчиненных элементов.

`ТестируемаяГруппаКомандногоИнтерфейса` может состоять из элементов двух типов:

- `ТестируемаяГруппаКомандногоИнтерфейса`,
- `ТестируемаяКнопкаКомандногоИнтерфейса`.

ТестируемаяКнопкаКомандногоИнтерфейса

Описывает команду командного интерфейса и предоставляет возможность имитировать ее нажатие.

ТестируемаяФорма

Описывает форму тестируемого приложения и предоставляет методы для управления формой:

- Получение коллекции подчиненных объектов формы;
- Поиск объекта с указанными параметрами в иерархии подчинённых объектов формы;
- Получение текущего активного элемента формы;
- Получение кнопки по умолчанию;
- Получение признака модифицированности;
- Получение командной панели формы;
- Изменить текущий элемент формы на следующий/предыдущий в соответствии с порядком обхода элементов формы (список элементов в редакторе управляемой формы).

ТестируемоеПолеФормы

Описывает поле ввода формы и предоставляет методы для управления этим элементом формы:

- Получение коллекции подчиненных объектов поля формы;
- Поиск объекта с указанными параметрами в иерархии подчинённых объектов поля формы;
- Определение вида элемента формы;
- Позволяет имитировать интерактивные действия для следующих видов поля управляемой формы:
 - Поле ввода;
 - Поле надписи;
 - Поле картинки;
 - Поле переключателя;
 - Поле флажка;
 - Поле индикатора;
 - Поле полосы регулирования;
 - Поле календаря;
 - Поле табличного документа.
- Не позволяет имитировать интерактивные действия для следующих видов поля управляемой формы:

- Поле текстового документа;
- Поле HTML-документа;
- Поле географической схемы;
- Поле форматированного документа;
- Поля диаграммы (любой);
- Поля графической схемы.
- Если объект связан с полем ввода, то предоставляются следующие возможности:
 - Получить текст, редактируемый в поле ввода;
 - Имитировать нажатие кнопки [Очистить](#);
 - Имитировать нажатие кнопки [Выбрать](#);
 - Имитировать нажатие кнопки [Открыть](#);
 - Имитировать открытие списка выбора;
 - Имитировать выбор строки из списка выбора;
 - Имитировать создание нового элемента из списка выбора;
 - Отменять редактирование строки в поле ввода;
 - Имитировать нажатие кнопок регулирования;
 - Для выпадающего списка:
 - Открывать список;
 - Ожидать завершения формирования списка;
 - Закрывать список;
 - Выполнять выбор из выпадающего списка.
- Если объект связан с полем вида флажок, то предоставляются следующие возможности:
 - Имитировать изменение состояния флажка.
- Если объект связан с полем вида переключатель, то предоставляются следующие возможности:
 - Имитировать выбор значения переключателя.
- Если объект связан с полем вида календарь, то предоставляются следующие возможности:
 - Имитировать установку выбранной даты;
 - Имитировать выбор даты;
 - Имитировать операции перехода вперед или назад (на месяц и год).
- Если объект связан с полем вида табличный документ, то предоставляются следующие возможности:
 - Получение адреса текущей области;
 - Получение текста текущей области;
 - Имитация начала редактирования текущей области;
 - Имитация окончания редактирования текущей области;
 - Получение текущего режима редактирования табличного документа.

Описывает таблицу формы и предоставляет методы для управления этим элементом формы:

- Получение активного элемента таблицы;
- Получение текста ячейки таблицы;
- Выполнять перемещение по таблице как по строкам, так и по колонкам;
- Выбрать строку;
- Добавить или удалить строку;
- Получение текущего режима редактирования таблицы;
- Изменять пометку удаления у строки;
- Выделять и получать выделенные строки;
- Изменять порядок сортировки таблицы;
- Выполнять переход по уровням иерархических списков;
- Свернуть или развернуть узел для иерархических списков;
- Проверить состояние узла иерархического списка (свернут или развернут) и возможность развертывания текущей строки иерархического списка.

ТестируемаяКнопкаФормы

Описывает кнопку формы и предоставляет возможность имитировать ее нажатие.

ТестируемаяГруппаФормы

Описывает группу элементов формы и предоставляет методы для управления ей:

- Развернуть группу;
- Свернуть группу;
- Получить текущую страницу группы.

ТестируемаяДекорацияФормы

Описывает декорацию формы и предоставляет возможность имитации нажатия гиперссылки или картинки.

28.5.4. Пример сценария тестирования

Ниже приведен простой пример, который создает новый элемент справочника **Товары** в некоторой конфигурации. Для основной формы элемента важно наличие поля с заголовком **Наименование** и кнопки с заголовком **Записать и закрыть**.

[Копировать в буфер обмена](#)

```
// Подключимся к тестируемому приложению
ТестируемоеПриложение = Новый ТестируемоеПриложение("localhost");
ТестируемоеПриложение.УстановитьСоединение();
// Найдем главное окно
ГлавноеОкноТестируемого = ТестируемоеПриложение.НайтиОбъект(Тип("ТестируемоеОкноКлиентскогоПриложения"));
ГлавноеОкноТестируемого.Активизировать();
// Выполним команду создания элемента справочника товаров
ГлавноеОкноТестируемого.ВыполнитьКоманду("elcib/command/Справочник.Товары.Создать");
ТестируемоеПриложение.ОжидатьОтображениеОбъекта(Тип("ТестируемаяФорма"), "Товар*");
ТестируемаяФорма = ТестируемоеПриложение.НайтиОбъект(Тип("ТестируемаяФорма"), "Товар*");
ТестируемаяФорма.Активизировать();
// Зададим наименование для нового товара
ЭлементФормы = ТестируемаяФорма.НайтиОбъект(Тип("ТестируемоеПолеФормы"), "Наименование");
ЭлементФормы.Активизировать();
ЭлементФормы.ВвестиТекст("Новый товар");
// Запишем элемент
ЭлементФормы = ТестируемаяФорма.НайтиОбъект(Тип("ТестируемаяКнопкаФормы"), "Записать и закрыть");
ЭлементФормы.Нажать();
```

28.5.5. Сервисные возможности

28.5.5.1. Запись журнала действий пользователя

Клиентское приложение можно запустить в режиме, позволяющем выполнять запись интерактивных действий. В дальнейшем этот журнал можно записать в файл. Подробное описание данного режима см. [здесь](#).

28.5.5.2. Преобразование журнала действий пользователя в сценарий автоматизированного тестирования

Записанный журнал действий пользователя можно преобразовать в текст на встроенном языке для последующего использования. Для выполнения такого преобразования можно использовать специальный инструмент, расположенный на диске ИТС: [Технологическая поддержка – 1С:Предприятие 8 – Универсальные отчеты и обработки – Преобразование журнала действий пользователя \(<http://its.1c.ru/db/metod81#content:5014:1>\)](#).

С помощью этого инструмента можно:

- Выполнять преобразование как файла с журналом (например, ранее записанным), так и текста, который можно ввести вручную (в формате журнала).
- Указывать вариант встроенного языка для получившегося текста на встроенном языке.
- Добавлять программный код, реализующий подключение к клиенту тестирования.
- Указывать необходимость разбиения длинных последовательностей действий на процедуры, каждая из которых работает с одним тестируемым окном прикладного решения.

Получившийся в результате преобразования текст можно использовать, например, для многократного автоматического исполнения преобразованного набора действий.

Глава 29. Механизм сравнения и объединения конфигураций

29.1. Общая информация

Режим сравнения и объединения конфигураций позволяет детально сравнить две конфигурации или расширения конфигурации и объединить их. Возможно выборочное объединение по результатам сравнения. Сравнение и объединение основной конфигурации и расширения конфигурации не поддерживается.

Режимом объединения можно воспользоваться, например, когда одну конфигурацию параллельно разрабатывают несколько человек. Время от времени им необходимо объединять результаты работы (или хотя бы один раз объединить все результаты в конце разработки).

Режим сравнения и объединения конфигураций можно также использовать для сравнения двух конфигураций без объединения.

Наконец, в ситуации, когда в исходную конфигурацию нужно загрузить сделанные изменения, также полезно предварительно просмотреть и оценить их, воспользовавшись режимом сравнения и объединения конфигураций.

Сравнению в этом режиме подвергаются как общие свойства объектов конфигураций, таких как справочники, документы, журналы, так и их отдельные реквизиты и реквизиты табличных частей. Отдельно сравниваются формы объектов: модули, тексты описаний и макеты. При этом результаты сравнения можно просмотреть детально.

29.2. Условия сравнения конфигураций

При запуске режима сравнения программа производит анализ конфигураций и устанавливает взаимное соответствие объектов.

Объекты, которым «не досталось пары», могут быть либо новыми, либо удаленными, либо это объекты, относящиеся к одной категории, но называемые в разных конфигурациях по-разному. Для этих объектов соответствие можно установить интерактивно (см. [здесь](#)).

При автоматической установке соответствия программа анализирует объекты конфигураций, имеющие одинаковые имена. Например, **Справочник.Товары** в одной конфигурации и **Справочник.Товары** в другой считаются одним справочником; им устанавливается соответствие, и они будут сравниваться.

Среди объектов, имеющих разные имена, отдельно проводится сравнение внутренних идентификаторов, которые конфигуратор присваивает автоматически при их создании. Присвоение внутренних идентификаторов необходимо для того, чтобы упростить установку соответствия объектов конфигураций и производить ее автоматически. Эта ситуация возникает, например, в том случае, когда в процессе развития конфигурации было изменено имя объекта. Если имена различны, а внутренние идентификаторы совпадают, то для таких объектов автоматически устанавливается соответствие.

Особым образом устанавливается соответствие форм. Форма может состоять из нескольких страниц. Сначала производится анализ по именам; затем для страниц, имеющих разные имена, производят анализ по заголовкам страниц (непустым); затем для страниц, имеющих разные заголовки, производят анализ по картинкам. Если установлено соответствие страниц, то будут сравниваться и объединяться расположенные в них элементы управления. Если соответствие страниц не установлено, то считается, что это разные страницы, и в результирующей форме (после объединения) будут присутствовать все такие страницы отдельно.

Для страниц, у которых все три свойства (**Внутренний идентификатор**, **Заголовок** и **Картинка**) пустые, будет производиться попарное объединение в соответствии с выбранным порядком следования.

29.3. Запуск режима сравнения и объединения конфигураций

Как будет видно ниже, хотя сравниваемые конфигурации можно считать в основном равноправными (приоритет при объединении можно отдать любой из них), в определенном аспекте именно текущая

конфигурация считается основной, в нее вносятся не повреждающие ее изменения из загружаемой конфигурации. Исходя из этих соображений, рекомендуется выбрать, какую из двух конфигураций сделать основной. Выбранная конфигурация становится основной, когда с ней запускается система «1С:Предприятие» в режиме Конфигуратор.

Загружаемую конфигурацию, которую можно будет сравнить и объединить с основной, необходимо предварительно сохранить в файл (см. [здесь](#)).

Для загрузки второй конфигурации нужно выбрать пункт [Конфигурация – Сравнение и объединение конфигураций](#). Появится стандартный диалог [Открыть файл конфигурации](#), в котором нужно найти и открыть файл с расширением *.cf загружаемой конфигурации.

Если основная конфигурация пустая (разработка прикладного решения еще не начиналась), то конфигуратор предложит выполнить полную загрузку конфигурации. При положительном ответе производится полная загрузка указанной конфигурации и открывается окно [Конфигурация](#). При отказе от полной загрузки производится сравнение конфигураций.

Процесс загрузки и сравнения загружаемой конфигурации с основной сопровождается сообщениями в панели состояния конфигуратора. В этих сообщениях указываются объекты конфигураций, сравниваемые в настоящий момент. Также в панели состояния показывается индикатор процесса сравнения.

Процесс загрузки и сравнения может занимать продолжительное время. Это зависит от размера сравниваемых конфигураций.

29.4. Сравнение конфигураций

В дополнение к основному режиму сравнения и объединения конфигураций, описанному в предыдущем разделе, при котором одна из сравниваемых конфигураций всегда выбирается основной, конфигуратор предоставляет возможность сравнить две произвольные конфигурации, например, расположенные в файлах, или конфигурацию хранилища (с выбором любой версии) и поставки и т. д.

Выбрать сравниваемые конфигурации можно с помощью меню [Конфигурация – Сравнить конфигурации](#).

На экран выводится диалог:

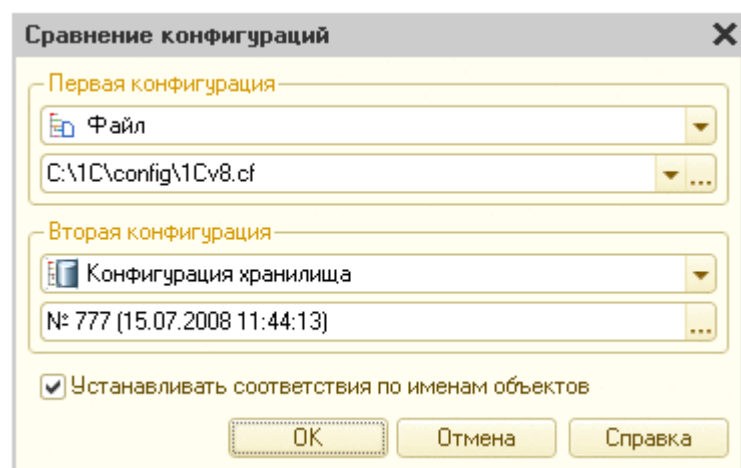


Рис. 456. Сравнение конфигураций

Диалог содержит два раздела выбора сравниваемых конфигураций. Приемы работы в разделах одинаковые.

Выбор сравниваемых конфигураций может производиться из основной конфигурации, конфигурации базы данных, конфигурации, хранящейся в файле, версии конфигурации хранилища, конфигурации поставщика (если текущая конфигурация находится на поддержке), расширением конфигурации расширением конфигурации, сохраненным в базе данных.

Если выбран файл или хранилище, то в соответствующий раздел диалога добавляется поле для выбора файла или версии хранилища. В случае выбора файла, можно выбирать файлы как собственно

конфигураций (файлы *.cf), так и расширений конфигурации (файлы *.cfe).

Если в базе данных сохранено более одного расширения, то в соответствующий раздел диалога добавляется поле выбора необходимого расширения.

Если сравниваемые конфигурации не являются потомками друг друга и поэтому для них не может быть произведена установка соответствия по внутренним идентификаторам, нужно установить флажок [Установить соответствие по именам объектов](#). Для начала сравнения нажать кнопку **ОК**. Конфигуратор произведет сравнение указанных конфигураций и выведет результат сравнения в окно [Сравнение конфигураций](#). Приемы работы в окне аналогичны приемам работы, описанным в разделе «Сравнение и объединение конфигураций», с тем лишь отличием, что в режиме сравнения можно только просмотреть выбранные объекты и получить отчет о различиях.

29.5. Особенности сравнения предопределенных данных

При сравнении предопределенных данных сопоставление элементов производится только по внутреннему идентификатору.

Уникальный (в пределах объекта метаданных) внутренний идентификатор присваивается предопределенному элементу при создании. Он сохраняется при любых перемещениях предопределенных данных между конфигурациями при объединении или копировании. В этом поведении идентификатора отличается от поведения соответствующего идентификатора объекта метаданных. Поэтому если в двух конфигурациях создать элементы предопределенных данных с одинаковым именем, при сравнении они будут считаться разными элементами. Единственный способ получить одинаковые предопределенные данные в двух конфигурациях – воспользоваться объединением или копированием.

При обновлении конфигурации базы данных производится сопоставление предопределенных элементов с существующими объектами базы данных. Неизменность внутреннего идентификатора обеспечивает корректную связь между элементом предопределенных данных и соответствующим ему объектом базы данных при перемещениях между конфигурациями. Для сопоставленных объектов замещение будет производиться в соответствии с выбором приоритетной конфигурации.

В связи с тем, что установка соответствия производится по внутренним идентификаторам, для иерархических структур предопределенные данные могут объединяться, даже если они находятся на разных уровнях иерархии. Результат объединения (уровень иерархии, в которой окажется объединяемый элемент) зависит от приоритетности конфигураций.

29.6. Особенности сравнения и объединения расширений конфигурации

Операция сравнения и объединения расширения поддерживается для:

- Расширения конфигурации, сохраненного в файл (*.cfe);
- Расширения конфигурации, сохраненного в информационной базе.

Не поддерживается сравнение и объединение между расширением конфигурации и основной конфигурацией в любых вариантах.

Существует возможность вручную устанавливать соответствие между объектами сравниваемых расширений.

При сравнении заимствованных объектов учитываются как собственно значения объектов, так и признак контролируемости. В отчет о сравнении попадают и различия в значениях объектов и различия в признаках контролируемости. В тоже время, если сравнивается собственный и заимствованный объект – то проверяется только значение объекта, без признака контролируемости.

При объединении свойств объектов следует учитывать следующие особенности:

- Свойство [Тип объекта](#) отображается, но недоступно для настройки.

- Для объектов с установленным признаком контролируемости, объединение выполняется следующим образом:
 - Для пары «собственный – заимствованный»: значения объединяются, признак контролируемости не используется.
 - Для пары «заимствованный – собственный»: значения объединяются, признак контролируемости не изменяется.
 - Для пары «заимствованный – заимствованный»: значения объединяются, признак контролируемости получается из конфигурации, с которой выполняется объединение.
- Сохраненная форма считается единым объектом, и действия с ней зависят от режима объединения:
 - Объединение выполняется с приоритетом расширения конфигурации – сохраненная форма не изменяется.
 - Объединение выполняется с приоритетом файла – сохраненная форма получается из файла.

29.7. Окно «Сравнение и объединение конфигураций»

После окончания процедуры сравнения на экран выводится окно [Объединение конфигураций](#). Оно позволяет:

- ознакомиться с тем, какие объекты в двух конфигурациях отличаются;
- выбрать объекты, по которым необходимо детально изучить различия;
- вызвать указанные объекты для просмотра и редактирования;
- задать, какие из объектов включить в объединение;
- установить соответствия объектов;
- установить режимы объединения конфигураций (для всей конфигурации или по каждому объекту);
- установить порядок подчиненных объектов (для всей конфигурации или по каждому объекту);
- сформировать отчет о различиях в конфигурациях;
- запустить процесс объединения конфигураций.

Окно (см. [рис. 457](#)) содержит табличное поле, которое состоит из трех колонок, если конфигурации не связаны друг с другом (о связанных конфигурациях см. [здесь](#)). В первой колонке приводятся данные основной конфигурации (той информационной базы, которая открыта конфигуратором), во второй колонке – данные загружаемой конфигурации, а третья предназначена для установки режима объединения и порядка подчиненных объектов.

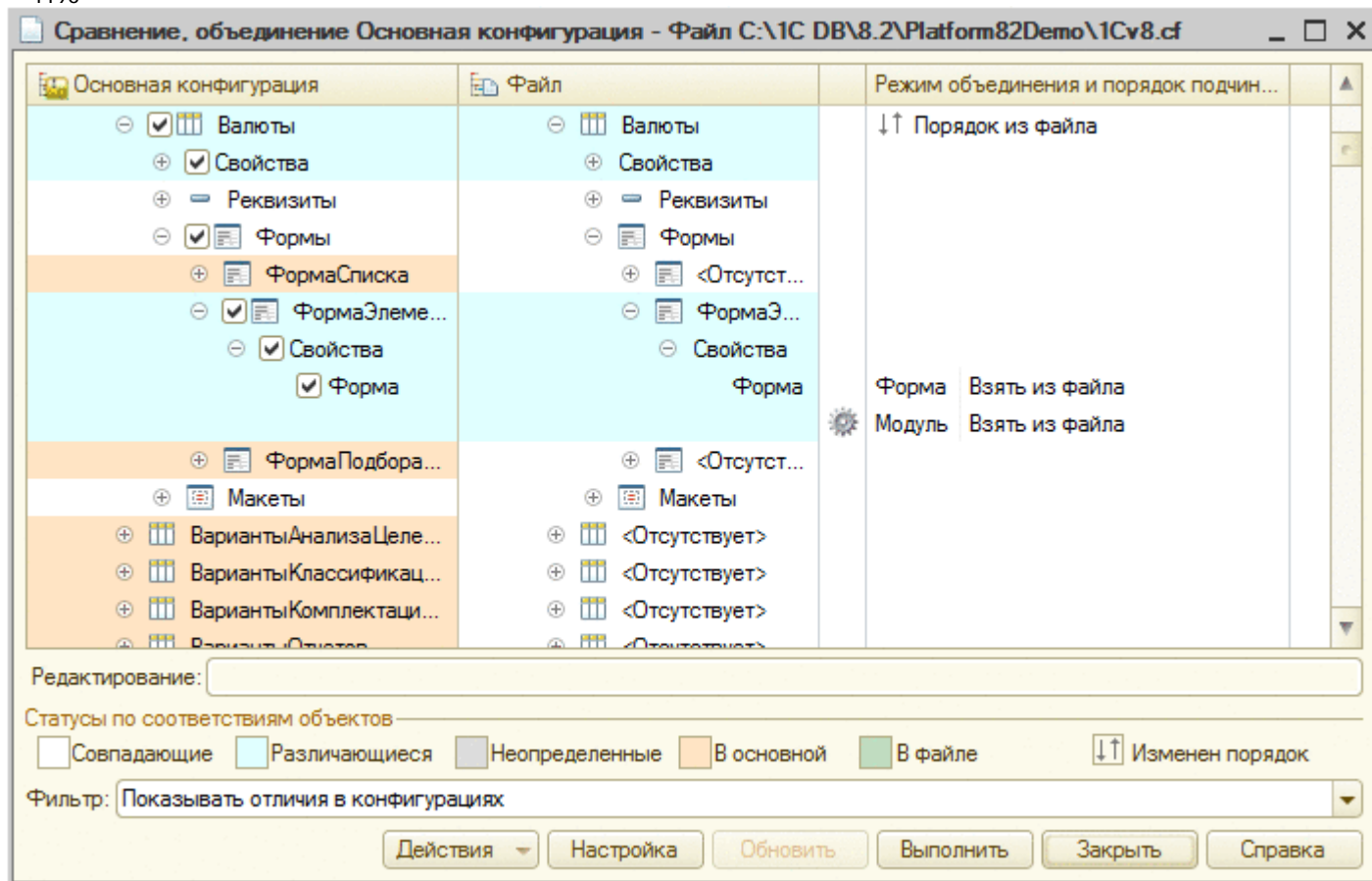


Рис. 457. Сравнение и объединение конфигурации

Признаком отличия (статус соответствия) объектов является цвет фона строк. Под табличным полем приведены возможные статусы.

Ниже расположено поле выбора **Фильтр**, с помощью которого устанавливается режим просмотра изменений. При открытии окна **Объединение конфигураций** установлен фильтр **Показывать отличия в конфигурациях**. Выбор других режимов позволяет изучить отличия в конфигурациях или совпадения.

Нажатие кнопки **Действия** открывает контекстное меню.

Пункт **Установить режим для всех** открывает диалог установки режима объединения (см. [здесь](#)) и установку порядка (см. [здесь](#)) всех объектов.

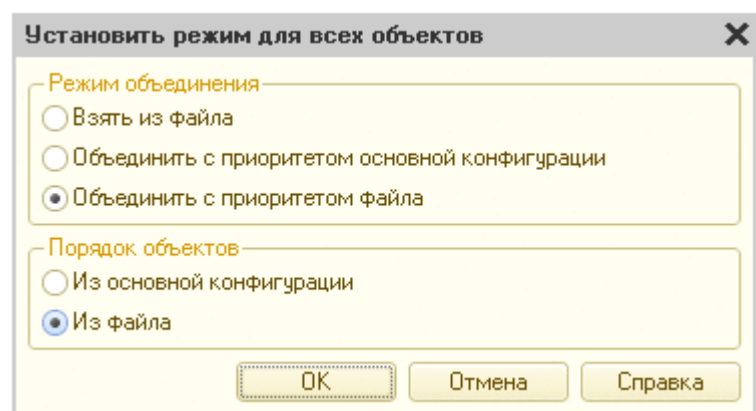


Рис. 458. Установка режимов для всех объектов

Пункт **Отметить по подсистемам...** позволяет установить отбор по подсистемам.

Пункт **Отчет о сравнении** позволяет получить отчет о сравнении двух конфигураций с нужной степенью подробности.

ПРИМЕЧАНИЕ. При открытом окне сравнения и объединения конфигураций не рекомендуется

выполнять длительные операции по работе с конфигурацией, например, операции с хранилищем, снятие с поддержки конфигурации поставщика и т. д.

29.8. Настройка сравнения конфигураций

Нажатие кнопки [Настройка](#) открывает диалог [Настройка сравнения конфигураций](#).

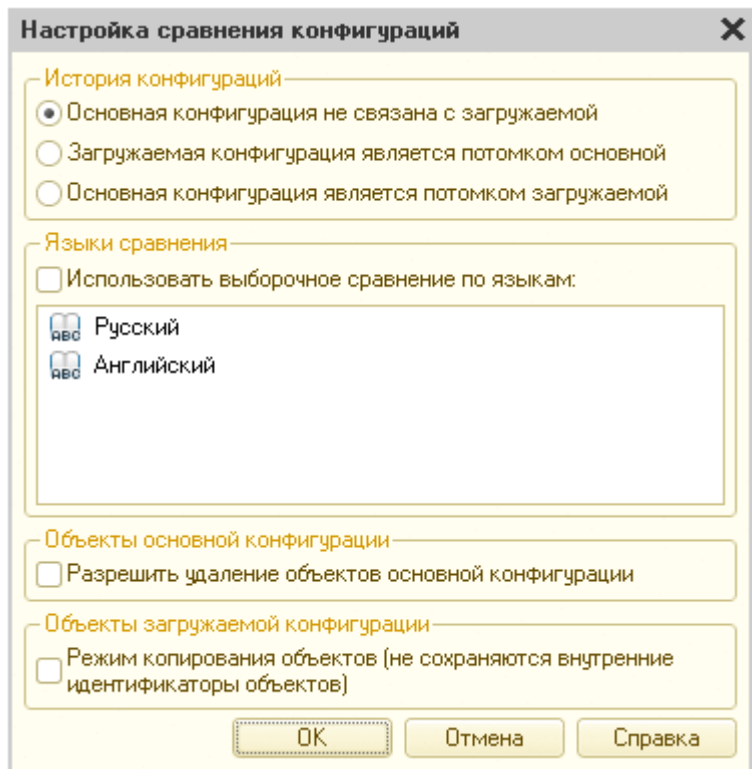


Рис. 459. Настройка сравнения конфигураций

В диалоге производится выбор режима сравнения конфигураций. По умолчанию установлен режим сравнения различных конфигураций (не связанных друг с другом).

Если конфигурации связаны, нужно выбрать характер связи. При закрытии диалога настройки (нажата кнопка [ОК](#)) в окно [Объединение конфигурации](#) добавляется специальная графа, в которой специальными пиктограммами показывается история изменений объектов.

Специальная графа добавляется для колонки [Загружаемая конфигурация](#), если загружаемая конфигурация является потомком основной конфигурации; для колонки [Основная конфигурация](#), если основная конфигурация является потомком загружаемой конфигурации.

Также в окно добавляется справочная информация для расшифровок статусов, объединенных в группу [Статусы](#) по истории объектов.

Если в конфигурации используется несколько языков, то можно произвести выборочное сравнение по языкам. Для этого нужно установить флажок [Использовать выборочное сравнение по языкам](#) и указать язык или языки, для которых необходимо произвести сравнение.

29.8.1. Группа «Объекты основной конфигурации»

При выполнении любого объединения конфигураций существует возможность удаления объектов основной конфигурации. По умолчанию эта возможность включена только в режиме обновления конфигурации поставщика. Для включения ее в других режимах следует установить флажок [Разрешить удаление объектов основной конфигурации](#).

Расстановка пометок удаления объектов поставщика по умолчанию производится по следующим правилам. Если пользователь изменял объект поставщика по сравнению с предыдущей версией конфигурации поставщика, то объект по умолчанию не помечается на удаление; если объект идентичен объекту поставщика предыдущей версии, то он на удаление помечается. Если объект был помечен на удаление (автоматически или вручную), то при нажатии кнопки [Выполнить](#) происходит контроль

ссылочной целостности. При обнаружении неразрешимых ссылок на удаляемый объект будет выдано окно показа таких ссылок, но, в отличие от неразрешимых ссылок, образующихся в результате отказа от копирования какого-либо объекта конфигурации поставщика (или любой другой конфигурации, участвующей в объединении), возможности продолжить объединение (и удаление объекта) в этом случае нет.

29.8.2. Группа «Объекты загружаемой конфигурации»

По умолчанию при объединении конфигурации объекты, отсутствующие в основной конфигурации, добавляются в нее с теми идентификаторами, которые существуют у них в загружаемой конфигурации.

Установка флажка [Режим копирования объектов...](#) позволяет выполнять добавление новых объектов по правилам, существовавшим в предыдущих версиях платформы: при добавлении объектов, отсутствующих в основной конфигурации, для них всегда генерируются новые идентификаторы.

При нажатии кнопки [ОК](#) производится повторное сравнение с учетом выбранной настройки.

29.9. Редактирование объектов

В обычном режиме (не используется хранилище и отсутствует поддержка) при открытом окне [Объединение конфигураций](#) допускается производить просмотр и редактирование объектов основной конфигурации.

Для редактирования свойств объекта следует указать нужный объект в колонке основной конфигурации и в палитре свойств исправить значение. Если редактируемый объект представляет собой форму, макет, интерфейс или другой «сложный» объект, его можно открыть для редактирования с помощью выбора пункта [Открыть форму \(макет/интерфейс\)](#) из контекстного меню. Объекты основной конфигурации также можно выбрать для редактирования в окне [Конфигурация](#).

Объекты загружаемой конфигурации доступны только для просмотра. Процедура доступа к объекту загружаемой конфигурации осуществляется аналогично процедуре доступа к объектам основной конфигурации.

Если было произведено любое изменение, кнопка [Выполнить](#) становится недоступной и всем объектам устанавливается статус [Неопределенный](#) (показывается серым фоном). Чтобы продолжить работу в окне [Объединение конфигураций](#), следует нажать кнопку [Обновить](#). Конфигуратор производит повторное сравнение конфигураций (с учетом произведенных изменений) и устанавливает соответствие объектов.

29.10. Особенности редактирования при групповой разработке и поддержке

При использовании режима групповой разработки в окне [Объединение конфигураций](#) допускается редактирование только захваченных в хранилище объектов основной конфигурации.

Объекты, запрещенные для редактирования, можно только просмотреть. Эти объекты нельзя указать для объединения (отсутствует флажок выбора).

Признаком запрета редактирования является наличие пиктограммы в виде замкнутого замка слева от пиктограммы объекта. Кроме того, под табличным полем выводится статус объекта, в который выводится текстовое сообщение о запрете редактирования.

Аналогичное поведение окна [Объединение конфигураций](#) наблюдается в случае нахождения конфигурации на поддержке. Если для какого-либо объекта установлено правило поставщика, запрещающее изменение, то данный объект можно только просмотреть. Данные объекты можно только сравнить, но объединение таких объектов не допускается.

29.11. Установка соответствия объектов

При сравнении конфигураций программа производит автоматическую установку соответствия объектов двух конфигураций друг другу. Так как соответствие производится по именам объектов, то в сравниваемых конфигурациях имена объектов могут совпадать. Однако суть объектов может быть разная. С другой стороны, имена могут быть различны, но суть объектов одна. В таких случаях для этих

объектов нужно провести ручную установку соответствия объектов.

Для отмены установленного соответствия следует выбрать объект и в контекстном меню выбрать пункт **Отменить соответствие объектов**. При этом объектам устанавливается статус **Неопределенный**. После завершения отмены соответствия для всех таких объектов следует нажать кнопку **Обновить**. Конфигуратор произведет повторное сравнение конфигураций.

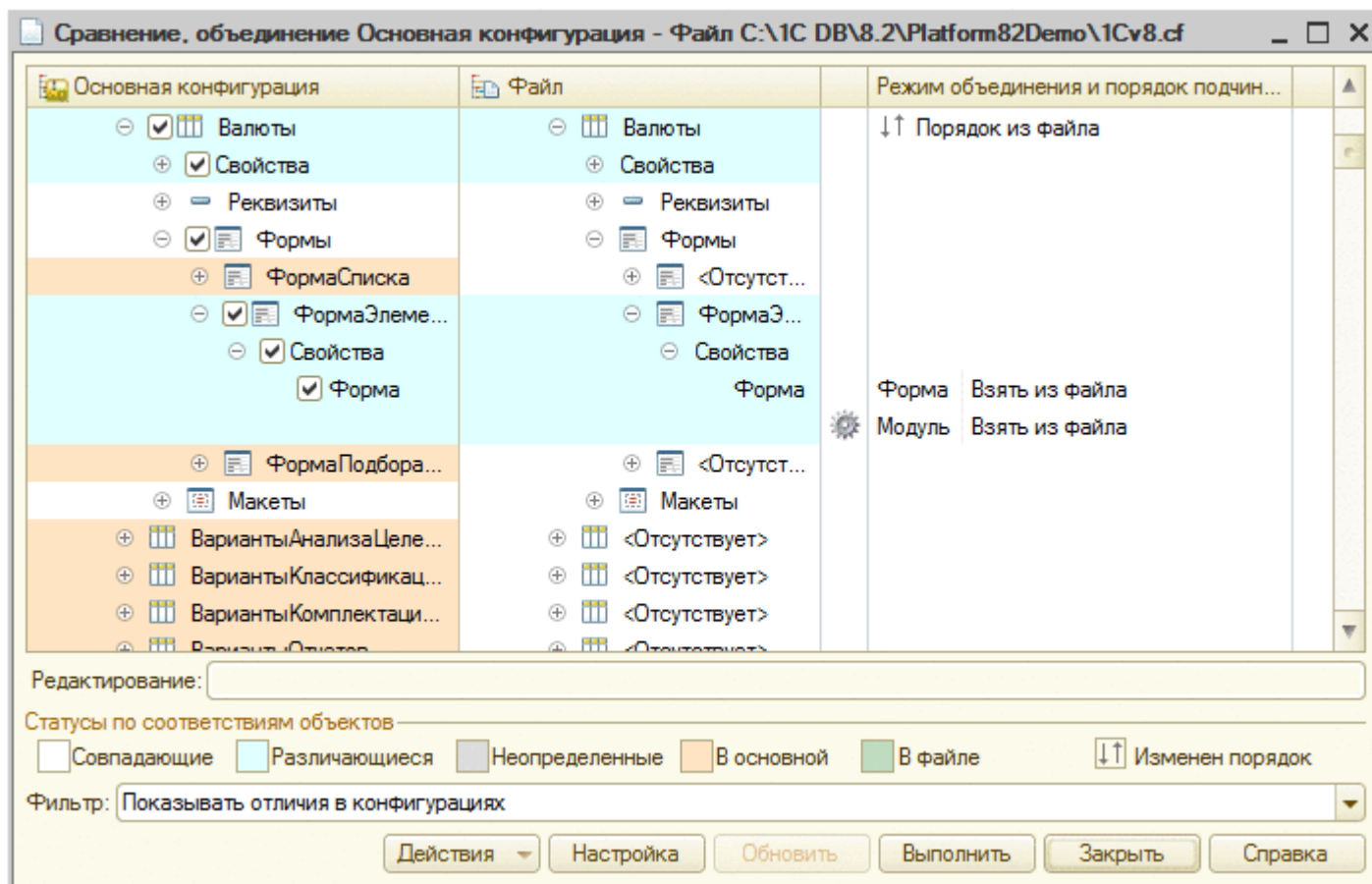


Рис. 460. Основное окно сравнения/объединения конфигураций

Для основной конфигурации объекты будут показаны бежевым фоном (присутствуют только в основной конфигурации), для загружаемой – зеленым (присутствуют только в загружаемой конфигурации).

Для установки соответствия нужно выбрать объект и в контекстном меню выбрать пункт **Установить соответствие объектов**. На экран выводится список объектов, с которыми допускается устанавливать соответствие. В список включаются объекты, совпадающие по виду, и для которых соответствие еще не установлено. Нужно выбрать объект и нажать кнопку **ОК**.

29.12. Режим объединения

Режим объединения можно установить для каждого объединяемого объекта. Существует два вида режима:

- **Взять из загружаемой конфигурации** (в зависимости от загружаемой конфигурации наименование режима объединения может отличаться);
- **Объединить** – в этом случае обычно указывается приоритет конфигураций при объединении.

Выбор режима **Взять из загружаемой конфигурации** означает, что объект конфигурации будет добавлен, если он новый, или замещен, если он измененный. Причем переносится вся структура объекта, модули, описания, формы. При этом приоритет однозначно отдается загружаемой конфигурации.

При выборе метода **Объединить** происходящее при объединении конфигураций зависит от установленного приоритета.

Если из загружаемой конфигурации необходимо взять только новое и максимально сохранить старое,

нужно выбрать **Объединить с приоритетом основной конфигурации**. В этом режиме добавляются только новые объекты.

Тексты объединяются следующим образом:

Текст	Приоритет	Результат
Удаленное (имеется в основной конфигурации, отсутствует в загружаемой)	С приоритетом основной	Остается в тексте
Новое (имеется в загружаемой конфигурации и отсутствует в основной)		Добавляется
Измененное		Добавляется в виде комментариев из загружаемой конфигурации, а то, что было в основной, остается неизменным
Удаленное	С приоритетом загружаемой	Превращается в комментарии
Новое		Добавляется
Измененное		Измененное в основной конфигурации превращается в комментарии, а из загружаемой конфигурации – добавляется

Для модуля формы выбор приоритета можно проиллюстрировать следующим примером.

Текст модуля основной конфигурации:

[Копировать в буфер обмена](#)

```
Процедура УточнитьСтатус(Элемент)
    // Вставить содержимое обработчика
    Сообщить("Сообщение из основной конфигурации");
КонецПроцедуры
```

Текст модуля загружаемой конфигурации:

[Копировать в буфер обмена](#)

```
Процедура УточнитьСтатус(Элемент)
    // Вставить содержимое обработчика
    Сообщить("Сообщение из загружаемой конфигурации");
КонецПроцедуры
```

Результат при объединении с приоритетом основной конфигурации:

[Копировать в буфер обмена](#)

```
Процедура УточнитьСтатус(Элемент)
    // Вставить содержимое обработчика
    //{{MRG[ <-> ]
    Сообщить("Сообщение из основной конфигурации");
    //{{MRG[ <-> ]
    //{{MRG[ <-> ]
    // Сообщить("Сообщение из загружаемой конфигурации");
    //{{MRG[ <-> ]
КонецПроцедуры
```

Результат при объединении с приоритетом загружаемой конфигурации:

[Копировать в буфер обмена](#)

```
Процедура УточнитьСтатус(Элемент)
    // Вставить содержимое обработчика
    //{{MRG[ <-> ]
    Сообщить("Сообщение из загружаемой конфигурации");
    //{{MRG[ <-> ]
    //{{MRG[ <-> ]
    // Сообщить("Сообщение из основной конфигурации");
```


//{ {MRG[<->]
 КонечПроцедуры

Если объединение модулей выполняется с помощью внешней программы (описание настроек использования внешней программы см. [здесь](#)), то результат объединения зависит от используемой внешней программы.

Для объектов, значения свойств которых определяются простым указанием в палитре свойств (например, [Синоним](#), [Комментарий](#)), зависимость результата объединения от приоритета и наличия значений представлена в таблице:

Значение в основной	Значение в загружаемой	Приоритет конфигурации	Результат (значение выбирается из...)
Задано	Задано	Приоритет основной	Основной
Задано	Не задано		Основной
Не задано	Задано		Загружаемой
Задано	Задано	Приоритет загружаемой	Загружаемой
Задано	Не задано		Основной
Не задано	Задано		Загружаемой

Если свойство какого-либо объекта может представлять результат составного значения объектов двух конфигураций (например, свойство [Вводится на основании](#)), то результат определяется простым объединением или выбирается из загружаемой конфигурации.

Если в конфигурации используется несколько языков и в настройке указан режим выборочного сравнения по языкам, то для таких объектов производится сравнение и объединение с учетом указанных в настройке языков.

Макеты объединяются следующим образом: результирующий макет будет состоять из макета основной и загружаемой конфигураций, разделенных строкой, имеющей желтый фон. Сначала в итоговом макете следует макет приоритетной конфигурации, а затем – другой макет через разделитель. Окончательное объединение следует провести вручную после сравнения.

ПРИМЕЧАНИЕ. В окне [Объединение конфигураций](#) свойство [Тип макета](#) не показывается. При объединении это свойство объединяется так же, как свойство [Макет](#).

При объединении управляемых форм не поддерживается графическое отображение отличий в окне [Объединение конфигураций](#), но возможно отображение результатов сравнения форм в текстовом виде в отчете о сравнении объектов. При сравнении форм выполняется сопоставление по следующим правилам:

- тексты модулей сравниваются по именам процедур;
- элементы формы сравниваются по типу элемента и его имени;
- параметры формы сравниваются по имени;
- команды сравниваются по имени;
- реквизиты сравниваются по именам;
- для командного интерфейса сравнение выполняется по виду команды (автоматическая или реализованная разработчиком) и имени команды.

Совпадающие элементы (по именам) в основной форме будут заменены (заменяются свойства, размеры и расположение), если выбран приоритет загружаемой конфигурации, и оставлены без изменения, если выбран приоритет основной формы. Некоторые свойства элементов формы могут быть объединены (например, список выбора, заголовки на разных языках и т. д.).

Режим объединения формы можно задавать как в целом для всей формы (сама форма и модуль формы) – для обычной и управляемой формы, так и отдельно для формы и ее модуля – только для управляемой формы, как показано на [рис. 461](#).

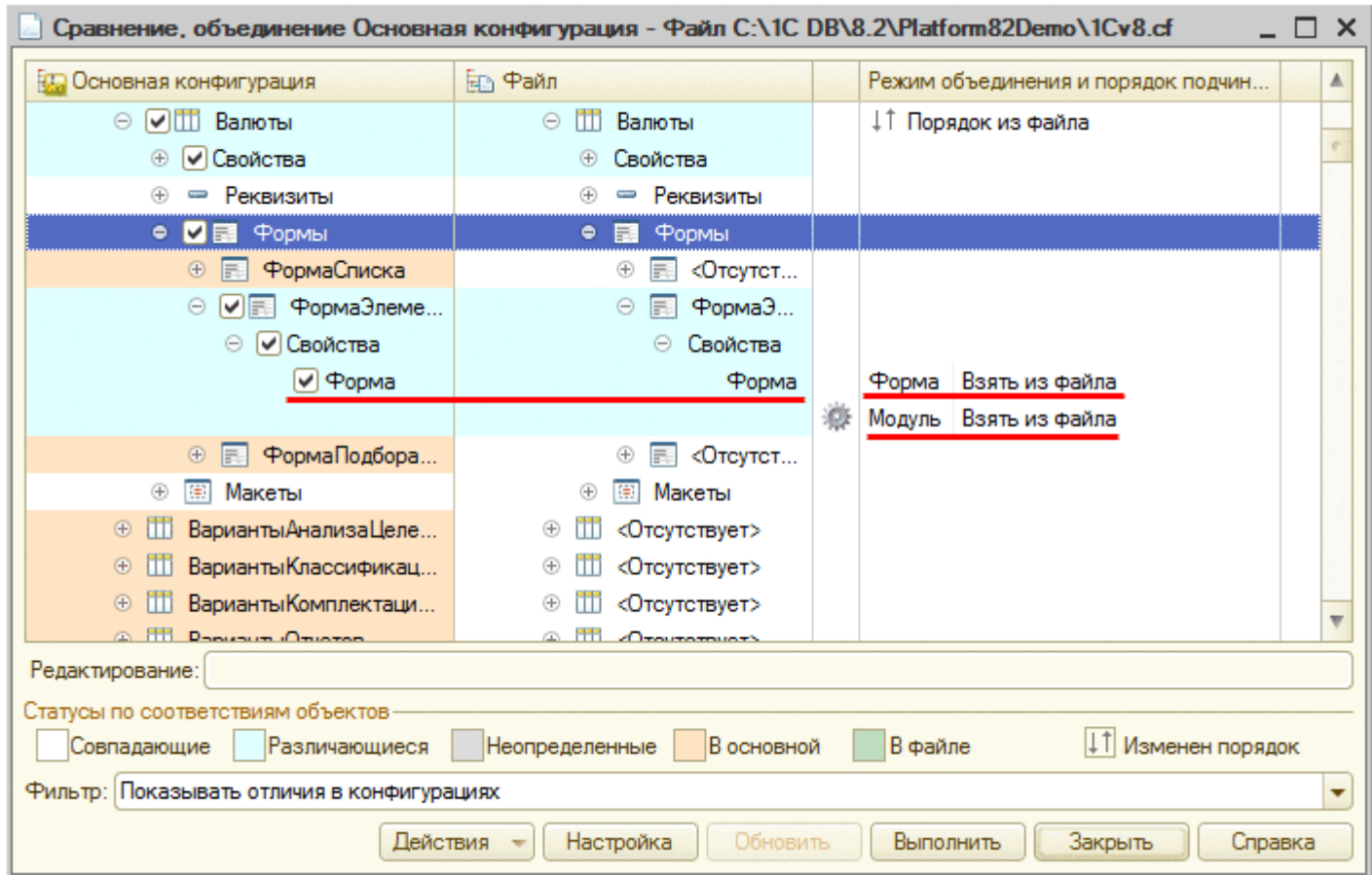


Рис. 461. Настройка объединения формы

Для установки единого правила для управляемой формы и ее модуля необходимо использовать контекстное меню у той формы, для которой необходимо установить правило. В открывшемся контекстном меню следует выбрать необходимый пункт из подменю **Режим объединения для подчиненных объектов**.

При нажатии на картинку шестеренки, расположенной в специальной колонке, можно открыть окно сравнения (или объединения) модулей (см. [здесь](#)). В зависимости от настроек конфигуратора (см. [здесь](#)), для этой операции будет использовано либо инструмент, встроенный в конфигуратор, либо внешняя программа. Если в данной колонке расположена картинка шестеренки с зеленой отметкой, то это означает, что пользователь выполнял настройки объединения в данном модуле. Если в колонке отображается картинка шестеренки с восклицательным знаком, это означает, что при выполнении объединения с помощью внешней программы есть неразрешенные конфликты объединения (только для обновления конфигурации, находящейся на поддержке).

Некоторые действия, которые можно выполнить в окне сравнения и объединения конфигурации, зависят от настроек конфигуратора (см. [здесь](#)):

- Команда контекстного меню **Показать различия в модулях**:
 - Настройка конфигуратора: **Настройка объединения двух конфигураций**;
 - Настройка внешней программы: **Объединение двух файлов**;
 - Результаты работы: игнорируется.
- Команда контекстного меню **Показать различия в модулях без учета структуры**:
 - Настройка конфигуратора: **Показать различия без учета структуры**;

- Настройка внешней программы: [Сравнение двух файлов](#);
- Результаты работы: игнорируется.
- Команда контекстного меню [Настройка объединения модулей](#):
 - Настройка конфигуратора: [Настройка объединения двух конфигураций](#);
 - Настройка внешней программы: [Объединение двух файлов](#);
 - Результаты работы: используется в качестве результата объединения.

При вызове команды будет использоваться тот инструмент, который указан в соответствующей настройке конфигуратора, а если в качестве инструмента выбрана какая-либо внешняя программа – для выполнения действия будет использоваться командная строка, заданная в соответствующей настройке внешней программы. Если для выбранного действия используется встроенный инструмент конфигуратора, то при выполнении операции сравнения будет недоступно редактирование результата объединения.

Для некоторых объектов (реквизитов объектов) выбор режима объединения может отсутствовать. Так, например, для реквизита с примитивным типом (например [Число](#)) режим объединения устанавливается только [Взять из загружаемой конфигурации](#).

Для некоторых объектов выбор приоритетности также может отсутствовать. Например, для свойства [Подсистемы](#) можно выбрать режимы объединения [Взять из загружаемой конфигурации](#) и [Объединить](#). Если выбран режим [Объединить](#), то происходит простое объединение указанных подсистем двух конфигураций.

Правила объединения предопределенных данных см. [здесь](#).

29.13. Установка порядка подчиненных объектов

Для видов объектов, которые могут иметь неограниченное число подчиненных объектов (например, подчиненные объекты [Реквизиты](#), [Формы](#), [Макеты](#), [Табличные части](#) справочников и документов), в некоторых случаях важен порядок следования объектов в каждой такой группе.

Если в процессе сравнения конфигураций конфигуратор определил соответствие объектов, входящих в группу, но их порядок в конфигурациях различен, то в колонке [Режим объединения](#) и порядок подчиненных объектов можно указать нужный порядок следования объектов.

Для указания порядка следует выбрать любой подчиненный объект и в третьей колонке выбрать вариант установки порядка: [Порядок из основной конфигурации](#) или [Порядок из загружаемой конфигурации](#).

Выбор порядка действует на все подчиненные объекты данного вида.

Если для выбранного объекта его подчиненные объекты изменены и расположены в другом порядке, то в третьей колонке можно выбрать только порядок. Режим объединения указывается в объекте верхнего уровня с помощью контекстного меню. Например, в обеих конфигурациях для документа [Счет](#) есть несколько подчиненных объектов типа [Формы](#). Они изменены, и порядок их следования также изменен. Для выбора режима объединения следует указать объект более высокого уровня ([Формы](#) или [Счет](#)) и в контекстном меню выбрать нужный режим объединения.

29.14. Просмотр различий

Перед объединением конфигураций полезно просмотреть конкретные различия их объектов.

Это можно сделать с помощью палитры свойств. Нужно открыть палитру свойств и выбрать нужный объект основной или загружаемой конфигурации. Для просмотра форм или макетов их можно открыть. При этом если производится сравнение с обычным файлом конфигурации, не являющимся поставкой (см. [здесь](#)), можно не просто просмотреть данные загружаемой конфигурации, но и внести в них необходимые исправления. Если исправления сделаны, то после окончания объединения конфигуратор предложит сохранить сделанные в загружаемой конфигурации изменения.

Для просмотра различий в модулях в контекстном меню объекта нужно выбрать пункт [Показать различия в модулях без учета структуры](#).

При выборе этого пункта открывается окно [Различие между](#). Если сравниваемые тексты большие, перед открытием окна может пройти несколько секунд. Подробнее о просмотре различий см. [здесь](#).

Для выборочного сравнения модулей предназначен режим [Объединение модулей](#) (см. ниже).

Для просмотра различий в макетах нужно выбрать в контекстном меню макета пункт [Показать различия в макетах](#).

На экран выводится окно сравнения.

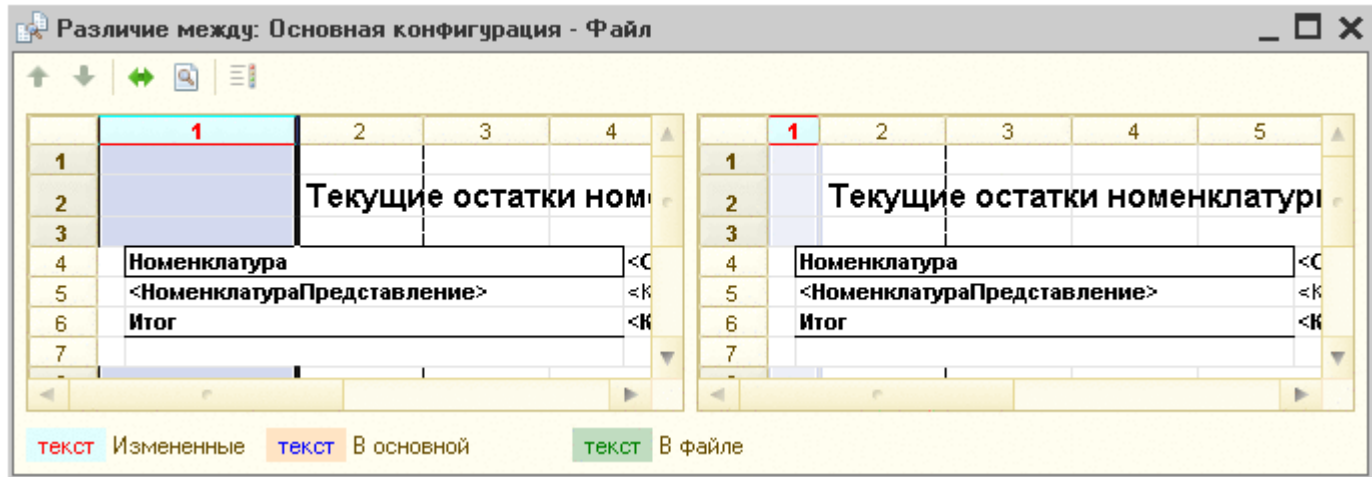


Рис. 462. Объединение макетов

Сравнение табличных документов производится по ячейкам. Если ячейки различны (любые свойства, кроме свойства [Параметр расшифровки](#)), то различие показывается цветом. Пояснение цвета показывается внизу окна сравнения.

29.15. Сравнение модулей

При объединении конфигураций можно настроить процесс объединения процедур и функций любого модуля: модуля приложения, общих модулей, модулей объектов, модулей форм.

Для этого в окне [Объединение конфигурации](#) нужно выбрать строку, соответствующую объединяемым модулям, и в специальной колонке щелкнуть мышью по символу шестеренки. Если в настройках конфигуратора (см. [здесь](#)) для выполнения операции сравнения модулей выбрана внешняя программа, то она будет запущена при щелчке мышью по картинке шестеренки. Правила работы в этом случае определяются этой внешней программой и с ними следует знакомиться в документации к этой внешней программе.

На экран выводится окно [Сравнение модулей](#).

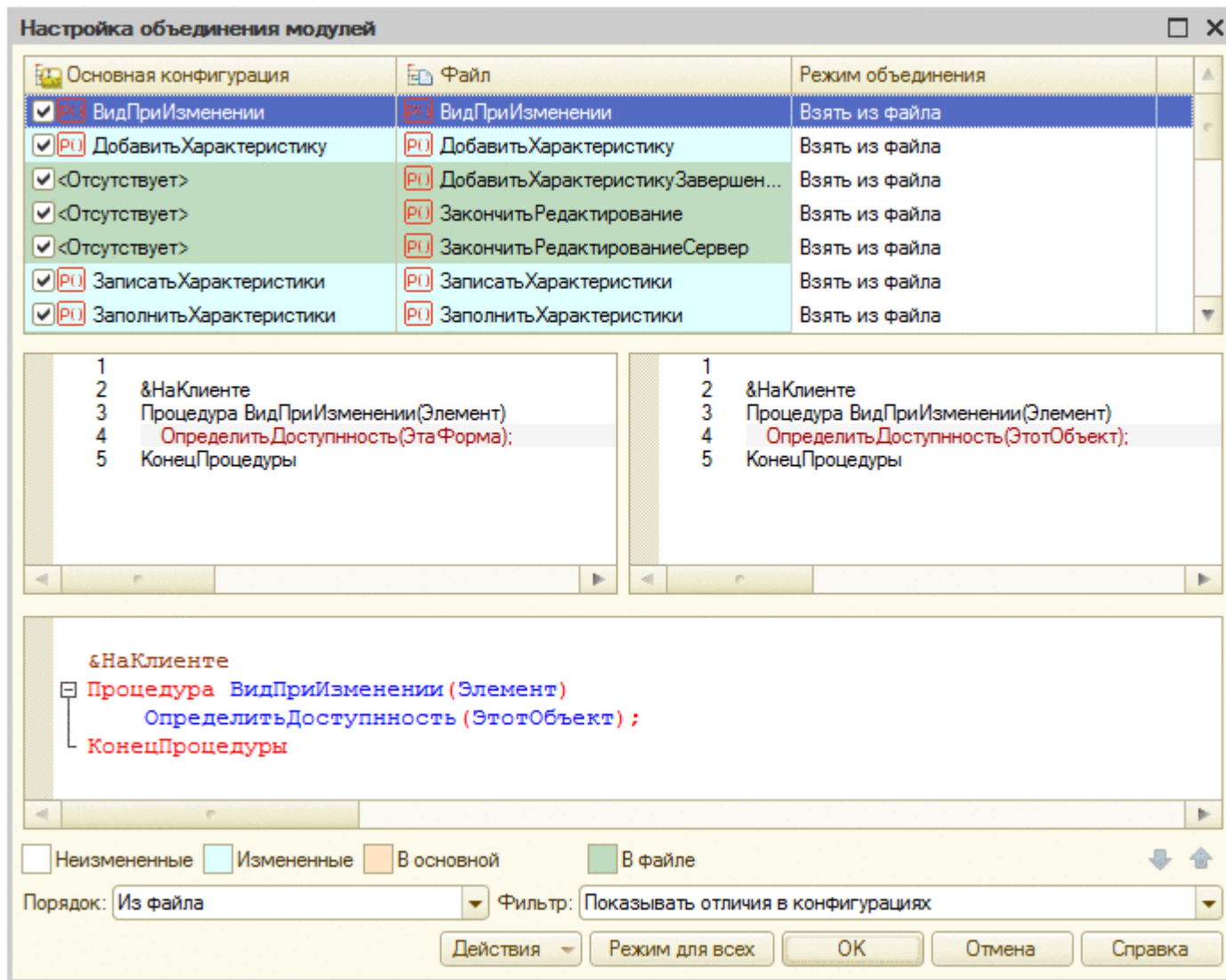


Рис. 463. Сравнение модулей

Верхняя часть окна содержит табличное поле, смысл которого соответствует табличному полю окна [Объединение конфигураций](#) с тем лишь отличием, что содержимое колонок формируется из наименований частей модуля: раздел определения переменных, раздел процедур и функций и раздел основной программы (далее объекты). Любой из этих разделов может отсутствовать. В нижней части формы отображается результат объединения по текущим правилам.

Состав табличного поля соответствует установленному фильтру (реквизит [Фильтр](#)).

В первой колонке по различающимся объектам следует установить пометки для тех объектов, по которым требуется произвести объединение. Для помеченных объектов можно установить режим объединения.

Сравнение производится по каждому разделу отдельно. Сравняются только те объекты, которым установлено соответствие. Первоначально производится автоматическая установка соответствия по именам.

Любое установленное соответствие процедур и функций можно отменить. Для этого нужно выбрать в списке строку с объектом и в контекстном меню выбрать пункт [Отменить соответствие объектов](#).

Если в сравниваемых модулях имена объектов были изменены, но суть их осталась прежней и требуется их сравнить, то следует произвести ручную установку соответствия объектов. Для этого следует выбрать любой из объектов, для которого будет произведена установка, и в контекстном меню выбрать пункт [Установить соответствие объектов](#).

На [рис. 464](#) видно, что метод `ДобавитьХарактеристику()` имеет разное содержание в основной конфигурации и в конфигурации из файла. В нижней части формы отображается результат объединения:

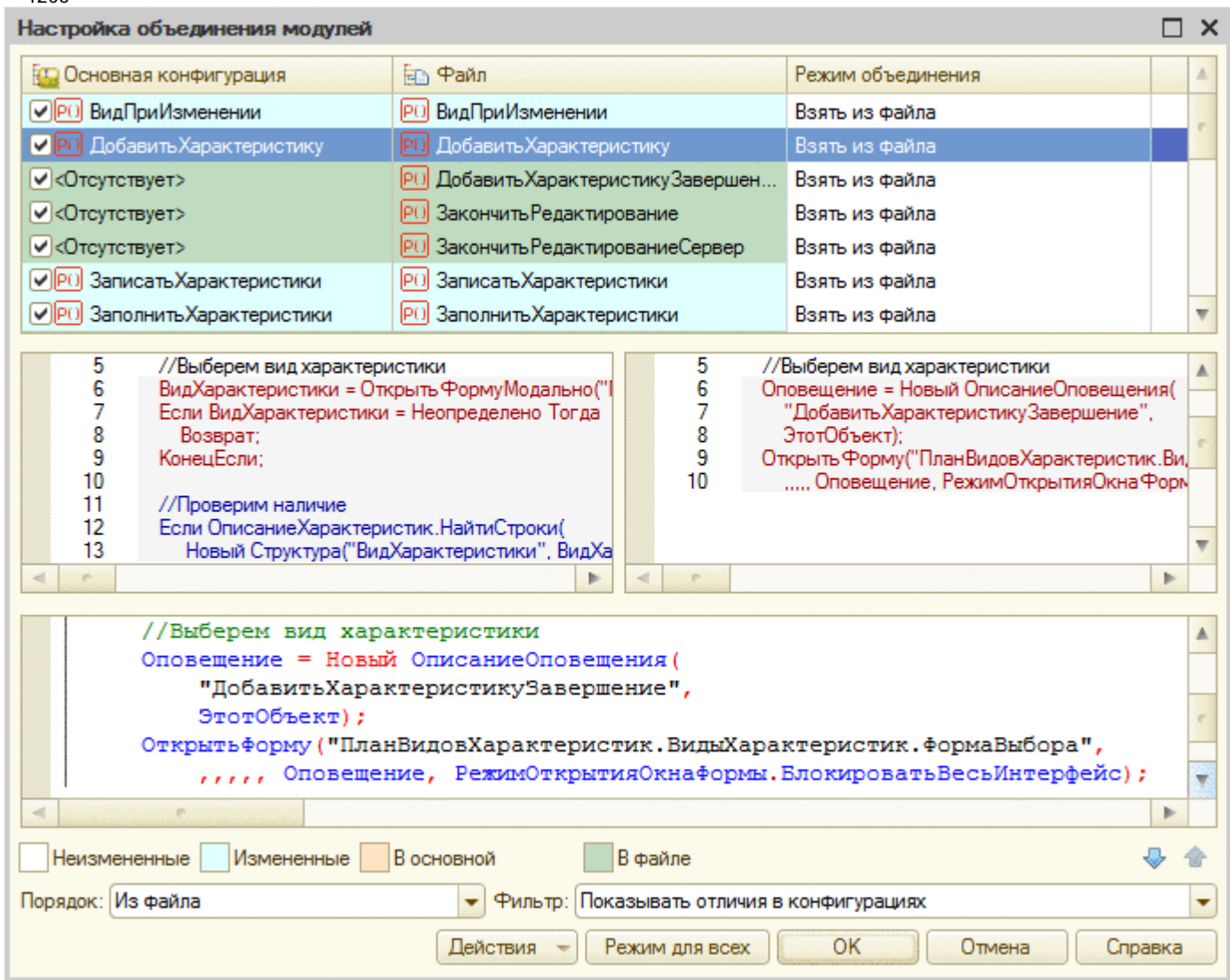


Рис. 464. Объединение модулей

В нижней части окна показывается модуль, который получится в результате объединения. Допускается выполнять редактирование получившегося текста непосредственно в процессе настройки объединения. В основную конфигурацию будет загружен текст модуля с учетом изменений, внесенных в процессе настройки.

Установка порядка следования объектов модуля определяется в соответствии со значением реквизита **Порядок**.

Для просмотра результата сравнения модулей без учета структуры нужно щелкнуть пункт **Действия – Показать различия в модулях без учета структуры**. На экран будет выведено окно **Различия между**. Подробнее о просмотре различий см. [здесь](#).

Для просмотра предварительного результата объединения модулей следует выбрать пункт **Действия – Показать результат объединения**.

После установки соответствия, определения порядка и режима объединения для сохранения настройки объединения модулей следует нажать кнопку **ОК**. Для отказа – кнопку **Отмена**.

После завершения процедуры объединения модулей в колонке **Режим объединения** и порядок подчиненных объектов окна **Объединение конфигураций** для модуля появится запись **Индивидуальная настройка**. Если в списке режимов объединения выбрать иной режим, то произведенная настройка теряется.

Собственно объединение модулей по выполненной настройке производится только для всей конфигурации.

29.16. Отбор объектов и запуск процесса объединения

Список объединяемых объектов формируется путем установки пометки слева от пиктограммы объекта основной конфигурации. Установка или снятие пометки для объекта верхнего уровня производит установку или снятие пометки для всех подчиненных объектов. Установленная пометка показывает, что данные объекты двух конфигураций будут объединяться.

Пометка объекта принимает состояние **Неопределено**, если для части подчиненных объектов пометка установлена, а для другой части снята.

При открытии окна **Объединение конфигурации** конфигуратор включает в список объединяемых объектов все допустимые объекты.

При нажатии кнопки **Выполнить** конфигуратор производит анализ данных конфигураций по установленным пометкам. Если не обнаруживается причин, препятствующих объединению, производится объединение. Объединенная конфигурация сохраняется автоматически.

Если производится выборочное объединение и выбраны такие объекты, которые ссылаются на другие объекты, не включенные в список объединяемых, то конфигуратор выводит список неразрешимых ссылок. Примером неразрешимой ссылки может служить выбор объекта, реквизит которого представляет ссылку на другой объект загружаемой конфигурации, не включенный в список объединяемых.

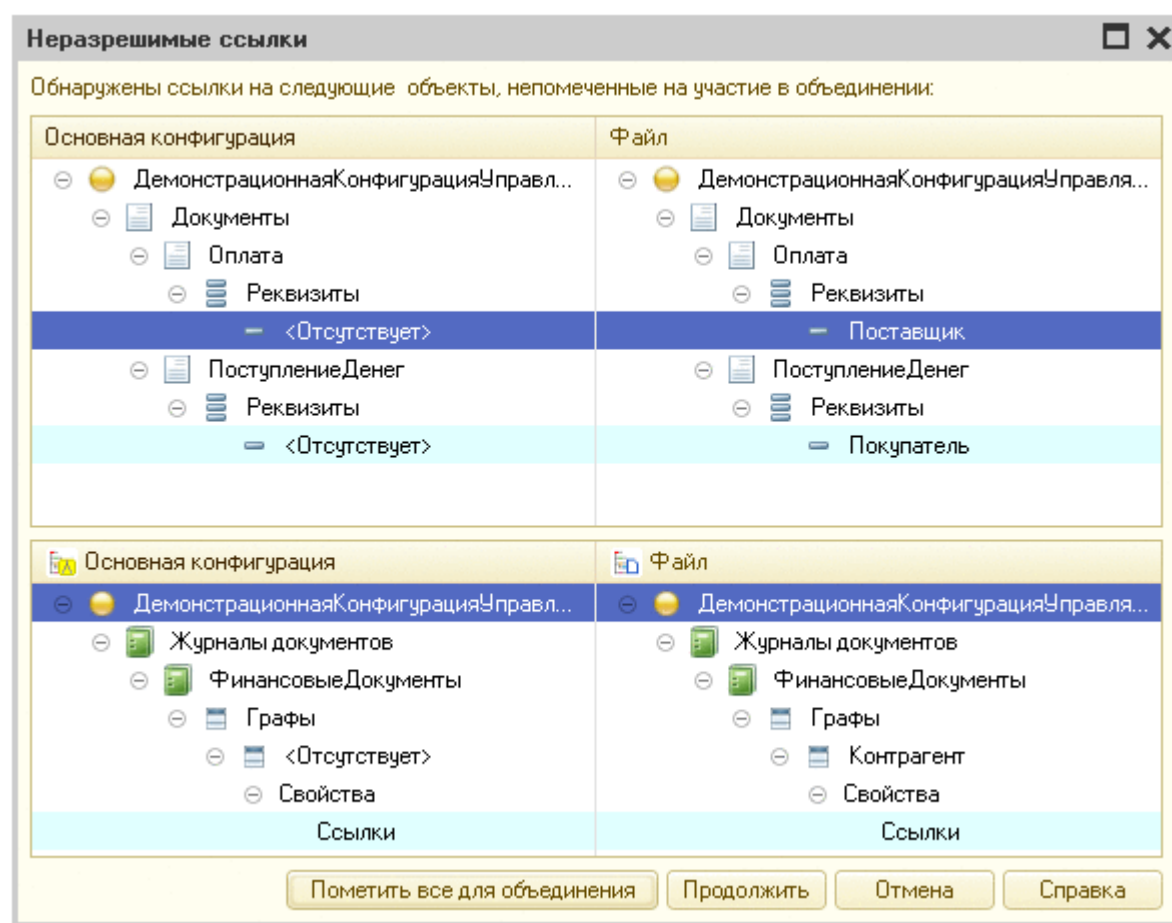


Рис. 465. Неразрешимые ссылки

Окно содержит два табличных поля. Верхнее содержит иерархический список всех найденных объектов, связанных с отмеченными объектами окна **Объединение конфигурации**. Нижнее – иерархический список объектов, использующих объект, указанный в верхнем списке.

Данное окно позволяет изучить связи выбранных объектов и принять решение о необходимости включения этих объектов в список объединяемых.

По кнопке **Пометить все для объединения** происходит установка пометок для всех объектов, необходимых для полноценного объединения. При нажатии кнопки данное окно закрывается, а в окне **Объединение конфигурации** устанавливаются пометки для всех необходимых объектов.

По кнопке **Продолжить** происходит объединение только помеченных объектов без учета наличия неразрешимых ссылок. В этом случае ответственность за результат объединения возлагается на

29.17. Загрузка конфигурации из файла

Если в процессе эксплуатации системы «1С:Предприятие» требуется полностью заменить конфигурацию, то для обновления нужно воспользоваться режимом загрузки конфигурации из файла.

Для загрузки изменений необходимо выбрать пункт [Конфигурация – Загрузить конфигурацию из файла](#) главного меню программы.

На экран будет выдан стандартный диалог открытия файла, при помощи которого следует найти и открыть новый файл конфигурации (по умолчанию [1Cv8.cf](#)).

Если текущая конфигурация не пустая, то на экран выводится вопрос: [Конфигурация не пустая. При загрузке измененной конфигурации она будет потеряна. Продолжить?](#) Если нажать кнопку [Да](#), то начнется процесс анализа различий старой и новой структур конфигурации и выполнение необходимых преобразований. На экран выдается вопрос о необходимости проведения обновления конфигурации базы данных.

Режим загрузки конфигурации из файла также используется для создания новой конфигурации, основу которой составляет загружаемая конфигурация.

Глава 3. Интерфейс приложения

Интерфейс приложения ориентирован на удобную работу пользователя и устроен таким образом, что, выполняя необходимые действия в прикладном решении, пользователь работает с различными формами. Каждая форма может быть открыта разными способами, которые будут рассмотрены далее.

Для того чтобы получить доступ к той или иной форме, следует найти в интерфейсе прикладного решения нужную команду (открытия списка, формирования отчета, ввода документа и т. д.) и выполнить ее. Процесс поиска нужной команды называется **навигацией** по прикладному решению.

Структура прикладного решения в приложении представляется пользователю в виде иерархии, которая формируется подсистемами и входящими в них объектами метаданных (см. [здесь](#)).

Формы, в которых работает пользователь, могут отображаться различными способами:

- на начальной странице,
- в отдельном окне,
- в отдельной закладке,
- в однооконном режиме – интерфейс [Такси](#).

Режим открытия форм (в отдельных окнах или в закладках) определяется отдельной настройкой клиентского приложения и начинает действовать после перезапуска клиентского приложения.

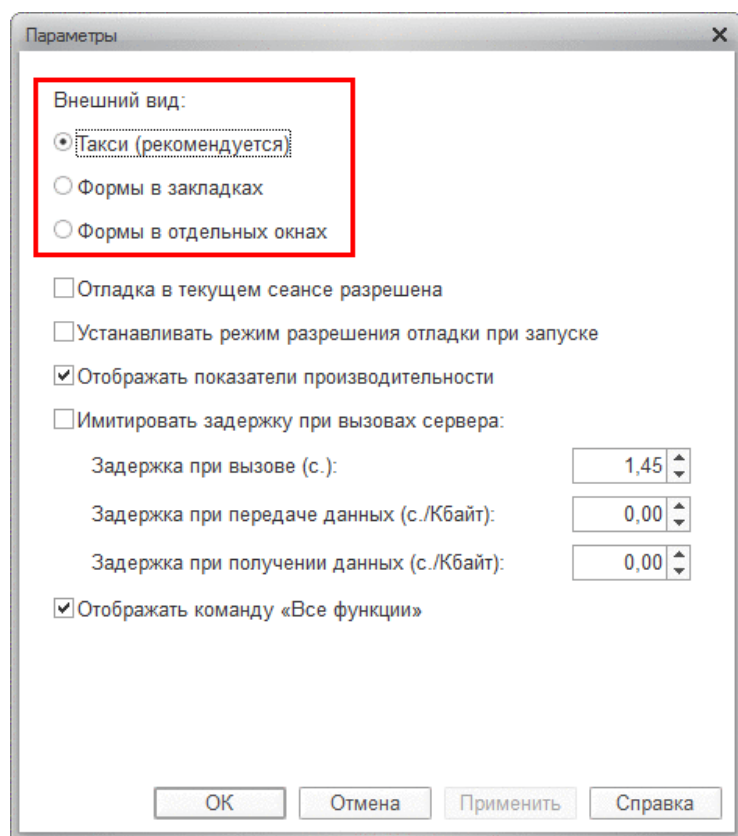


Рис. 36. Установка внешнего вида прикладного решения

Эта настройка выполняется в диалоге настройки параметров клиентского приложения, который доступен с помощью команды [Главное меню – Сервис – Параметры](#). После выбора необходимого варианта внешнего вида, система предложит перезапустить прикладное решение.

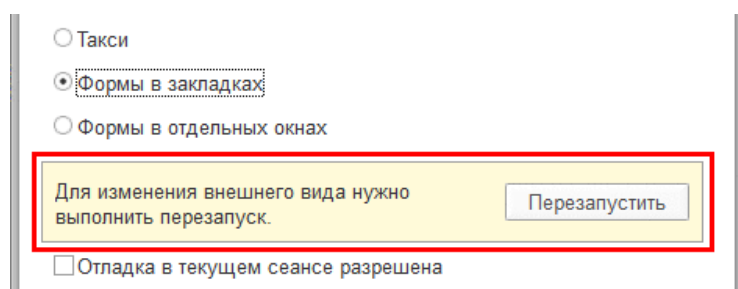


Рис. 37. Предложение перезапуска при смене внешнего вида

При нажатии кнопки [Перезапустить](#) будет произведен перезапуск прикладного решения и оно откроется в выбранном внешнем виде.

В «1С:Предприятии» существует два вида окон:

- основное окно,
- вспомогательное окно.

Каждый вид окон в «1С:Предприятии» предназначен для выполнения определенных задач. Основное окно приложения предназначено для навигации по прикладному решению и вызова различных команд. Вспомогательное окно предназначено для работы с объектами информационной базы (например, с документами или элементами справочников), построения отчетов или выполнения обработки данных. Роль вспомогательного окна может выполнять закладка, отображаемая в рабочей области основного окна.

Главное меню основных и вспомогательных окон «1С:Предприятия» содержит только общие команды, не имеющие прикладной специфики, и не может быть изменено разработчиком конфигурации.

3.1. Открытие форм в отдельных окнах

В этом режиме формы открываются во вспомогательных окнах. Каждое окно «1С:Предприятия» появляется на панели задач операционной системы и в переключателе окон по нажатию клавиш **Alt + Tab**.

3.1.1. Основное окно приложения

При работе с «1С:Предприятием» используется одно основное окно приложения. Структура основного окна организована таким образом, чтобы пользователь эффективно осуществлял навигацию по приложению, т. е. быстро находил нужные разделы и команды.

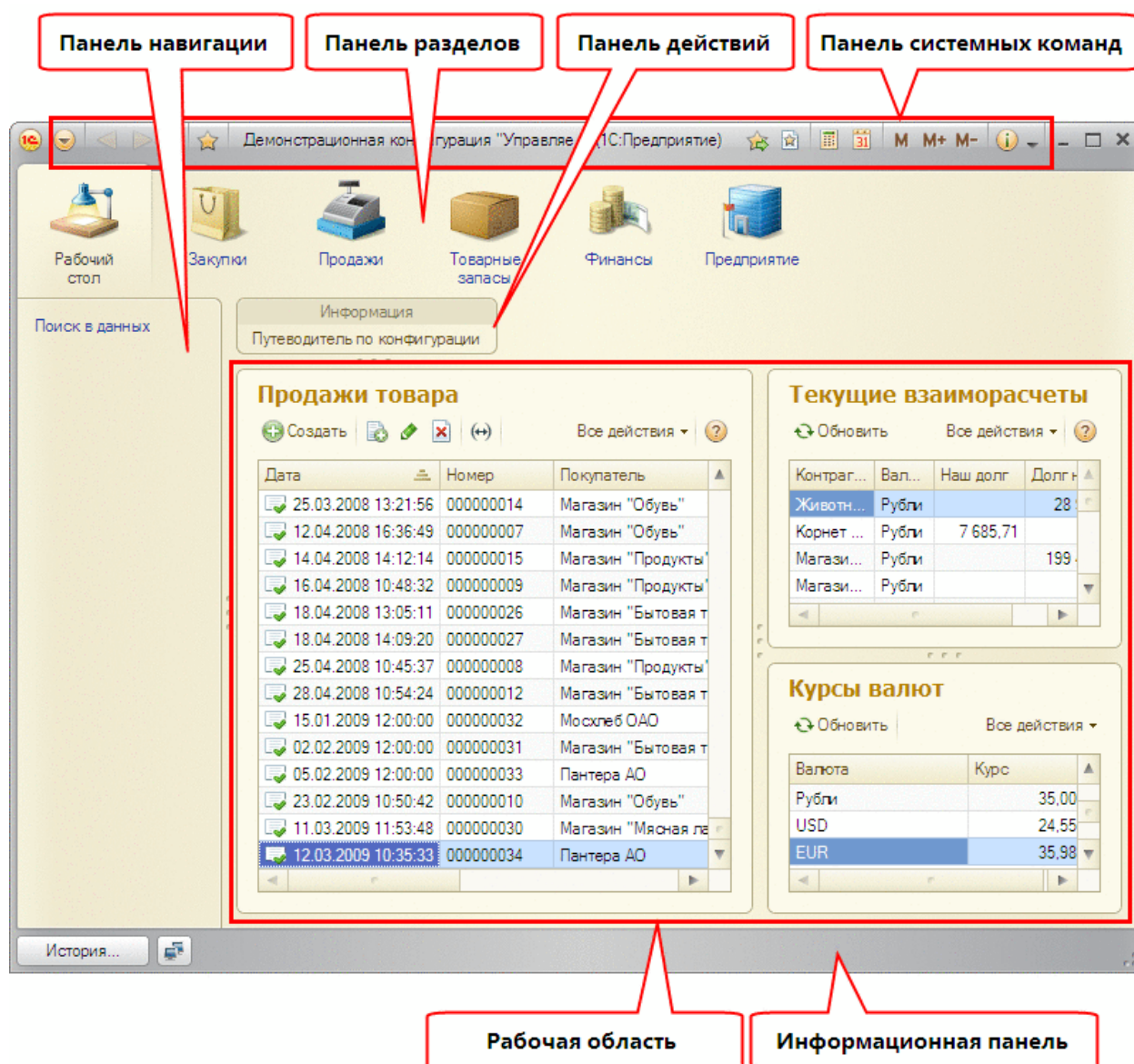


Рис. 38. Структура основного окна приложения

Каждое состояние основного окна можно рассматривать как своеобразное рабочее место. Например, перейдя в раздел **Товарные запасы**, пользователь увидит набор команд для выполнения наиболее частых действий, связанных с управлением товарными запасами.

3.1.1.1. Панель разделов

Панель разделов показывает список подсистем верхнего уровня и позволяет быстро переключаться между ними. Каждый раздел

соответствует подсистеме (например, [Продажи](#), [Закупки](#), [Запасы](#)). Для повышения наглядности каждой подсистеме можно установить понятную картинку. Отсутствие картинки не препятствует отображению подсистемы в панели разделов. Первым разделом всегда является основной раздел, который называется [Рабочий стол](#). На рабочем столе собраны самые необходимые и часто используемые инструменты прикладного решения. Остальные разделы соответствуют подсистемам первого уровня иерархии.

Панель разделов имеет возможность горизонтальной прокрутки, однако с помощью прав доступа можно ограничить набор разделов так, что прокрутка не потребуется.

ПРИМЕЧАНИЕ. Если в панели разделов нет ни одного раздела (они недоступны или скрыты пользователем), то панель разделов автоматически скрывается и основное окно приложения автоматически переключается на рабочий стол конфигурации.

3.1.1.2. Панель навигации

Панель навигации отображает структуру конфигурации в соответствии с разделом, выбранным в панели разделов. Если у подсистемы имеются подчиненные подсистемы, то они будут отображаться как группы с возможностью сворачивания. В дальнейшем подсистемы, подчиненные подсистемам верхнего уровня, мы будем называть подразделами.

При нажатии на гиперссылки этой панели, как правило, происходит открытие форм списков. При этом формы открываются непосредственно в основном окне, замещая друг друга.

Щелчок по гиперссылке приводит к открытию формы в рабочей области основного окна приложения.

В панели навигации могут располагаться команды трех стандартных групп: [Важное](#), [Обычное](#) и [См. также](#). Команды группы [Важное](#) будут выделены **полужирным** шрифтом. Если какой-либо группы нет, то она пропускается. Если разработчик конфигурации определил собственные группы команд из категории [Панель навигации](#), то они будут располагаться непосредственно перед группой [См. также](#) (которая всегда является самой последней группой в панели навигации).

При отображении группы [См. также](#) действуют следующие правила:

- команды группы [См. также](#) из отображаемого раздела и всех его подразделов визуальнo размещаются в одном списке [См. также](#);
- команды выводятся в порядке разделов и подразделов, который задан при конфигурировании;
- внутри разделов и подразделов команды выводятся в порядке, который задан при конфигурировании.

Порядок разделов (подсистем верхнего уровня), подразделов (подчиненных подсистем) и команд в группе [См. также](#) панели навигации можно изменять в редакторе фрагмента командного интерфейса.

3.1.1.3. Панель действий

[Панель действий](#) содержит команды, которые соответствуют текущему разделу, выбранному в панели разделов. Эти команды объединены в стандартные группы: [Создать](#), [Отчеты](#), [Сервис](#) и группы, созданные разработчиком.

Группа [Создать](#) включает в себя команды создания новых объектов информационной базы, например, документов или элементов справочников. В этой группе размещаются команды создания для всех объектов, входящих в подсистему, однако по умолчанию видимость этих команд отключена. Разработчик должен включить видимость для наиболее часто используемых команд создания новых объектов. Группа [Отчеты](#) содержит команды открытия отчетов, а группа [Сервис](#) – команды открытия обработок. Если в какой-либо группе нет ни одной команды, то группа не показывается. Если разработчик конфигурации определил собственные группы команд, принадлежащие категории [Панель действий](#), то они располагаются после группы [Сервис](#).

При отображении групп панели действий действуют следующие правила:

- команды одинаковых групп из отображаемого раздела и всех его подразделов визуальнo размещаются в одном списке;
- команды выводятся в порядке разделов и подразделов, который задан при конфигурировании;
- внутри разделов и подразделов команды выводятся в порядке, который задан при конфигурировании.

Порядок разделов (подсистем верхнего уровня), подразделов (подчиненных подсистем) и команд в группах панели действий можно изменять в редакторе командного интерфейса.

Высота панели действий по умолчанию зависит от количества команд в группах и не превышает трех строк при автоматическом определении высоты панели. Однако пользователь может произвольно изменять высоту с помощью разделителя, расположенного под панелью. Если при текущей высоте панели действий в ней не помещаются все команды какой-либо группы, то в правом нижнем углу группы будет размещена пиктограмма, по нажатию на которую будет раскрыто меню, содержащее все команды группы.

3.1.1.4. Информационная панель

Информационная панель предназначена для обращения к последним данным, редактируемым пользователем, а также для отображения перечня последних оповещений (информация о тех или иных действиях, выполненных в системе). Подробнее см. [здесь](#).

3.1.1.5. Область системных команд

Область системных команд позволяет выполнять ряд действий, не зависящих от прикладной специфики конфигурации, но помогающих в управлении системой. Кроме того, в этой области расположены различные команды, помогающие в работе с окном.



Рис. 39. Левая часть области системных команд

В левой части области системных команд расположена кнопка вызова главного меню, а также команды перехода по точкам навигации и меню работы с избранным пользователем. **Точка навигации** – это фрагмент интерфейса основного окна приложения, в который можно перейти с помощью команд, расположенных в основном окне приложения.

В «1С:Предприятии» избранное – это список ссылок, специально отобранных самим пользователем для быстрого перехода к тем или иным разделам конфигурации, точкам навигации, формам объектов информационной базы, а также формам отчетов и обработок.

Существует возможность программного управления списком избранного. Для этого используется объект *ИзбранноеРаботыПользователя*. Данный объект можно получить из хранилища системных настроек.

ПРИМЕЧАНИЕ. При работе с избранным следует помнить, что список избранного является одноуровневым.

В список избранного нельзя добавить ссылку на стандартную функцию (вызываемую с помощью команды [Главное меню – Все функции – Стандартные](#)).

Во вспомогательных окнах область системных команд выглядит другим образом (см. [здесь](#)).

В правой части области системных команд расположены различные вспомогательные команды, которые помогают в работе с окном. Перечень команд может быть настроен с помощью соответствующего меню настройки.

3.1.2. Вспомогательное окно

Вспомогательным называется окно, которое открывается для выполнения какого-либо действия, а не для навигации по всему приложению в целом. В таких окнах, например, открываются формы документов или элементов справочников, формы отчетов и обработок.

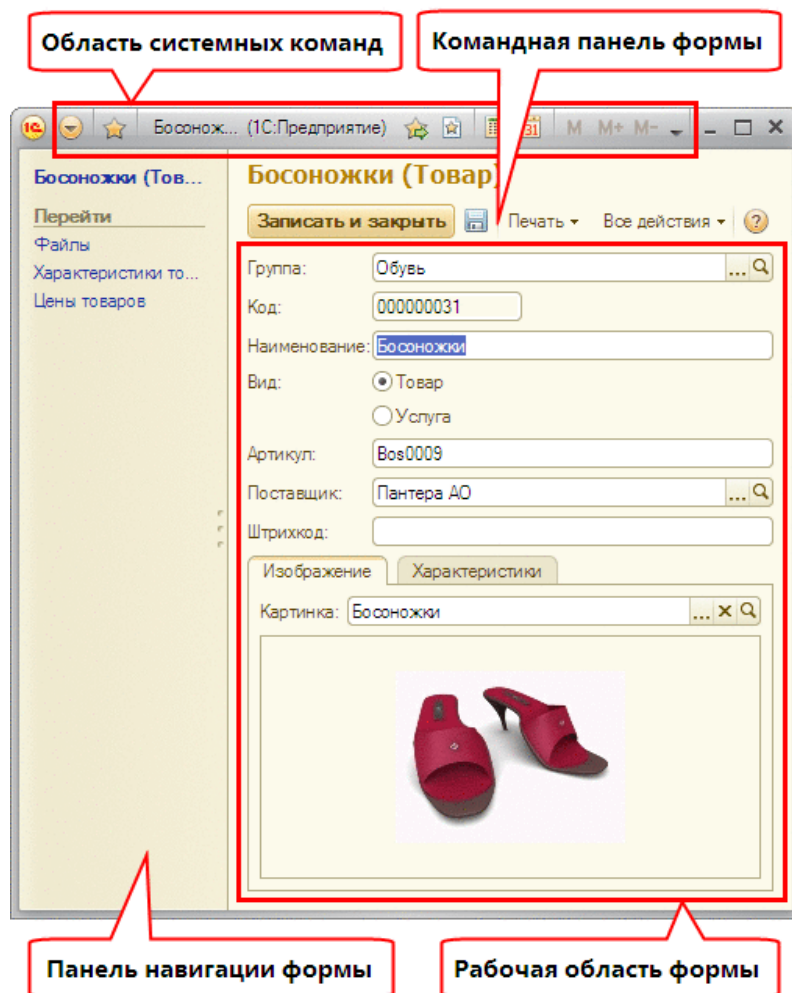


Рис. 40. Структура вспомогательного окна приложения

Заккрытие вспомогательного окна не приводит к закрытию всего приложения.

Вспомогательных окон может быть открыто сколько угодно, но для каждого объекта информационной базы (например, документа или элемента справочника) оно одно, т. е. нельзя открыть несколько окон для просмотра одного и того же документа, если это специально не предусмотрено в конфигурации.

Во вспомогательном окне можно выделить **основную форму** – форму, которая отображается по умолчанию при открытии вспомогательного окна и которая открывается при нажатии на первую ссылку панели навигации формы. При нажатии на другие ссылки панели навигации формы будут открываться **вспомогательные формы**.

3.1.2.1. Панель навигации формы

Панель навигации вспомогательного окна (панель навигации формы) позволяет просматривать различные сведения, логически связанные с данными, которые отображает основная форма окна.

В панели навигации могут располагаться команды трех групп: **Важное**, **Перейти** и **См. также**. Команды, относящиеся к группам **Важное** и **Перейти**, будут объединены общим заголовком **Перейти**, при этом команды группы **Важное** располагаются первыми, выделены полужирным шрифтом и отделены от остальных команд, расположенных под заголовком **Перейти**, небольшим отступом. Если какой-либо группы нет, то она пропускается. Если разработчик конфигурации определил собственные группы команд из категории **Панель навигации формы**, то они будут располагаться непосредственно перед группой **См. также** (которая всегда является самой последней группой в панели навигации формы).

Существует возможность возврата к основной форме. Для этого нужно нажать гиперссылку в верхней части панели навигации (выделена **полужирным** шрифтом).

3.1.2.2. Командная панель формы

Командная панель формы содержит команды, непосредственно связанные с объектом, который отображается в основной форме. Существуют стандартные группы **Важное** и **Создать на основании**; кроме того, на панели могут размещаться группы, созданные разработчиком. Каждая команда из группы **Важное** будет отображаться своей кнопкой на панели. Группа **Создать на основании** будет оформлена в виде подменю панели. Также в виде подменю будут оформлены все группы команд, созданные разработчиком. Команды формы, такие как **Сохранить и закрыть**, будут расположены перед командами группы **Важное**.

Если в какой-либо группе нет ни одной команды, то группа не показывается. Если разработчик конфигурации добавил в эту область собственные группы, то они располагаются после группы **Создать на основании**.

3.1.2.3. Область системных команд

Левая часть области системных команд вспомогательного окна содержит только команду открытия главного меню и команды работы с избранным. В правой части расположены различные вспомогательные команды, которые помогают в работе с окном (например,

команды получения и перехода по ссылке, вызова калькулятора, календаря и т. д.). Правая часть области может быть настроена с помощью соответствующего меню настройки.

3.2. Открытие форм в закладках

В этом режиме формы открываются в качестве закладок рабочей области основного окна. Некоторые формы по-прежнему открываются в отдельных окнах (например, окна выбора), но эти окна не появляются на панели задач.

Форму, открытую в закладке, можно открыть во вспомогательном окне, если этого требует логика работы. В этом случае вспомогательное окно будет отображаться на панели задач операционной системы и будет доступно при переключении окон по **Alt + Tab**.

3.2.1. Основное окно приложения

При работе с «1С:Предприятием» используется одно основное окно приложения. Структура основного окна организована таким образом, чтобы пользователь эффективно осуществлял навигацию по приложению, т. е. быстро находил нужные разделы и команды.

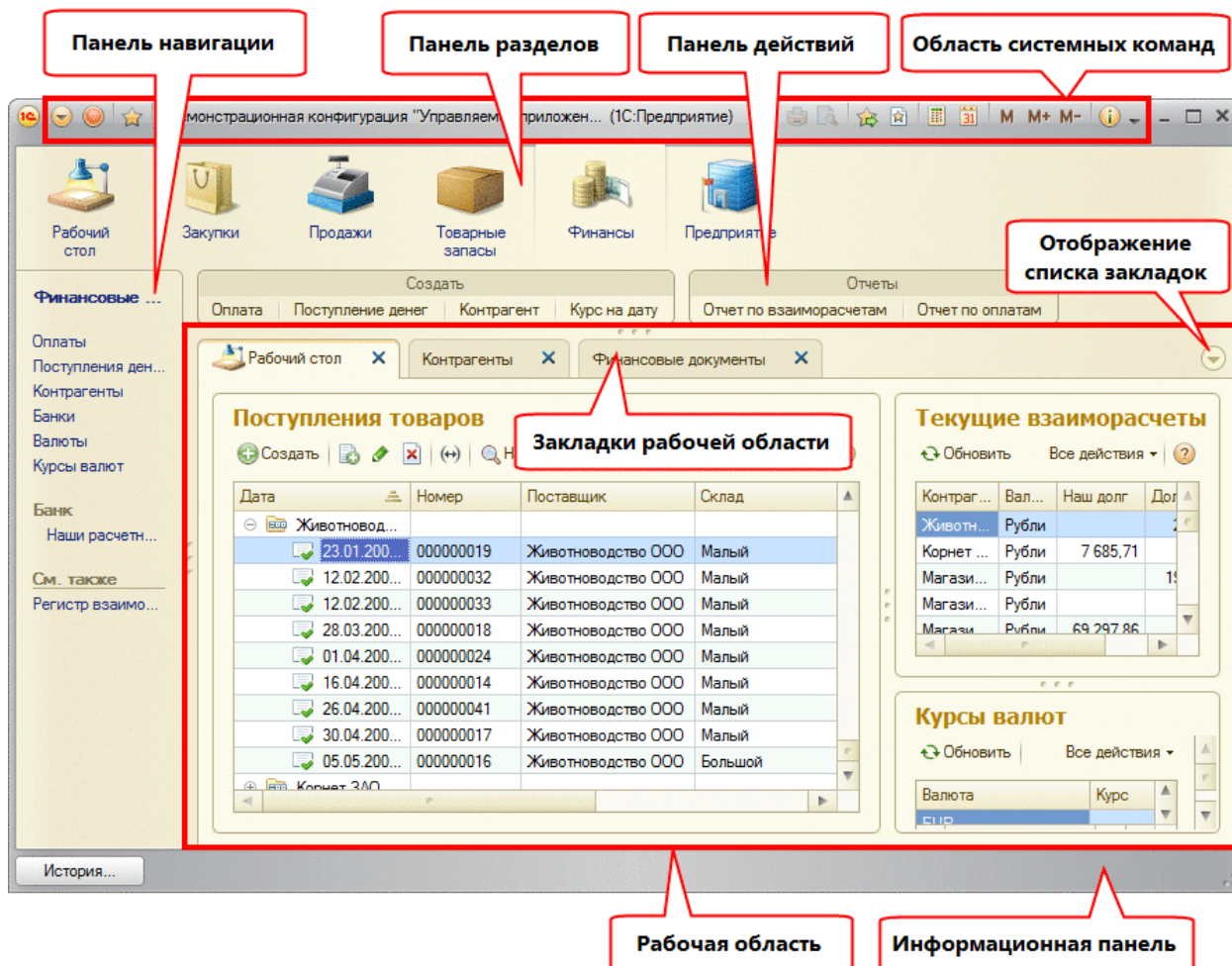


Рис. 41. Основное окно с закладками

Каждое состояние основного окна можно рассматривать как своеобразное рабочее место. Например, перейдя в раздел **Финансы**, пользователь увидит набор команд для выполнения наиболее частых действий, связанных с финансовыми операциями.

Каждая закладка, отображаемая в рабочей области основного окна, может быть закрыта (включая закладку **Рабочий стол**). Это можно сделать или с помощью контекстного меню (команда **Закрыть**), или нажав на синюю картинку в правой части каждой закладки.

ПРИМЕЧАНИЕ. Закладка **Рабочий стол** всегда располагается первой среди закладок.

Любую закладку (кроме закладки **Рабочий стол**) можно также открыть во вспомогательном окне. Для этого нужно воспользоваться командой **Открыть в отдельном окне** контекстного меню закладки.

Заголовок закладки изменяет размер в зависимости от количества закладок в основном окне. При увеличении количества закладок размер заголовка закладки уменьшается, и заголовок может оказаться нечитаемым. В этом случае для получения списка открытых закладок можно воспользоваться специальной кнопкой, расположенной справа от закладок.

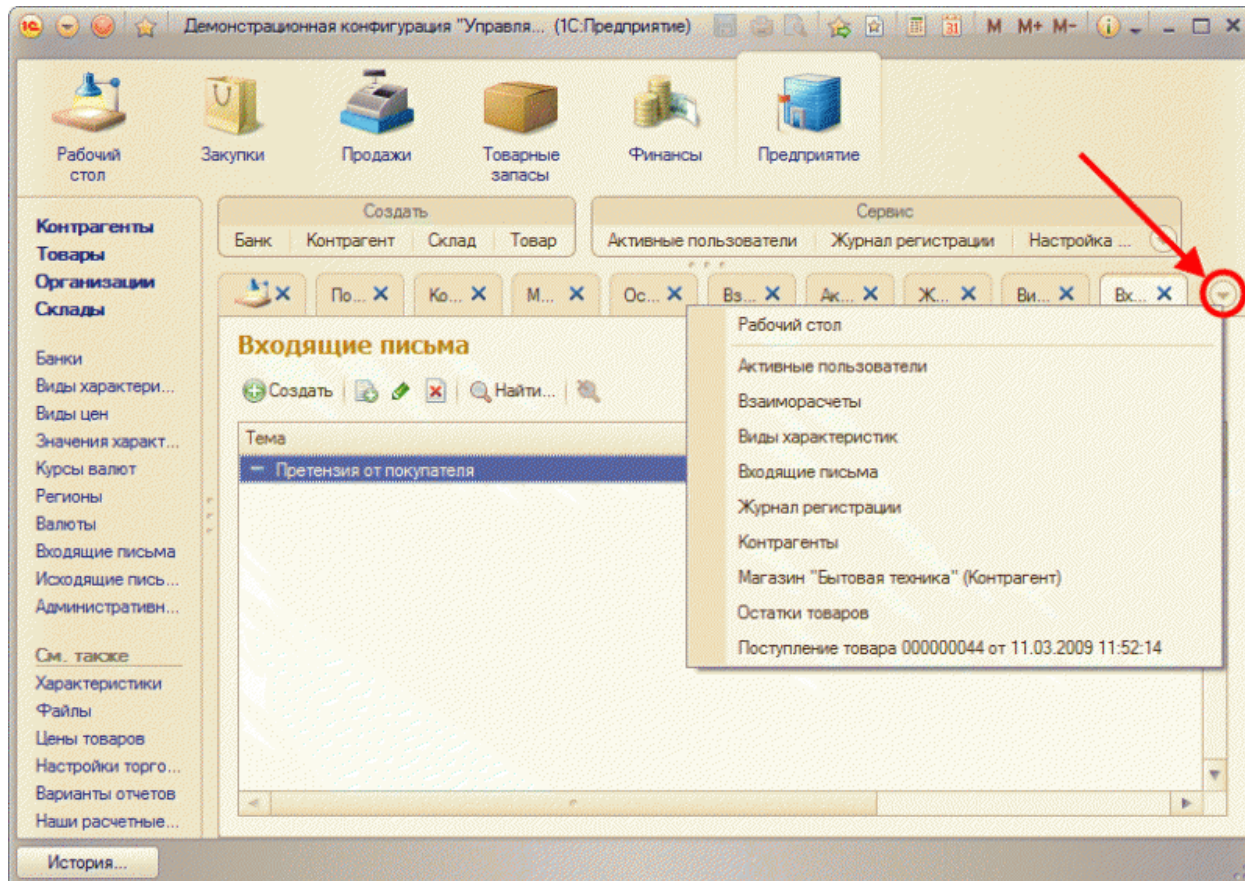


Рис. 42. Список закладок

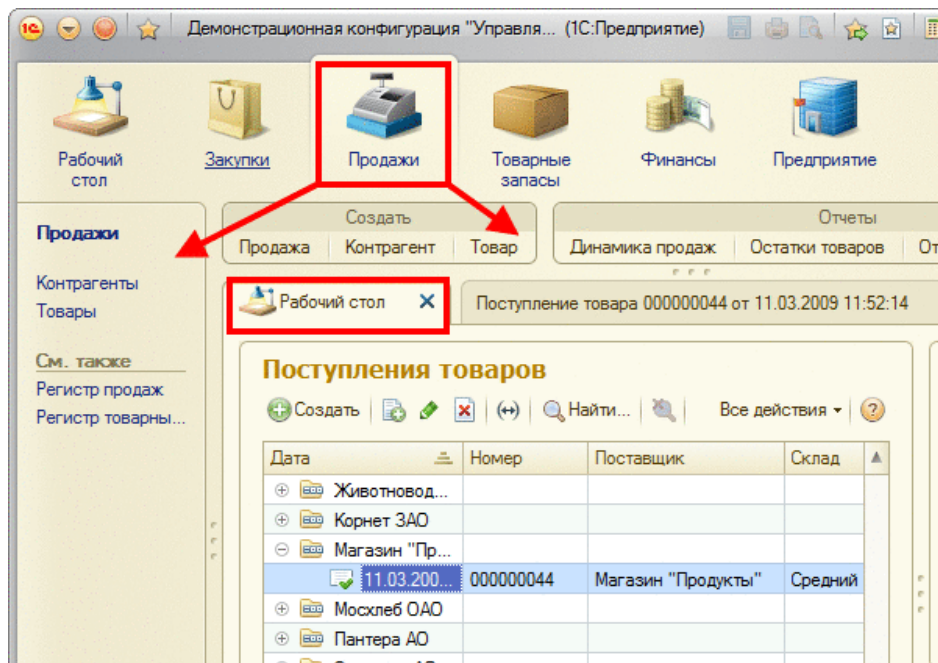
Если в форме, открытой в закладке, выполняется открытие блокирующей (например, формы выбора) или модальной формы, то такая форма не отображается отдельной закладкой и не присутствует в списке окон на панели задач. Такая форма блокирует содержимое только той закладки, в которой такая блокирующая форма была открыта. При этом остается возможность переключаться между закладками. В случае переключения на закладку с открытой блокирующей формой она будет отображена на экране.

3.2.1.1. Панель разделов

Панель разделов показывает список подсистем верхнего уровня и позволяет быстро переключаться между ними. Каждый раздел соответствует подсистеме (например, [Продажи](#), [Закупки](#), [Запасы](#)). Для повышения наглядности каждой подсистеме можно установить понятную картинку. Отсутствие картинки не препятствует отображению подсистемы в панели разделов. Первым разделом всегда является [Рабочий стол](#). На рабочем столе собраны самые необходимые и часто используемые инструменты прикладного решения. Остальные разделы соответствуют подсистемам первого уровня иерархии.

Панель разделов имеет возможность горизонтальной прокрутки, однако с помощью прав доступа можно ограничить набор разделов так, что прокрутка не потребуется.

При переходе по разделам не меняется информация, отображаемая в рабочей области, но меняется содержимое панели навигации и панели действий.



ПРИМЕЧАНИЕ. Если в панели разделов нет ни одного раздела (они недоступны или скрыты пользователем), то панель разделов автоматически скрывается и основное окно приложения автоматически переключается на рабочий стол конфигурации.

3.2.1.2. Панель навигации

Панель навигации отображает структуру конфигурации в соответствии с разделом, выбранным в панели разделов. Если у подсистемы имеются подчиненные подсистемы, то они будут отображаться как группы с возможностью сворачивания. В дальнейшем подсистемы, подчиненные подсистемам верхнего уровня, мы будем называть подразделами.

При нажатии на гиперссылки этой панели, как правило, происходит открытие форм списков. При этом формы открываются непосредственно в основном окне, замещая друг друга.

Щелчок по гиперссылке приводит к открытию формы в рабочей области основного окна приложения.

В панели навигации могут располагаться команды трех стандартных групп: **Важное**, **Обычное** и **См. также**. Команды группы **Важное** будут выделены **полужирным** шрифтом. Если какой-либо группы нет, то она пропускается. Если разработчик конфигурации определил собственные группы команд из категории **Панель навигации**, то они будут располагаться непосредственно перед группой **См. также** (которая всегда является самой последней группой в панели навигации).

При отображении группы **См. также** действуют следующие правила:

- команды группы **См. также** из отображаемого раздела и всех его подразделов визуальнo размещаются в одном списке **См. также**;
- команды выводятся в порядке разделов и подразделов, который задан при конфигурировании;
- внутри разделов и подразделов команды выводятся в порядке, который задан при конфигурировании.

Порядок разделов (подсистем верхнего уровня), подразделов (подчиненных подсистем) и команд в группе **См. также** панели навигации можно изменять в редакторе фрагмента командного интерфейса.

3.2.1.3. Панель действий

Панель действий содержит команды, которые соответствуют текущему разделу, выбранному в панели разделов. Эти команды объединены в стандартные группы: **Создать**, **Отчеты**, **Сервис** и группы, созданные разработчиком.

Группа **Создать** включает в себя команды создания новых объектов информационной базы, например, документов или элементов справочников. В этой группе размещаются команды создания для всех объектов, входящих в подсистему, однако по умолчанию видимость этих команд отключена. Разработчик должен включить видимость для наиболее часто используемых команд создания новых объектов. Группа **Отчеты** содержит команды открытия отчетов, а группа **Сервис** – команды открытия обработок. Если в какой-либо группе нет ни одной команды, то группа не показывается. Если разработчик конфигурации определил собственные группы команд, принадлежащие категории **Панель действий**, то они располагаются после группы **Сервис**.

При отображении групп панели действий действуют следующие правила:

- команды одинаковых групп из отображаемого раздела и всех его подразделов визуальнo размещаются в одном списке;
- команды выводятся в порядке разделов и подразделов, который задан при конфигурировании;
- внутри разделов и подразделов команды выводятся в порядке, который задан при конфигурировании.

Порядок разделов (подсистем верхнего уровня), подразделов (подчиненных подсистем) и команд в группах панели действий можно изменять в редакторе командного интерфейса.

Высота панели действий по умолчанию зависит от количества команд в группах и не превышает трех строк при автоматическом определении высоты панели. Однако пользователь может произвольно изменять высоту с помощью разделителя, расположенного под панелью. Если при текущей высоте панели действий в ней не помещаются все команды какой-либо группы, то в правом нижнем углу группы будет размещена пиктограмма, по нажатию на которую будет раскрыто меню, содержащее все команды группы.

3.2.1.4. Информационная панель

Информационная панель предназначена для обращения к последним данным, редактируемым пользователем, а также для отображения перечня последних оповещений (информация о тех или иных действиях, выполненных в системе). Подробнее см. [здесь](#).

3.2.1.5. Область системных команд

Область системных команд позволяет выполнять ряд действий, не зависящих от прикладной специфики конфигурации, но помогающих в управлении системой. Кроме того, в этой области расположены различные команды, помогающие в работе с окном.

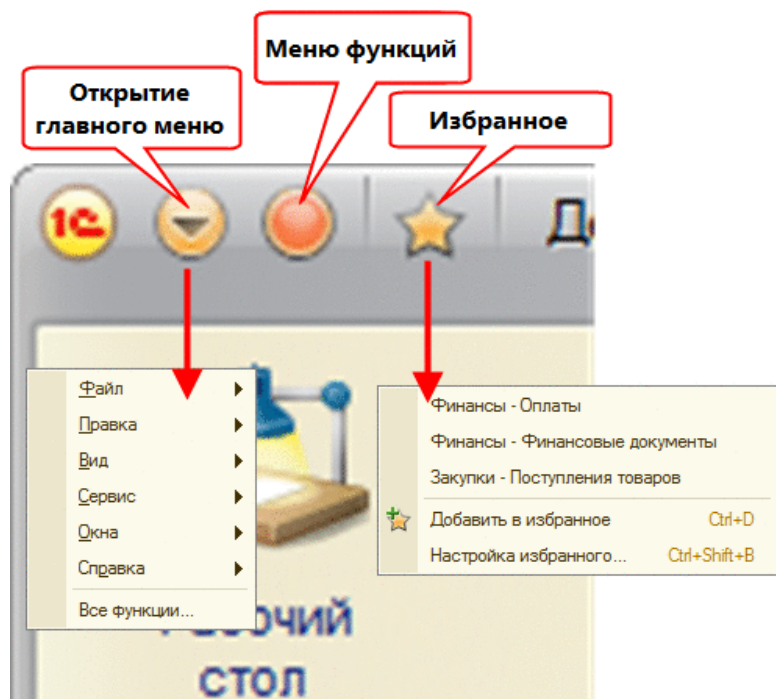


Рис. 44. Левая часть области системных команд

В левой части области системных команд расположена кнопка вызова главного меню, а также команды вызова меню функций и меню работы с избранным пользователя.

В главном меню расположено меню **Вид**, которое позволяет настроить внешний вид окна приложения.

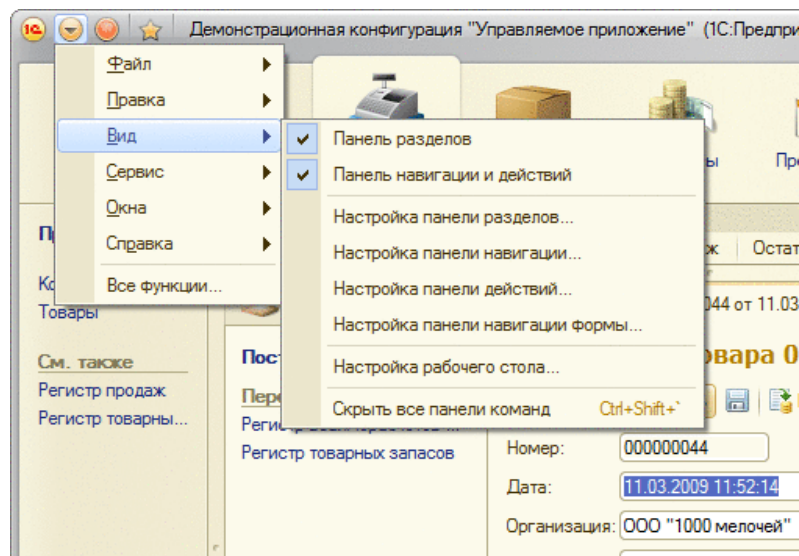


Рис. 45. Меню «Вид»

В частности, данное меню позволяет отключать видимость панели разделов, панелей навигации и действий. Это позволит увеличить место под формы и уменьшить вероятность появления полос прокрутки. Также следует обратить внимание на команду **Скрыть все панели команд**. После выполнения команды скрывается панель разделов, панель навигации и панель действий.

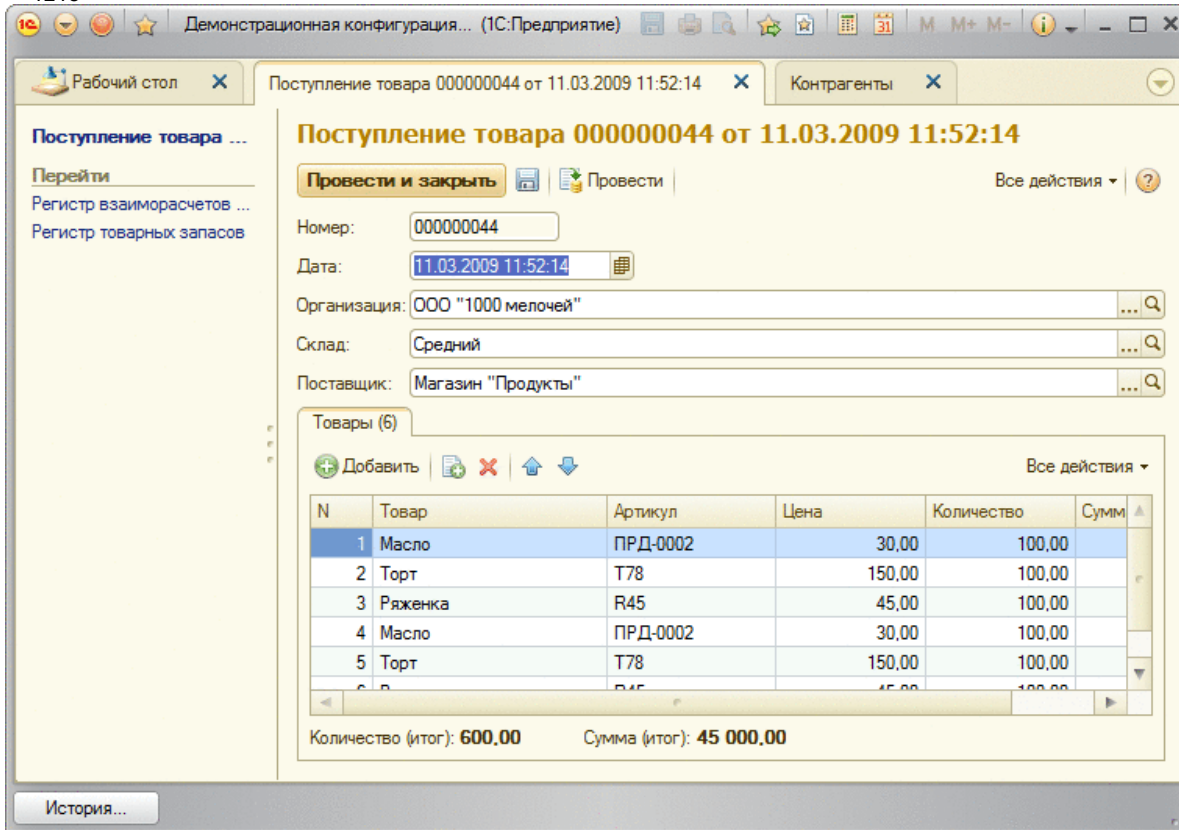


Рис. 46. Скрытые панели команд

Повторное выполнение команды возвращает скрытые панели. Нужно помнить следующую особенность: команда [Скрыть все панели команд](#) не будет работать, если панель разделов, панель навигации и панель действий скрыты с помощью меню [Вид](#).

Для ускорения и упрощения доступа к командам той или иной подсистемы служит специальный режим – [Меню функций](#). Для вызова этого режима следует нажать специальную кнопку в области системных команд.



Рис. 47. Меню функций

Меню функций представляет собой панель разделов и специальную форму, которая содержит все команды выбранного раздела. Перечень разделов и состав команд соответствуют текущим настройкам пользователя. Первая колонка формы соответствует панели навигации, остальные колонки – группам панели действий выбранной подсистемы. Отображение панели разделов не зависит от того, в каком состоянии находится настройка [Главное меню – Вид – Панель разделов](#). После выбора необходимой команды основное окно

возвращается к тому виду, какой был до вызова меню функций. Состав команд обновляется, если в окне функций выбрать другой раздел.

В «1С:Предприятии» **избранное** – это список ссылок, специально отобранных самим пользователем для быстрого перехода к тем или иным разделам конфигурации, точкам навигации, формам объектов информационной базы, а также формам отчетов и обработок.

Существует возможность программного управления списком избранного. Для этого используется объект `ИзбранноеРаботыПользователя`. Данный объект можно получить из хранилища системных настроек.

ПРИМЕЧАНИЕ. При работе с избранным следует помнить, что список избранного является одноуровневым.

В список избранного нельзя добавить ссылку на стандартную функцию (вызываемую с помощью команды **Главное меню – Все функции – Стандартные**).

Во вспомогательных окнах область системных команд выглядит другим образом (см. [здесь](#)).

В меню **Окна** отображается основное окно, закладки окна и вспомогательные окна. Флажком отмечается:

- активное вспомогательное окно;
- активная закладка основного окна;
- собственно основное окно, если в нем не открыто ни одной закладки.

В правой части области системных команд расположены различные вспомогательные команды, которые помогают в работе с окном. Перечень команд может быть настроен с помощью соответствующего меню настройки. Доступность и состав команд зависят от текущей формы в активной закладке.

3.2.2. Вспомогательное окно

Описание вспомогательного окна см. [здесь](#).

3.3. Интерфейс «Такси»

В этом режиме формы в основном открываются в рабочей области основного окна. Некоторые формы по-прежнему открываются в отдельных окнах (например, окна выбора), но эти окна не появляются на панели задач. Переключение между окнами с помощью сочетания клавиш **Alt+Tab** невозможно. Можно переключаться между открытыми формами с помощью специальных кнопок, присутствующих в заголовке всех форм, которые открыты в рабочей области основного окна приложения.

3.3.1. Основное окно приложения

При работе с «1С:Предприятием» используется одно основное окно приложения. Структура основного окна организована таким образом, чтобы пользователь эффективно осуществлял навигацию по приложению, т. е. быстро находил нужные разделы и команды. В основном окне приложения не выполняется открытия форм (кроме рабочей области).

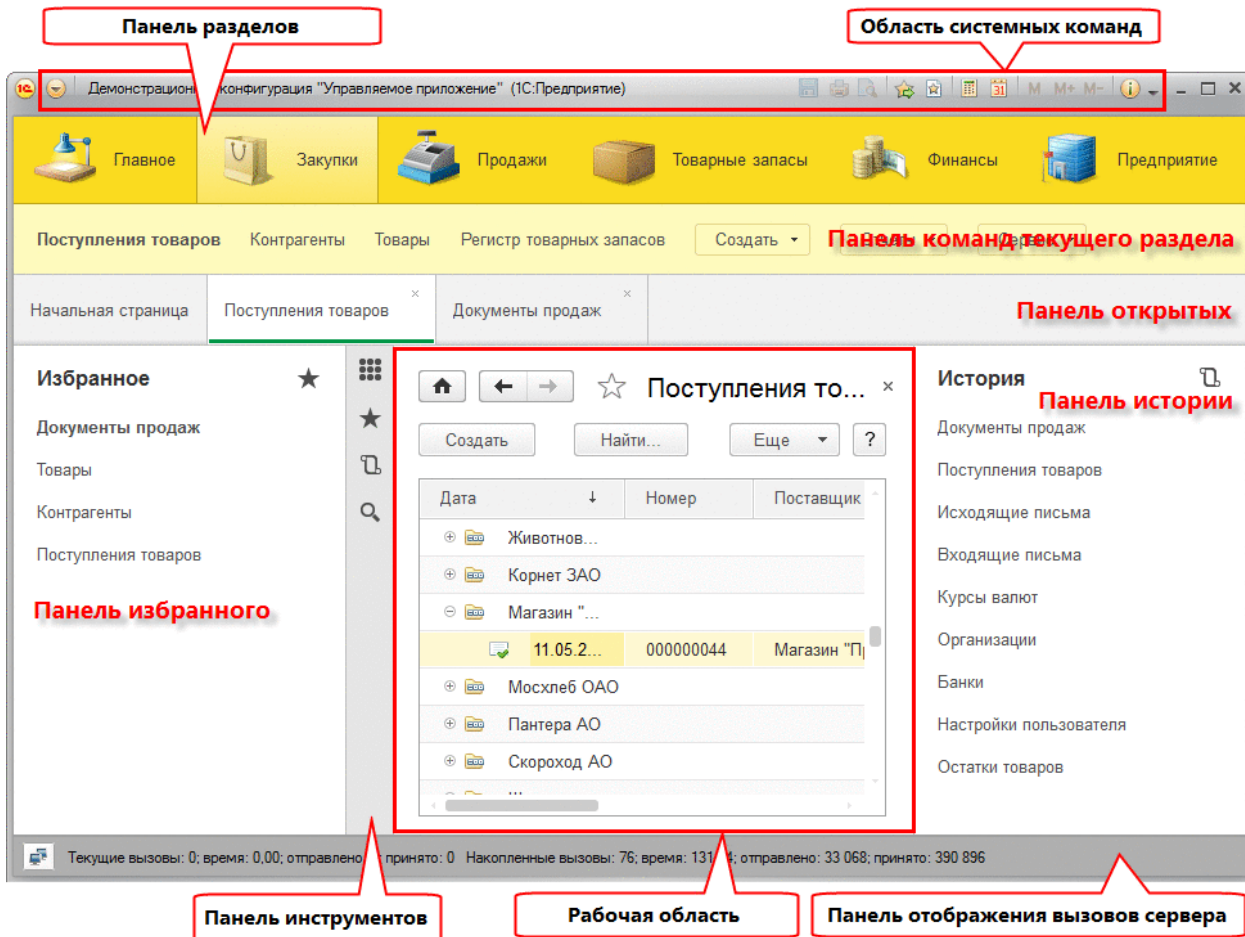


Рис. 48. Основное окно в интерфейсе «Такси»

При работе в интерфейсе **Такси**, в основном окне приложения могут произвольно располагаться несколько различных панелей (см. [здесь](#)). В рабочей области основного окна приложения отображаются все формы, с которыми работает пользователь. Исключением являются блокирующие формы, которые открываются в отдельных окнах. Если в форме, открытой в рабочей области, выполняется открытие блокирующей (например, формы выбора) формы, то такая форма отображается в виде отдельного окна, но не присутствует в списке окон на панели задач. В зависимости от режима открытия окна (см. [здесь](#)), будет заблокирована работа либо с формой-владельцем, либо со всем интерфейсом системы. При этом если вспомогательное окно открыто в режим блокировки окна-владельца, остается возможность открывать другие формы и использовать командный интерфейс прикладного решения. В случае переключения на форму, из которой открыта форма, блокирующая работу с владельцем, блокирующая форма будет отображена на экране.

Рассмотрим более подробно составные части основного окна приложения.

3.3.1.1. Начальная страница

Работа с прикладным решением начинается с начальной страницы.

Начальная страница

Поступления товаров

Создать Найти... Отменить поиск Создать на основании Еще ?

Дата	Номер	Поставщик	Склад	Организация	Валюта вза
Шлюзовая...					
ЭлектроБ...					

Продажи товара

Создать Найти... Отменить поиск Создать на основании Печать Еще ?

Дата	Номер	Организация	Покупатель	Склад
06.09.2012 9:...	000000015	ООО "Товары"	Шлюзовая ООО	Средний
10.09.2012 14:...	000000016	ООО "Товары"	Магазин "Продукты"	Малый
12.09.2012 15:...	000000017	ООО "Все для дома"	Попов Б.В. ИЧП	Малый

Рис. 49. Начальная страница

На этой странице прикладной разработчик располагает формы, которые наиболее востребованы (по его замыслу) в данном прикладном решении. Доступность форм на этой странице определяется правами доступа и функциональными опциями. Начальную страницу невозможно закрыть.

Настройка интерфейса начальной страницы выполняется с помощью свойств конфигурации (см. [здесь](#)). Пользователь также может сформировать персональный состав начальной страницы из списка доступных форм, которые прикладной разработчик разрешил размещать на начальной странице. Для этого необходимо воспользоваться специальным диалогом настройки начальной страницы: [Главное меню – Вид – Настройка начальной страницы](#).

3.3.1.2. Панель разделов

Панель разделов показывает список подсистем верхнего уровня и позволяет быстро выбирать нужный набор функций. Каждый раздел соответствует подсистеме (например, [Продажи](#), [Закупки](#), [Запасы](#)). Для повышения наглядности каждой подсистеме можно установить понятную картинку. Отсутствие картинки не препятствует отображению подсистемы в панели разделов. Первым разделом всегда является основной раздел, который называется [Главное](#). В основном разделе собраны самые необходимые и часто используемые инструменты прикладного решения. Остальные разделы соответствуют подсистемам первого уровня иерархии.

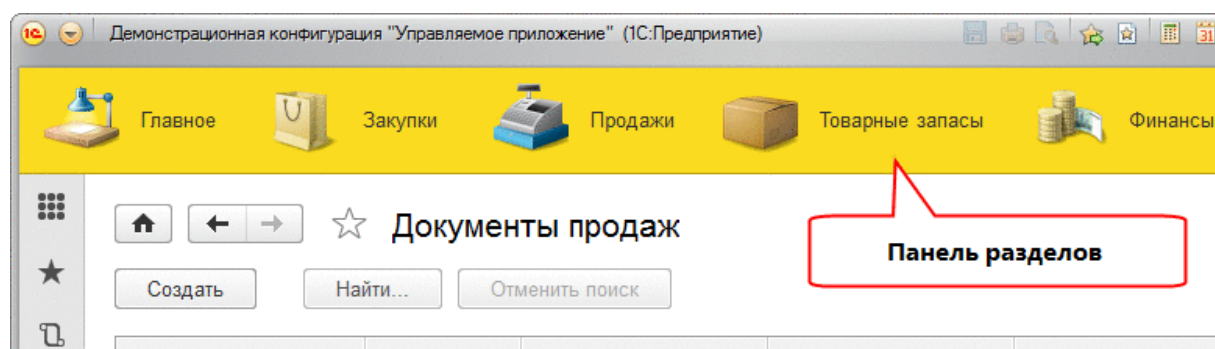


Рис. 50. Панель разделов

Фактически, элемент панели разделов группирует команды системы, логически связанные с каким-либо направлением работы в рамках прикладного решения. Например, действия, связанные с управлением закупками (раздел [Закупки](#)), управлением товарными запасами (раздел [Товарные запасы](#)) и т. д. Панель разделов не является инструментом переключения автоматизированных рабочих мест.

Панель разделов имеет возможность прокрутки. Состав панели разделов для конкретного пользователя определяется его правами доступа и установленными функциональными опциями. Видимость доступных разделов может регулироваться самим пользователем с помощью команды [Главное меню – Вид – Настройка панели разделов](#).

Действие при выборе раздела зависит от того, отображается в главном окне панель команд текущего раздела или нет (см. [здесь](#)):

- Панель не отображается. В этом случае выбор какого-либо раздела приводит к отображению на экране меню функций данного раздела. В этом случае текущий раздел никак не отображается на экране (когда закрыто меню функций).

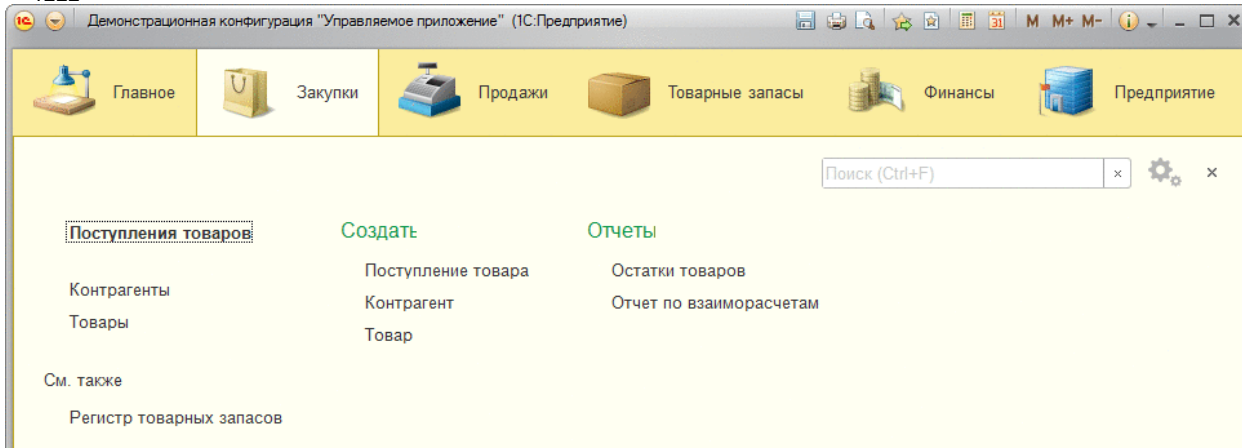


Рис. 51. Меню функций при выборе раздела

- Панель отображается. В этом случае при выборе раздела не будет открыто меню функций, но будет изменено содержимое панели команд текущего раздела. Если отображаемой информации не достаточно, то можно открыть меню функций повторным выбором раздела. Также в этом случае на панели разделов всегда отображается текущий раздел.

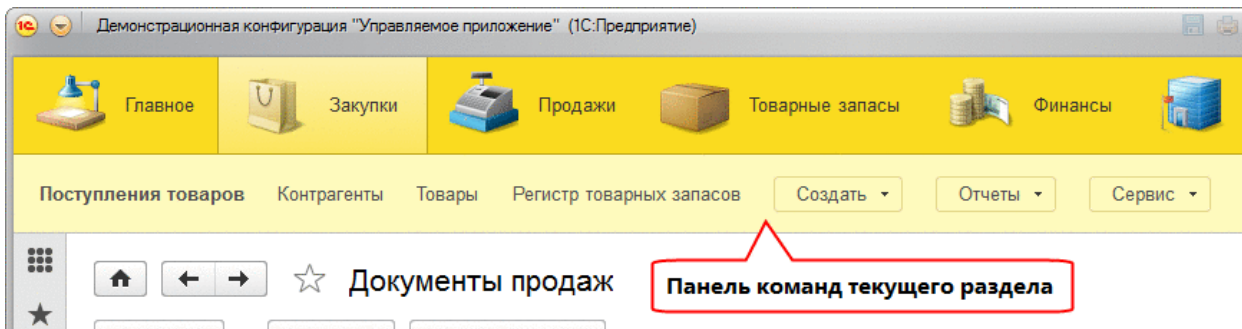


Рис. 52. Панель команд текущего раздела

Меню функций содержит команды панели навигации и панели действий. Из меню функций имеется возможность добавить какую-либо команду в избранное. Для этого достаточно нажать звездочку, расположенную слева от имени команды. Для того чтобы удалить команду из избранного, необходимо еще раз нажать на картинку звездочки.

В правом верхнем углу меню функций присутствует поле ввода для поиска и меню настроек. В меню находятся команды настройки панели навигации (команда [Настройка навигации](#)) и панели действий (команда [Настройка действий](#)).

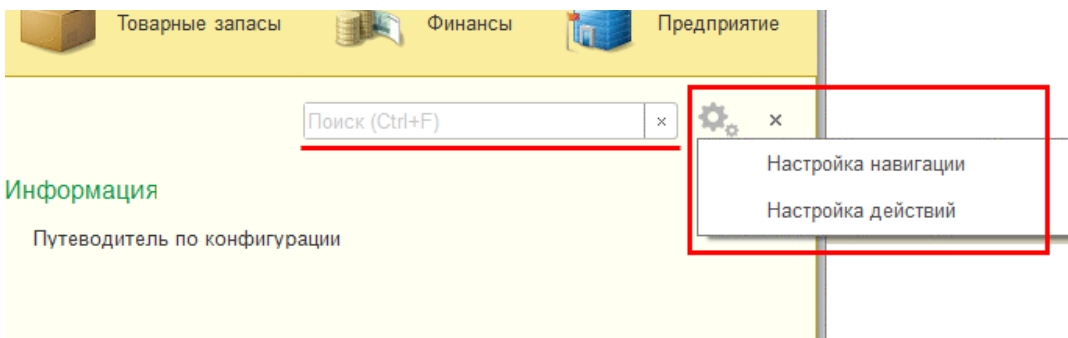


Рис. 53. Сервис в меню функций

Для начала поиска следует нажать **Ctrl+F** или начать набор текста. После ввода текста в левой части панели с меню функций появляется текст [Поиск по функциям](#) и отображается анимация (на время поиска). Поиск выполняется по всем командам всех разделов, а не только по командам текущего раздела. Результат поиска отображается следующим образом:

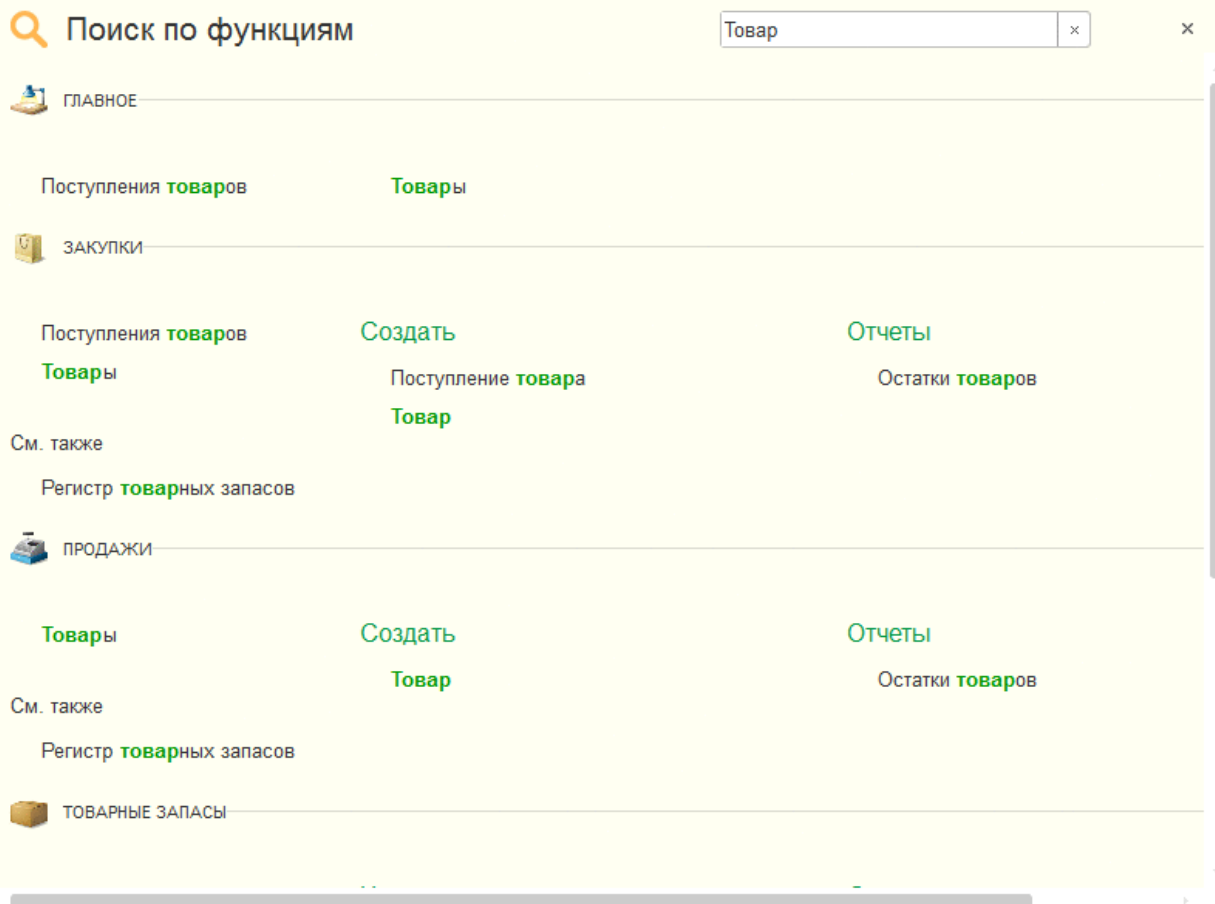


Рис. 54. Результат поиска по функциям

В результат поиска попадают команды, представление которых содержат любое из слов в строке поиска в любой позиции. Найденные вхождения подсвечиваются. Команды категории **Важное** не выделяются полужирным шрифтом. Найденные команды группируются по разделам (вместе с отображением картинки раздела) и находятся в тех же колонках меню функций, что и при отключенном поиске. Поиск выполняется не только по представлениям команд, но и по представлениям групп команд (кроме группы **См. также**). Если одно или несколько слов найдены в названии группы (раздела), то поиск в представлениях команд выполняется без учета этого слова (слов).

Для выхода из режима поиска можно очистить поле поиска, нажать кнопку очистки поля поиска или нажав клавишу **ESC**.

Открыть меню функций можно несколькими способами:

- Выбором раздела;
- С помощью команды панели инструментов (см. [здесь](#)).
- С помощью клавиатуры:
 - С помощью клавиши **F10**;
 - С помощью сочетания клавиш **Ctrl+Shift+T**;
 - С помощью сочетания клавиш **Alt+2**. В этом случае курсор будет установлен на первую команду панели навигации;
 - С помощью сочетания клавиш **Alt+3**. В этом случае курсор будет установлен на первую команду панели действий.

ПРИМЕЧАНИЕ. Если в панели разделов нет ни одного раздела (они недоступны или скрыты пользователем), то панель разделов автоматически скрывается.

3.3.1.3. Панель команд текущего раздела

Панель команд текущего раздела содержит команды, соответствующие текущему разделу.

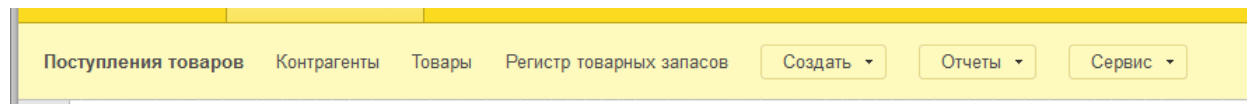


Рис. 55. Панель команд текущего раздела

Панель команд текущего раздела:

- Команды панели навигации располагаются в начале панели. Команды группы **Важное** выделены полужирным шрифтом.

В случае, когда панель отображается горизонтально, отображается максимально возможное количество команд (минимум одна). Если все команды не помещаются в панель – на панели отображается подменю **Еще**, которое содержит оставшиеся команды.

Если панель расположена вертикально, то отображается не менее одной и не более 10 команд панели навигации. Если в панели присутствует подменю **Еще** – оно будет располагаться 11 элементом командой панели.

- После команд панели навигации располагаются подменю для всех непустых групп команд панелей навигации и панели действий выбранного раздела.

3.3.1.4. Панель открытых

Данная панель содержит список форм, открытых в текущем сеансе работы с прикладным решением.

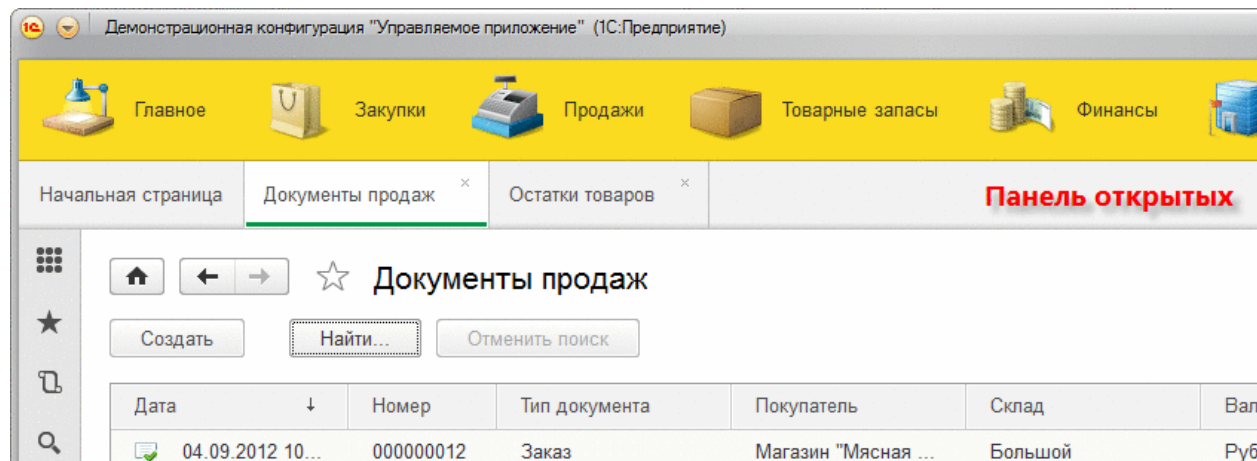


Рис. 56. Панель открытых

Первой открытой формой всегда является начальная страница (если она настроена в прикладном решении). Формы рабочей области начальной страницы не отображаются как отдельные формы в панели открытых. Нажатие картинки («крестик») в правом верхнем углу элемента панели приведет к закрытию формы, которая отображается соответствующим элементом.

3.3.1.5. История

«1С:Предприятие» сохраняет историю работы пользователя, которую можно применять для быстрого доступа к недавно открытым, созданным или отредактированным объектам информационной базы (документам, элементам справочников и др.). История содержит навигационные ссылки на элементы прикладного решения, упорядоченные по времени использования. История может отображать в виде формы истории и в виде панели истории.

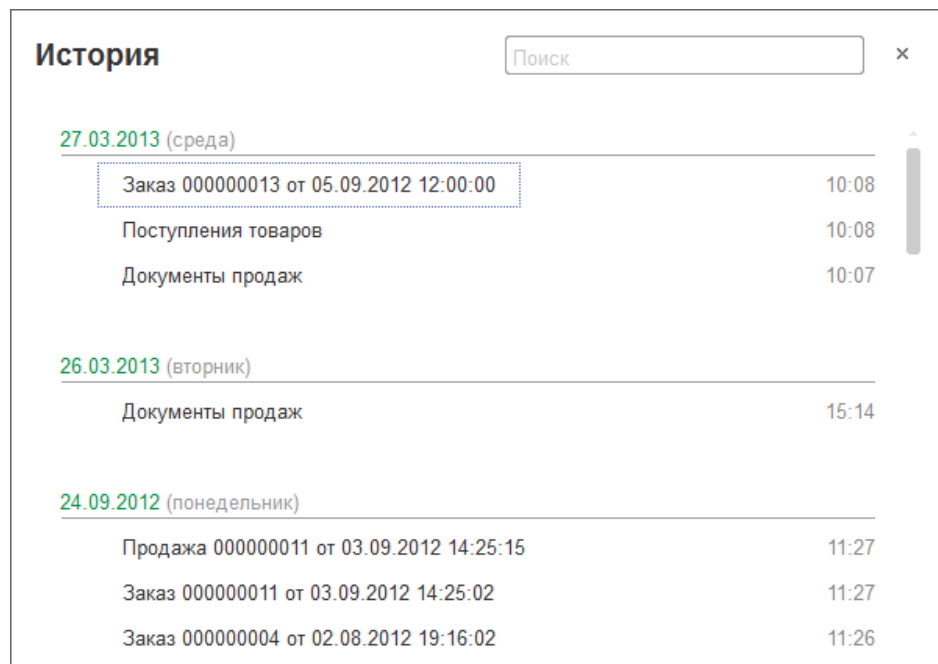


Рис. 57. Форма истории

Элементы списка группируются по датам, когда использовалась та или иная форма прикладного решения. В рамках даты элементы упорядочены по времени использования, которое отображается справа от представления элемента.

История работы пользователя хранится в информационной базе. При этом в истории хранится только одна запись на один объект информационной базы (запись о последующем изменении замещает запись о предыдущем изменении данного объекта в течении дня). Хранится не более 400 записей для конкретного пользователя. Если число сохраненных элементов равно 400, то при добавлении новых элементов истории наиболее старые события истории удаляются.

Если список истории достаточно большой, то облегчить поиск в нем можно с помощью поиска. Для ввода поискового запроса предназначено специальное поле в верхней части формы, доступ к которой возможен с помощью сочетания клавиш **Ctrl+F** или просто начав набирать требуемый текст.

Если история отображается в панели основного окна (см. [здесь](#)), то такой список обладает следующими особенностями:

- Отсутствует группировка по датам и отображение времени использования объекта прикладного решения.
- При горизонтальном расположении панели в ней отображается столько элементов, сколько помещается в отведенном месте.
- При вертикальном расположении панели в ней отображается не более 10 первых элементов.
- При нажатии на заголовок панели (слово **История**) открывается форма истории.

Открыть форму истории можно с помощью команды панели инструментов или сочетания клавиш **Ctrl+Shift+H**.

Информация попадает в историю работы пользователя несколькими путями:

- При выполнении интерактивных операций (открытие, создание, запись);
- При отображении оповещения пользователя, в котором присутствует навигационная ссылка на какой-либо объект системы (в историю попадет объект, на который указывает навигационная ссылка). Подробнее о механизме оповещений см. [здесь](#).
- При программном изменении истории работы пользователя через свойство глобального контекста **ИсторияРаботыПользователя**.

Если выполняется программная запись или открытие объекта, то информация о таком объекте не попадет в историю.

3.3.1.6. Избранное

Избранное содержит список навигационных ссылок на различные элементы прикладного решения, которые пользователь отметил для частого использования. Избранное может отображать в виде формы избранного и в виде панели избранного.

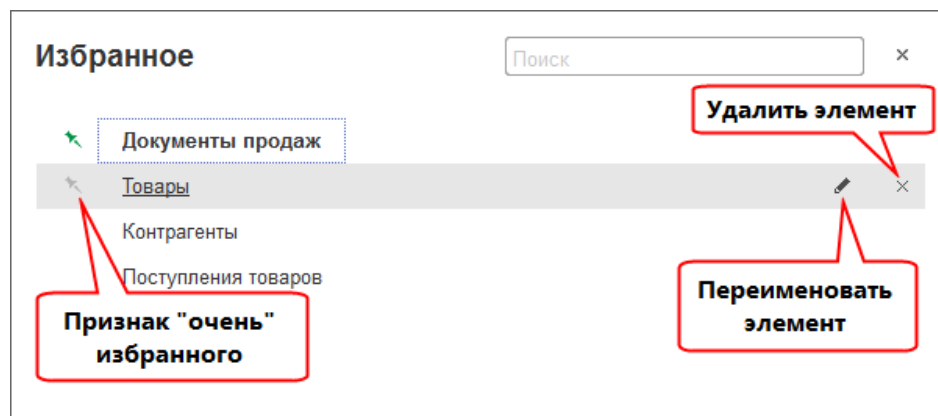


Рис. 58. Форма избранного

Среди добавленных элементов пользователь может выделить элементы, которые он использует чаще всего. Для этого необходимо нажать пиктограмму кнопки, расположенной слева от элемента. Такой элемент будет выделен полужирным шрифтом и при последующем открытии избранного будет расположен в верхней части панели и не будет изменять своего положения. Также имеется возможность переименовать элемент или удалить его из списка.

Если список избранного достаточно большой, то облегчить нахождение в нем нужного элемента можно с помощью функции поиска. Для ввода поискового запроса предназначено специальное поле в верхней части формы, доступ к которой возможен с помощью сочетания клавиш **Ctrl+F** или просто начав набирать требуемый текст.

Если избранное отображается в панели основного окна (см. [здесь](#)), то такой список обладает следующими особенностями:

- Отсутствует возможности поиска, переименования, удаления и отметки важного элемента избранного.
- При горизонтальном расположении панели в ней отображается столько элементов, сколько помещается в отведенном месте.
- При вертикальном расположении панели в ней отображается не более 10 первых элементов.
- При нажатии на заголовок панели (слово **Избранное**) открывается форма избранного.

Открыть форму избранного можно с помощью команды панели инструментов или сочетания клавиш **Ctrl+Shift+B**.

Избранное недоступно, если у пользователя отсутствует право **СохранениеДанныхПользователя** (см. [здесь](#)).

3.3.1.7. Панель инструментов

Панель инструментов предназначена для быстрого доступа к следующим функциям системы:

- Меню функций. Подробнее см. [здесь](#);

1226

- Избранное работы пользователя (если для текущего пользователя установлено право [СохранениеДанныхПользователя](#) (см. [здесь](#)). Подробнее см. [здесь](#);

- История работы пользователя. Подробнее см. [здесь](#);

- Форма полнотекстового поиска (если он включен). Подробнее см. [здесь](#).

Вышеприведенный список соответствует картинкам на панели инструментов.

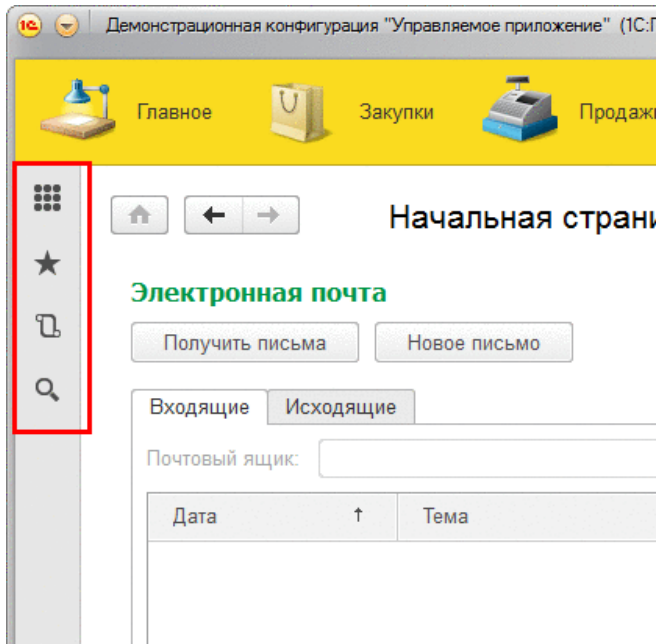


Рис. 59. Панель инструментов

3.3.1.8. Форма полнотекстового поиска

Чтобы воспользоваться полнотекстовым поиском, система предоставляет специальную форму, которую можно вызвать из панели инструментов (см. [здесь](#)). Эта форма доступна только в том случае, если включен полнотекстовый поиск.

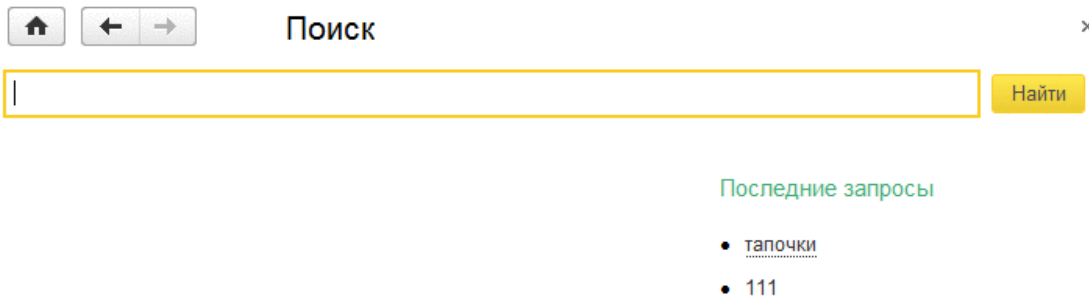


Рис. 60. Форма полнотекстового поиска

В правой части формы отображаются последние запросы, введенные данным пользователем. Для быстрого получения результата поиска, достаточно нажать гиперссылку с нужным текстом.

При использовании полнотекстового поиска есть возможность применять специальный язык, описание которого см. [здесь](#).

Если прикладного разработчика не устраивает системная форма полнотекстового поиска, то он может ее заменить с помощью свойства конфигурации [Основная форма поиска](#) (см. [здесь](#)).

Открыть форму полнотекстового поиска можно с помощью команды панели инструментов или сочетания клавиш **Ctrl+Shift+F**.

3.3.1.9. Область системных команд

Область системных команд позволяет выполнять ряд действий, не зависящих от прикладной специфики конфигурации, но помогающих в управлении системой. Кроме того, в этой области расположены различные команды, помогающие в работе с системой.

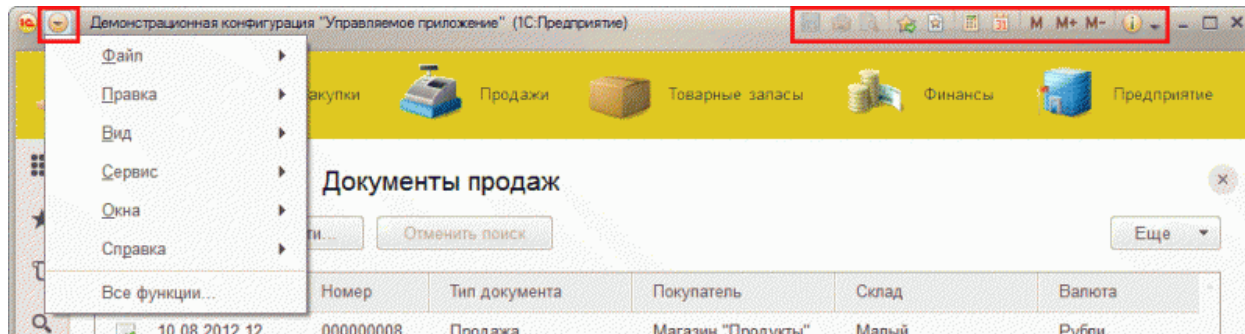


Рис. 61. Область системных команд

В левой части области системных команд расположена кнопка вызова главного меню прикладного решения. В частности, главное меню содержит меню **Вид**, который позволяет управлять внешним видом основного окна приложения.

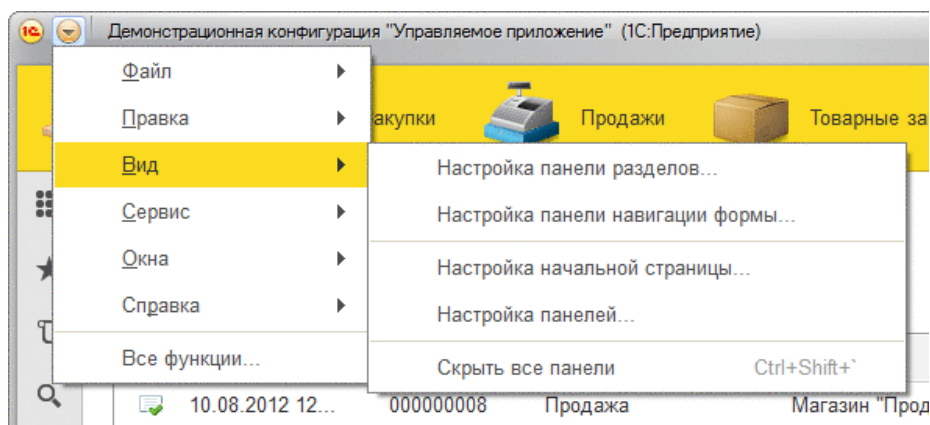


Рис. 62. Меню «Вид»

Данное меню позволяет настроить отображение панели разделов, выполнить настройку начальной страницы и расположение панелей. Также в этом меню расположена команда, позволяющая скрыть все панели основного окна приложения, кроме панели инструментов. Повторное выполнение команды возвращает скрытые панели. Панель инструментов будет располагаться в такой ориентации, как это задано пользователем в редакторе панелей.

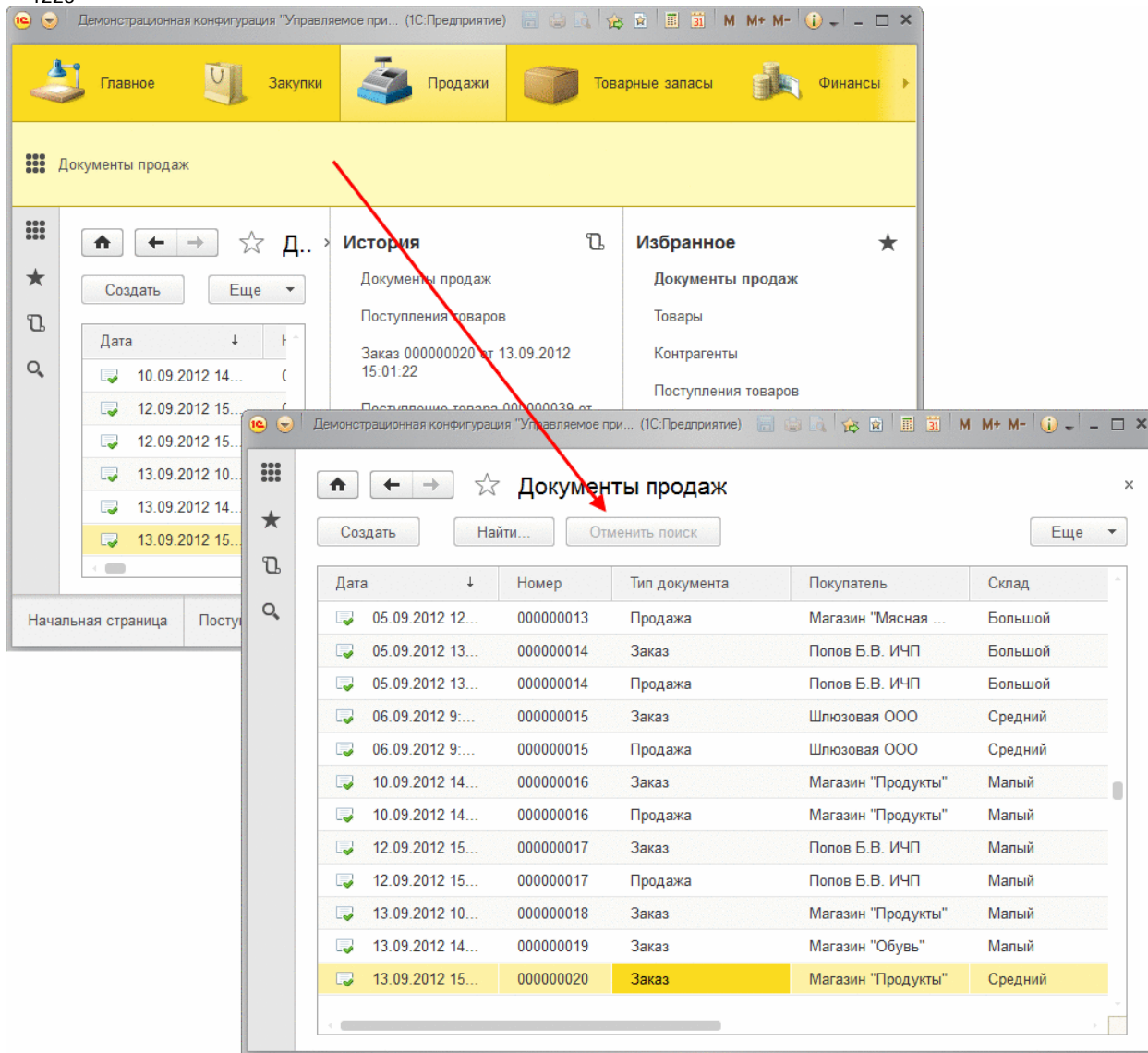


Рис. 63. Скрыть все панели

В правой части области системных команд располагаются различные сервисные команды, помогающие работе с системой и не зависящие от используемого прикладного решения, например, команды печати, получения навигационной ссылки и перехода по ней и т. д.

3.3.1.10. Настройка панелей

3.3.1.10.1. Интерактивная

Интерфейс **Такси** позволяет выполнять настройку отображения и местоположения различных панелей в основном окне приложения. Для этого служит **Редактор панелей**. Для его вызова необходимо использовать команду главного меню приложения **Главное меню – Вид – Редактор панелей...**

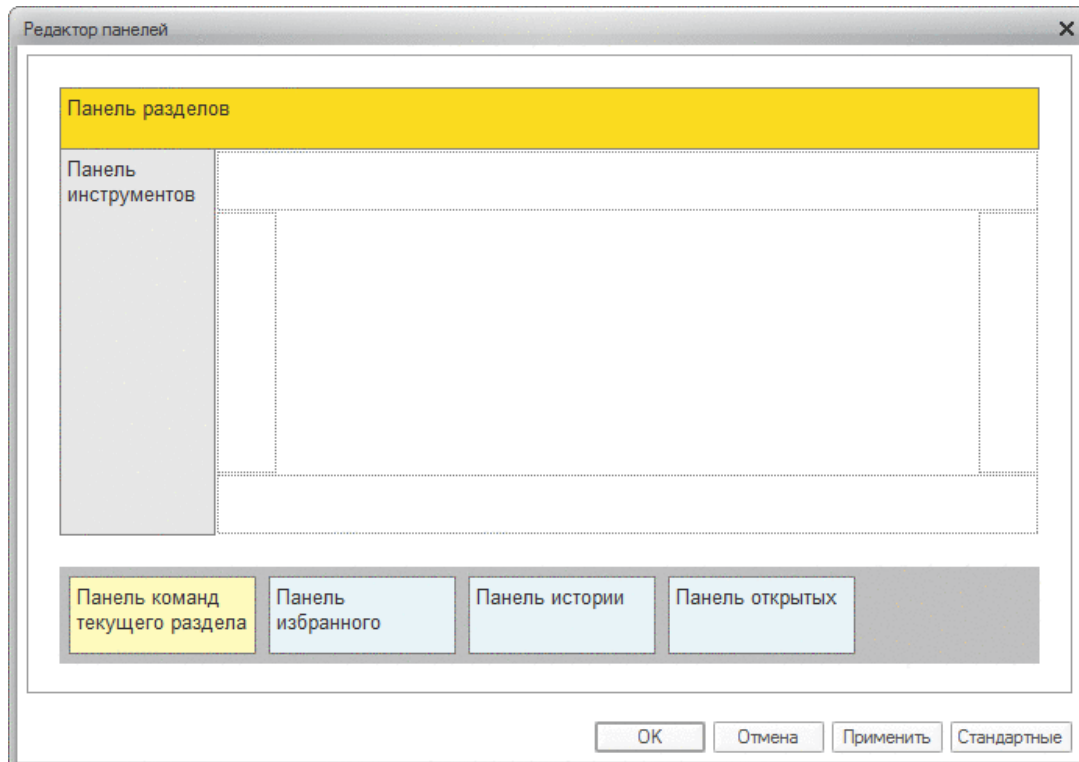


Рис. 64. Редактор панелей

Открыв редактор, пользователь имеет возможность перемещать мышью соответствующий элемент основного окна приложения (нижняя часть формы, выделенная серым фоном) в нужное место. Элементы следует перетаскивать внутрь светлых прямоугольников, выделенных серой линией. В этом случае панели будут находиться в разных строках или колонках. Можно перетащить элемент на уже существующий элемент. В этом случае элементы будут расположены в одной строке или колонке. В центре редактора отображается рабочая область начальной страницы. В нее невозможно поместить ни одну панель.

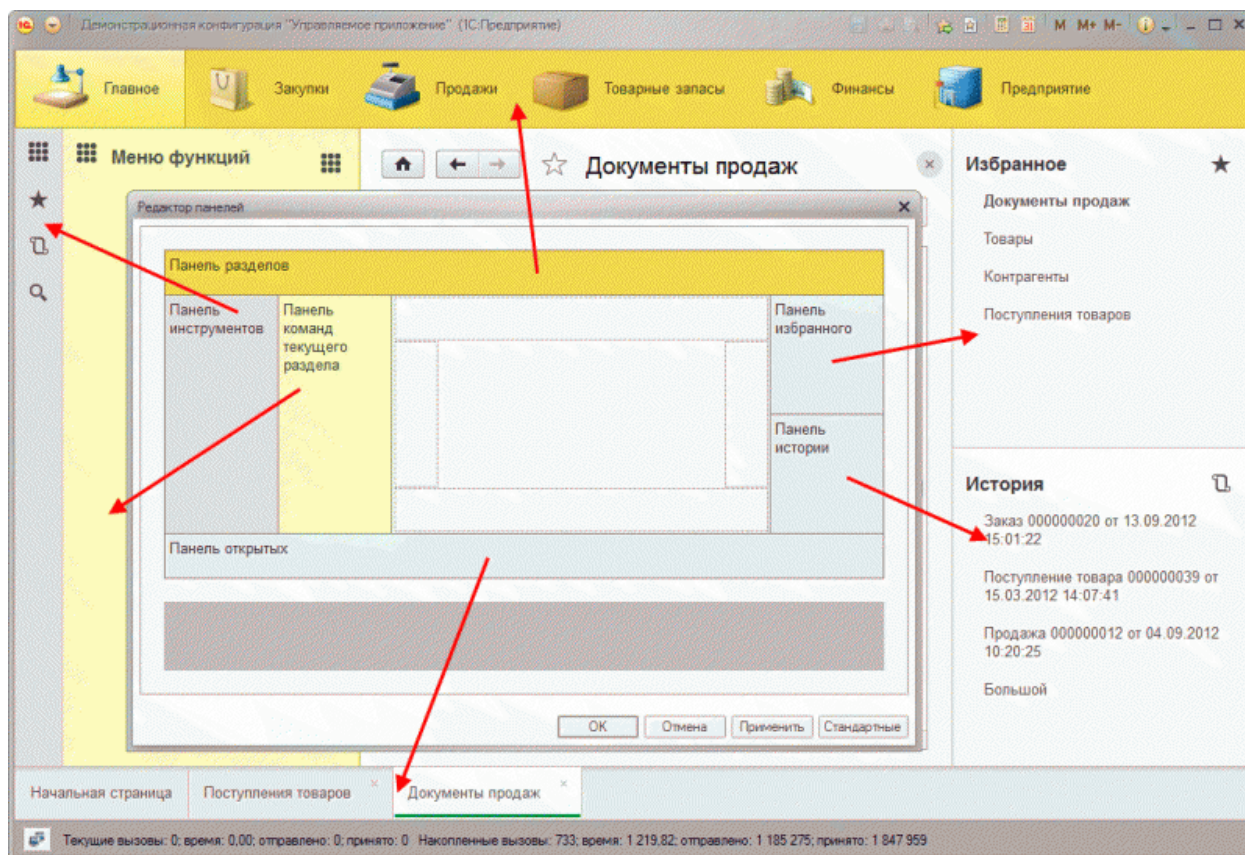


Рис. 65. Пример расположения панелей

Для удаления элемента, нужно перенести его со схемы обратно на элемент с серым фоном.

Кроме рабочей области основного окна приложения, остальные элементы могут быть убраны из основного окна приложения. Например, можно создать минималистический интерфейс прикладного решения, когда в основном окне присутствуют только рабочая область, панели инструментов и панель открытых.

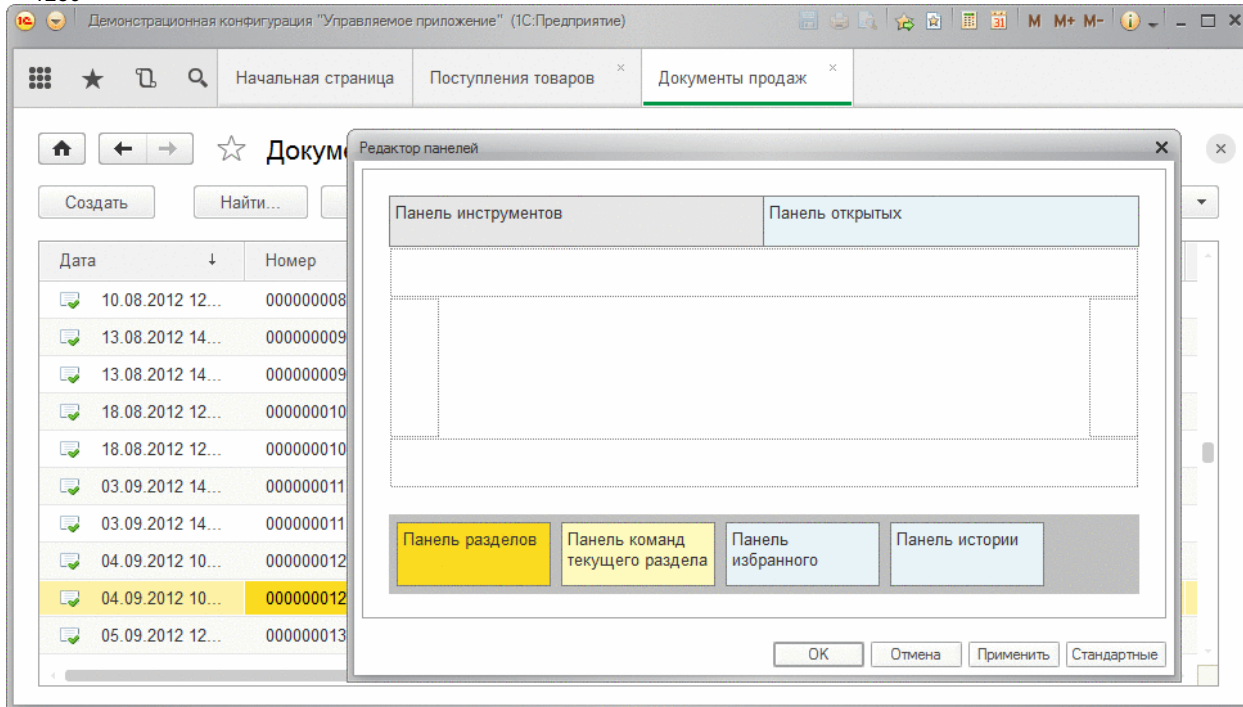


Рис. 66. Минималистичный интерфейс

При нажатии кнопки **Стандартные**, интерфейс будет приведен в состояние, которые прикладной разработчик задал с помощью свойства конфигурации **Интерфейс клиентского приложения** (см. [здесь](#)).

3.3.1.10.2. Программная

Настройку отображения и местоположения различных панелей в основном окне приложения можно выполнить также из встроенного языка. Для выполнения настройки следует воспользоваться методами `ПолучитьСостав()` и `УстановитьСостав()` объекта `НастройкиИнтерфейсаКлиентскогоПриложения`. Таким образом, имеется возможность получить текущие настройки интерфейса клиентского приложения, модифицировать их нужным образом и установить обратно.

Настройки клиентского приложения хранятся в стандартном хранилище системных настроек. Ключ настроек – `Общее/НастройкиИнтерфейсаКлиентскогоПриложения` (см. [здесь](#)). Чтобы изменить настройки необходимо в начале прочитать их из хранилища:

```
Настройки = ХранилищеСистемныхНастроек.Загрузить("Общее/НастройкиИнтерфейсаКлиентскогоПриложения");
```

Затем настройки можно модифицировать и поместить обратно в хранилище:

```
ХранилищеСистемныхНастроек.Сохранить("Общее/НастройкиИнтерфейсаКлиентскогоПриложения", "", Настройки);
```

После сохранения нового варианта настроек, необходимо выполнить перестроение интерфейса клиентского приложения с помощью вызова метода `ОбновитьИнтерфейс()`.

Рассмотрим пример, в рамках которого необходимо при каждом начале работы системы принудительно устанавливать упрощенный вариант интерфейса клиентского приложения, который состоит только из одной панели (панели инструментов), расположенной вдоль левого границы основного окна приложения. Для реализации примера необходим общий модуль с установленными признаками `Сервер` и `Вызов сервера`. Пусть общий модуль имеет имя `РаботаСИнтерфейсом`. В общем модуле необходимо поместить следующую процедуру:

```
Процедура МинимальныйИнтерфейс() Экспорт
    Настройки = Новый НастройкиИнтерфейсаКлиентскогоПриложения;
    НастройкиСостава = Новый НастройкиСоставаИнтерфейсаКлиентскогоПриложения;
    НастройкиСостава.Лево.Добавить(Новый ЭлементНастройкиСоставаИнтерфейсаКлиентскогоПриложения("ПанельИнструментов"));
    Настройки.УстановитьСостав(НастройкиСостава);
    ХранилищеСистемныхНастроек.Сохранить("Общее/НастройкиИнтерфейсаКлиентскогоПриложения", "", Настройки);
КонецПроцедуры
```

В модуле управляемого приложения необходимо разместить обработчик события `ПриНачалеРаботыСистемы`:

```
Процедура ПриНачалеРаботыСистемы()
    РаботаСИнтерфейсом.МинимальныйИнтерфейс();
    ОбновитьИнтерфейс();
КонецПроцедуры
```

В результате, вне зависимости от того, какие настройки осуществил пользователь в предыдущем сеансе работы, каждый новый сеанс будет начинаться с минимального интерфейса.

3.3.2. Вспомогательное окно

Вспомогательное окно – это окно приложения, в котором открываются формы, созданные в прикладном решении. Вспомогательное окно может быть открыто в рабочей области основного окна приложения и в отдельном окне. Режим открытия вспомогательного окна приложения зависит от значения свойства формы [Режим открытия окна](#) (см. [здесь](#)).

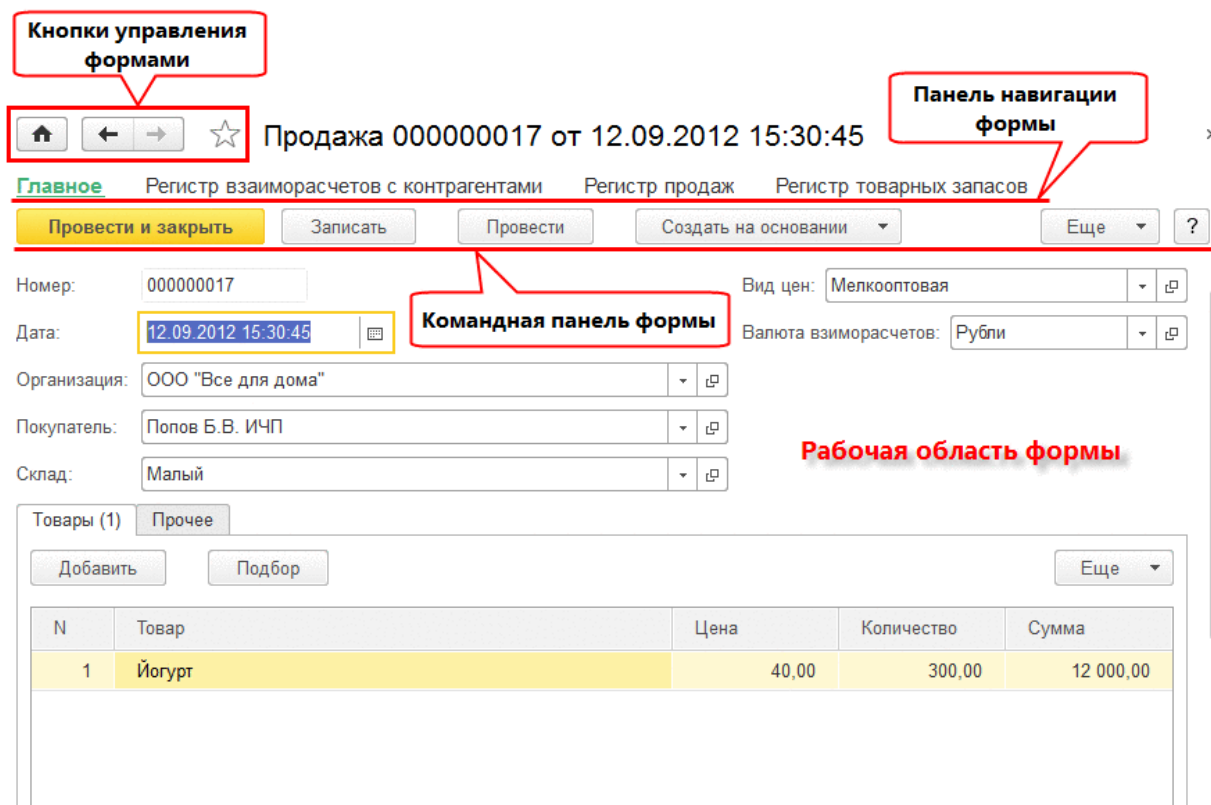


Рис. 67. Структура вспомогательного окна приложения

Закрытие вспомогательного окна не приводит к закрытию всего приложения. Вспомогательных окон может быть открыто сколько угодно, но для каждого объекта информационной базы (например, документа или элемента справочника) оно одно, т. е. нельзя открыть несколько окон для просмотра одного и того же документа, если это специально не предусмотрено в прикладном решении.

Во вспомогательном окне можно выделить **основную форму** – форму, которая отображается по умолчанию при открытии вспомогательного окна и которая открывается при нажатии на **Главное** панели навигации формы. При нажатии на другие ссылки панели навигации формы будут открываться **вспомогательные формы**.

Кнопки управления формами располагаются во все вспомогательных формах. Эти кнопки имеют следующее назначение (слева направо):

- Кнопка перехода к начальной странице;
- Кнопки перехода к предыдущей/следующей форме предназначены для перехода по списку открытых окон. Кнопка перехода к следующей форме активизируется только после того, как хотя бы раз была нажата кнопка перехода к предыдущему окну.
- Кнопка помещения формы в список избранного (см. [здесь](#)). Если форма отсутствует в списке избранного – цвет звездочки соответствует фону формы. В противном случае звездочка будет окрашена в желтый цвет. Данная кнопка недоступна, если у пользователя отсутствует право [СохранениеДанныхПользователя](#) (см. [здесь](#)).

3.3.2.1. Панель навигации формы

Панель навигации вспомогательного окна (панель навигации формы) позволяет просматривать различные сведения, логически связанные с данными, которые отображает основная форма окна.

В панели навигации могут располагаться команды трех групп: **Важное**, **Перейти** и **См. также**. Общее расположение команд следующее:

- Первой командой всегда является команда **Главное** (выделена **полужирным** шрифтом), которая предназначена для открытия основной формы вспомогательного окна приложения.
- Затем следуют команды группы **Важное**.
- Затем следуют команд группы **Перейти**.
- Последними следуют команды группы **См. также**. Если разработчик конфигурации определил собственные группы команд из категории **Панель навигации формы**, то команды из этих групп будут располагаться непосредственно перед командами группы **См. также** (команды этой группы всегда располагаются в самом конце панели навигации формы).

Если команды не помещаются на панели навигации, то последней командой панели будет выступать подменю **Еще**, в котором будут располагаться все команды, которые не поместились на отображаемой части панели навигации.

3.3.2.2. Командная панель формы

Командная панель формы содержит команды, непосредственно связанные с объектом, который отображается в основной форме. Существуют стандартные группы [Важное](#) и [Создать на основании](#); кроме того, на панели могут размещаться группы, созданные разработчиком. Каждая команда из группы [Важное](#) будет отображаться своей кнопкой на панели. Группа [Создать на основании](#) будет оформлена в виде подменю панели. Также в виде подменю будут оформлены все группы команд, созданные разработчиком. Команды формы, такие как [Сохранить и закрыть](#), будут расположены перед командами группы [Важное](#).

Если в какой-либо группе нет ни одной команды, то группа не показывается. Если разработчик конфигурации добавил в эту область собственные группы, то они располагаются после группы [Создать на основании](#).

В том случае, если все команды командной панели не помещаются на форме, они помещаются в специальное подменю [Еще](#), которое располагается последней командой панели. Исключением является кнопка вызова справки, которая располагается правее подменю [Еще](#) в том случае, если для формы установлена справочная информация.

Глава 30. Групповая разработка конфигурации

30.1. Общая информация

Разработка конфигурации авторским коллективом может производиться последовательно, когда разработчики договариваются о модификации объектов конфигурации, производят изменения, а затем объединяют результат. Такой режим связан с риском случайного изменения других объектов, с необходимостью понимания всех процессов изменений и взаимосвязи объектов. Объединение конфигураций при этом должен производить специалист, хорошо представляющий направление разработки.

Снизить вероятность ошибок, повысить производительность и упростить процесс разработки призван механизм групповой разработки конфигурации, рассматриваемый в данной главе.

Под групповой разработкой понимается одновременная работа группы пользователей (разработчиков) по изменению конфигурации, при которой модификация объекта конфигурации доступна только тому разработчику, который предварительно захватил данный объект.

Для групповой разработки создается хранилище, в которое средствами конфигуратора помещается конфигурация. Организуется доступ разработчиков к хранилищу конфигурации: либо в рамках локальной сети, либо удаленный доступ с использованием веб-сервера. Для хранилища назначается администратор, который осуществляет формирование списка пользователей, имеющих доступ к хранилищу. Административные права могут быть назначены и другим пользователям.

При групповой разработке конфигурация рассматривается как набор объектов, закрытых для изменения. Чтобы произвести изменения в объекте, его нужно захватить. Одновременно объект может быть захвачен только одним пользователем. Пользователь может захватить произвольное число не захваченных другими пользователями объектов.

Приемы работы с захваченными объектами конфигурации ничем не отличаются от приемов работы в обычном режиме. Допускается редактирование свойств самого объекта, удаление и создание подчиненных объектов.

Для добавления объектов требуется, чтобы был захвачен объект, которому он подчинен. Например, для добавления константы необходимо захватить корневой объект конфигурации. Для добавления реквизита или формы объекта – сам объект.

Для удаления незахваченного объекта требуется захватить сам объект, а также объект, которому он подчинен, и все подчиненные ему объекты.

После работы с захваченными объектами результат их модификации можно поместить в хранилище. С другой стороны, если какие-либо незахваченные объекты были изменены, то можно получить обновление данных объектов сразу после помещения этих объектов в хранилище конфигурации автором изменений.

Результаты работы с хранилищем можно просмотреть в истории хранилища. Версии конфигурации, сохраненные в хранилище, можно открыть для просмотра, сравнить с текущей конфигурацией, конфигурацией базы данных, а также сравнить различные версии хранилища.

Объекты конфигурации тесно взаимосвязаны друг с другом. Поэтому хранилище конфигурации поддерживает целостность метаданных при захвате или помещении объектов в хранилище.

Таким образом, механизм групповой разработки обеспечивает синхронизацию работы группы разработчиков по модификации конфигурации.

30.2. Администрирование хранилища

30.2.1. Создание хранилища

Хранилище конфигурации создается для разделения доступа к объектам конфигурации.

Для создания хранилища нужно выбрать пункт [Конфигурация – Хранилище конфигурации – Создать хранилище](#). На экран выводится диалог:

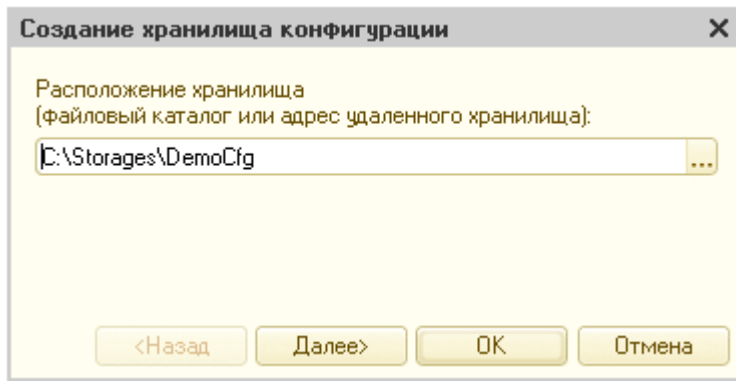


Рис. 466. Создание хранилища конфигурации

В поле [Каталог хранилища](#) нужно указать каталог расположения хранилища. По указанному каталогу не должно быть зарегистрировано другое хранилище конфигурации. Нажать кнопку [Далее >](#).

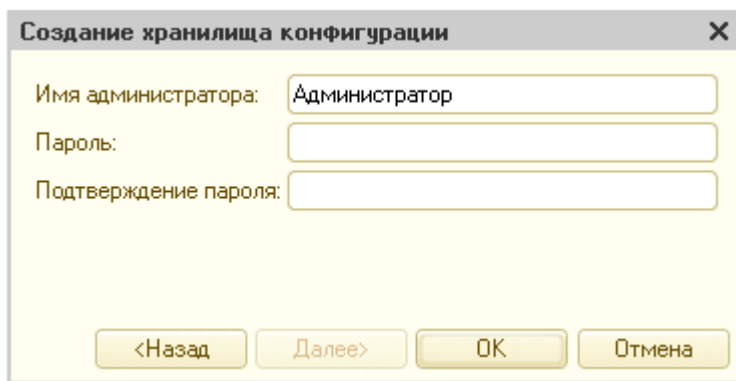


Рис. 467. Задание администратора хранилища

Указать имя администратора хранилища и пароль доступа. Администратор хранилища производит формирование списка пользователей, имеющих право доступа в хранилище конфигурации.

После нажатия клавиши **ОК** конфигуратор производит анализ данных, формирование структуры объектов и запись данных в указанный каталог. Этот процесс может занять некоторое время (в зависимости от размера конфигурации). По окончании формирования хранилища конфигуратор предлагает пользователю подключиться к хранилищу.

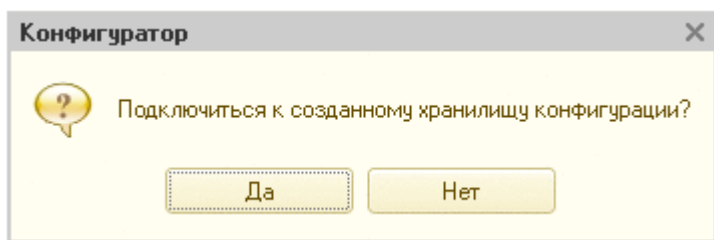


Рис. 468. Подключиться к хранилищу?

Если по указанному адресу зарегистрировано хранилище конфигурации, то конфигуратор выводит предупреждение о невозможности создания хранилища.

30.2.2. Подключение к хранилищу

Для подключения к хранилищу конфигурации нужно выбрать пункт [Конфигурация – Хранилище конфигурации – Подключиться к хранилищу](#).

Если текущая конфигурация не была подключена к хранилищу, то перед подключением конфигуратор сообщает, что текущая конфигурация будет замещена конфигурацией из хранилища, и запрашивает

подтверждение на подключение.

ВНИМАНИЕ! Конфигурация, полученная из хранилища, заменяет текущую конфигурацию и сохраняется на диске без дополнительных сообщений.

После утвердительного ответа на экран выводится диалог подключения:

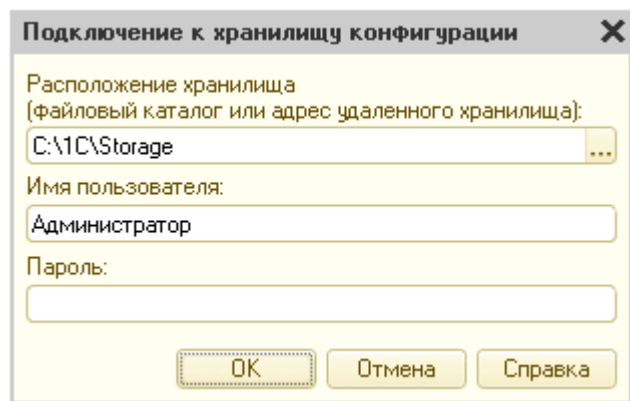


Рис. 469. Подключение к хранилищу конфигурации

Нужно ввести имя пользователя и пароль, если он задан.

Если под указанным именем никто не зарегистрирован, выводится предупреждение: [Ошибка аутентификации! Проверьте правильность введенного имени пользователя и пароля.](#)

Для каждого пользователя к данному хранилищу может быть подключена только одна конфигурация. Поэтому конфигуратор сначала производит проверку регистрации подключения по данному пользователю.

Если имя пользователя зарегистрировано и для этого пользователя информационная база еще не была подключена к хранилищу конфигураций, то производится подключение и устанавливается связь: в хранилище по данному пользователю записывается информация о конфигурации и ее месте расположения.

Если конфигурация пользователя подключена к хранилищу и связь не изменялась, то выводится окно конфигурации и пользователь может начинать работу с объектами.

Если для данного пользователя в данный момент времени есть открытая конфигурация, связанная с указанным пользователем, то конфигуратор выводит предупреждение об аутентификации.

Если для данного пользователя в данный момент времени нет открытой информационной базы, но данные о месте расположения не совпадают с данными о связи (конфигурация расположена в другом каталоге, или подключение производится с другого рабочего места), то конфигуратор выводит сообщение: [Для данного пользователя уже имеется конфигурация, связанная с данным хранилищем конфигурации. Продолжить?](#) Если нажать кнопку [Да](#), то происходит установка нового подключения, текущая конфигурация заменяется конфигурацией из хранилища и устанавливается новая связь текущей конфигурации с хранилищем для данного пользователя.

После подключения конфигуратор выводит окно конфигурации:

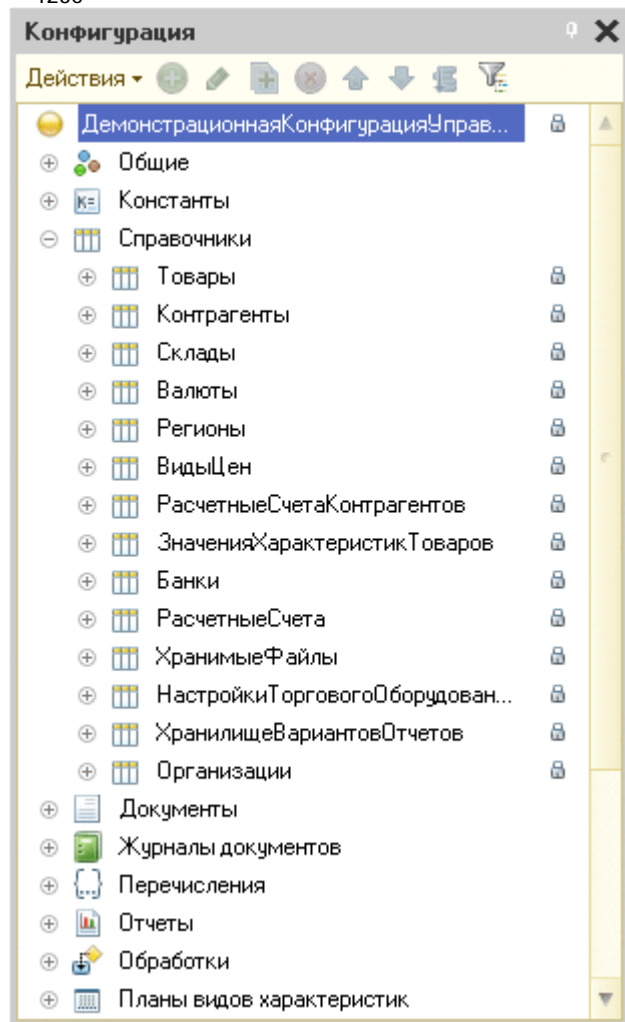


Рис. 470. Конфигурация, подключенная к хранилищу

Состояние объектов конфигурации отображается специальными значками, размещенными вдоль правой границы окна **Конфигурация**. Значки объекта и подчиненного объекта отличаются размером (для подчиненного объектов они меньше) и видом (у подчиненных объектов присутствует изображение ветви дерева).

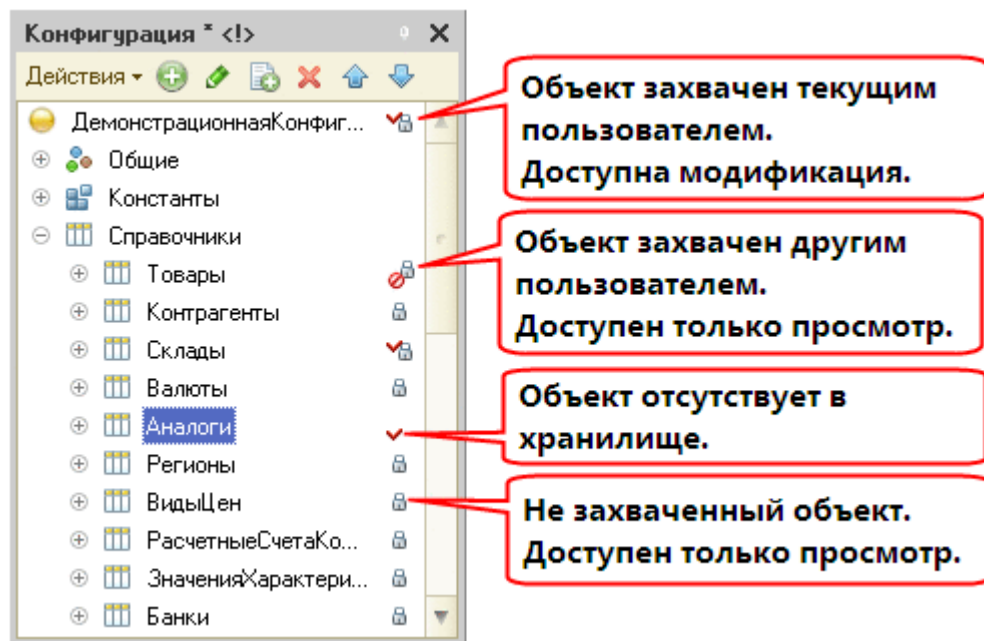


Рис. 471. Значки объектов

Возможные состояния объектов изображены на [рис. 471](#).

Если объект захвачен другим пользователем (см. [рис. 471](#)), то данный объект может быть

модифицирован только тем пользователем, который захватил данный объект. Подсказка к объекту метаданных, появляющаяся над пиктограммой текущего состояния объекта в хранилище конфигурации, содержит имя пользователя, захватившего объект, дату захвата объекта пользователем, имя компьютера, с которого было осуществлено соединение пользователя с хранилищем конфигурации, и месторасположение информационной базы.

Созданные пользователем и еще не помещенные в хранилище объекты, а также объекты, удаленные из хранилища, но еще присутствующие в локальной конфигурации, обозначаются значком в виде отметки красного цвета.

При первом подключении конфигуратор запоминает параметры подключения (каталог хранилища, имя пользователя) для данной информационной базы. При очередном открытии конфигурации конфигуратор выводит запрос на соединение с хранилищем перед открытием окна [Конфигурация](#).

Если при подключении к хранилищу конфигурации неправильно указано имя пользователя или пароль или указано имя пользователя, подключенного к данному хранилищу в данный момент времени, то выдается сообщение: [Ошибка аутентификации! Проверьте правильность введенного имени пользователя и пароля](#).

При попытке подключения ранее не подключенной конфигурации к хранилищу конфигурации с указанием имени пользователя, для которого зарегистрировано подключение (установлена связь) к данному хранилищу, но для иной информационной базы (в данный момент неоткрытой), конфигуратор выводит сообщение: [Для данного пользователя уже имеется конфигурация, связанная с данным хранилищем конфигурации. Продолжить?](#) При утвердительном ответе производится загрузка конфигурации из хранилища (текущая конфигурация полностью замещается и сохраняется на диске) и устанавливается новая связь информационной базы и хранилища. В этом случае прежняя связь будет утрачена (информационная база, которая была ранее подключена к хранилищу, уже не сможет соединиться с этим пользователем без переподключения).

Если производится попытка подключения к хранилищу конфигурации для открытой в данный момент конфигурации, то конфигуратор выдает предупреждение об ошибке с указанием имени пользователя.

Не все комбинации режима совместимости хранилища конфигурации и версий «1С:Предприятия», подключающихся к хранилищу:

	Версия 8.3.2	Версия 8.3.3	Не использовать
Версия 8.3.6	Да	Да	Да
Версия 8.3.5	Да	Да	
Версия 8.3.4	Да	Ограниченно	
Версия 8.3.3	Да	Ограниченно	
Версия 8.3.4 и младше	Да		

В колонках перечислены режимы совместимости, в строках указаны номера версий конфигуратора, с помощью которого выполняется попытка подключения к хранилищу конфигурации. Значение [Да](#) в ячейке означает, что определенная версия конфигуратора может подключиться к хранилищу с указанным режимом совместимости. Значение [Ограниченно](#) означает, что с хранилищем без ограничений могут работать только конфигураторы версий 8.3.3 и 8.3.4. Если к хранилищу, которое находится в режиме совместимости с версией 8.3.2, подключается конфигуратор версии 8.3.5 и 8.3.4, то конфигуратор версии 8.3.4 не может поместить в хранилище измененные объекты.

30.2.3. Ведение списка пользователей хранилища

Для создания нового пользователя хранилища нужно выбрать пункт меню [Конфигурация – Хранилище конфигураций – Администрирование хранилища](#) и на закладке [Пользователи](#) выбрать пункт [Действия – Добавить](#). На экран выводится окно редактирования информации о пользователе. На закладке [Основные](#) вводится наименование пользователя и пароль доступа к хранилищу.

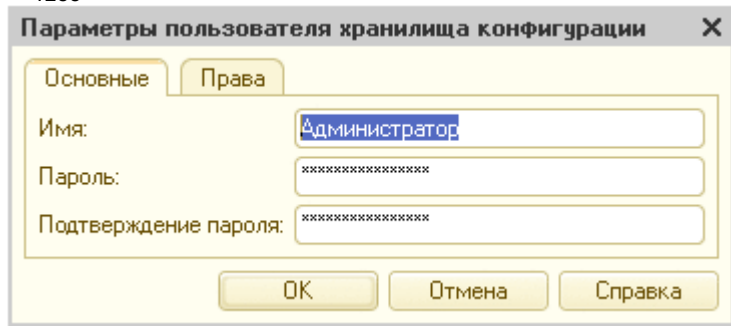


Рис. 472. Параметры пользователя хранилища

На закладке **Права** указывается, какими правами обладает данный пользователь.

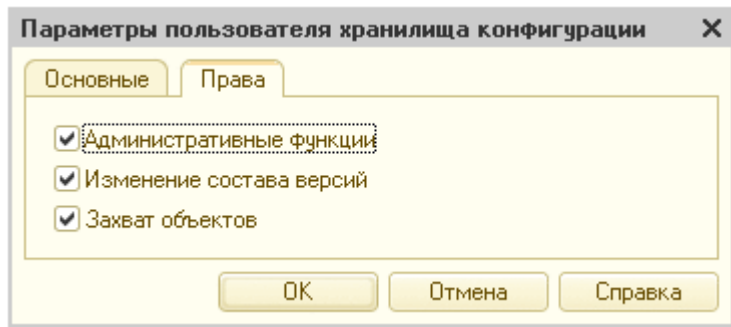


Рис. 473. Права пользователя хранилища

Административные функции – пользователь имеет права создания, удаления и отключения пользователей, отмены захвата, изменять режим совместимости хранилища и выполнять процедуру оптимизации.

Изменение состава версий – пользователь имеет права выполнения отката и сокращения истории версий (подробнее см. [здесь](#)).

Захват объектов – пользователям с данным правом доступны захват и внесение изменений в конфигурацию хранилища. Если право не установлено, то объекты хранилища можно только просмотреть.

30.2.4. Кеш версий хранилища

Для ускорения работы с хранилищем конфигурации используются два кеша версий. Один кеш – локальный, он располагается на стороне компьютера, на котором работает конфигуратор, подключенный к хранилищу. Другой кеш – глобальный, он расположен на том компьютере, где расположен каталог с хранилищем. Кеш предназначен для ускорения некоторых операций по работе с хранилищем. В кеш попадают версии в результате выполнения следующих операций:

- Подключение к хранилищу;
- Открытие версии конфигурации в хранилище;
- Сравнение версий конфигурации в хранилище;
- Сохранение версии конфигурации в хранилище;
- При выборочном сравнении следующих объектов конфигурации:
 - Формы (при работе через сервер хранилища);
 - Права доступа (при работе с хранилищем конфигурации, расположенной на сетевом ресурсе или через сервер хранилища);
 - Командные интерфейсы (при работе с хранилищем конфигурации, расположенной на сетевом

ресурсе или через сервер хранилища).

При необходимости выполнить какую либо из вышеперечисленных операций, система последовательно анализирует локальный и глобальный кеши версий. Если поиск не привел к положительному результату – выполняется получение нужной версии непосредственно из хранилища.

Управление местом расположения и объемом кешей выполняется с помощью диалога администрирования хранилища или конфигурационных файлов.

30.2.5. Администрирование хранилища конфигурации

Для администрирования хранилища конфигурации предназначено специальное окно, вызываемое пунктом меню [Конфигурация – Хранилище конфигураций – Администрирование хранилища](#).

На экран выводится окно (см. [рис. 474](#)).

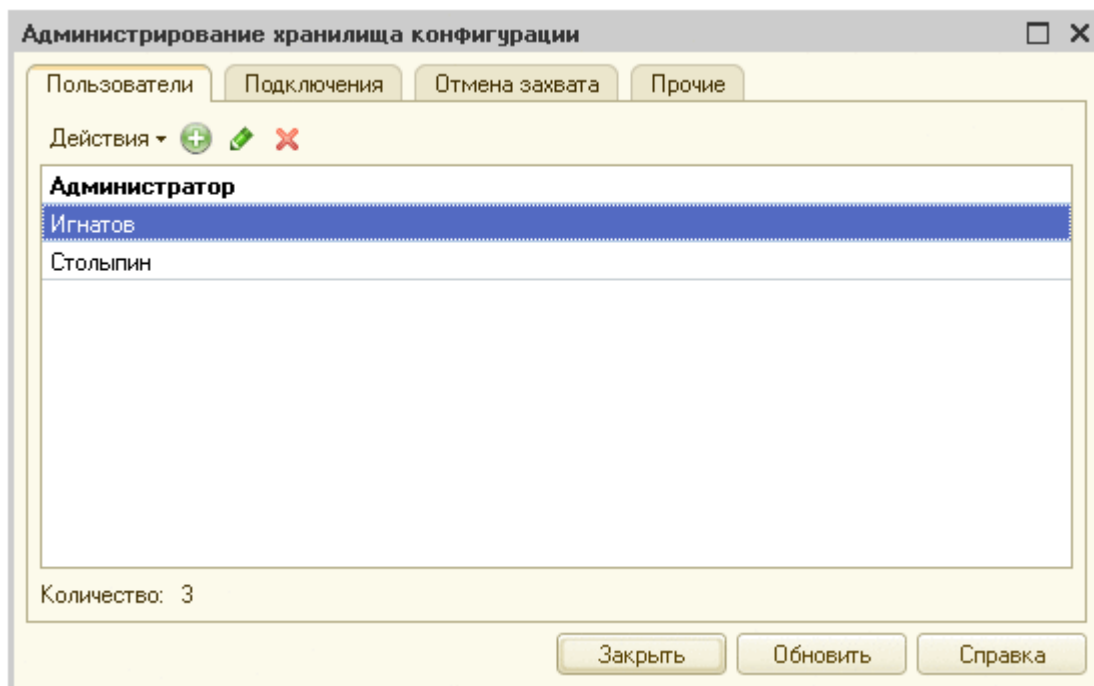


Рис. 474. Администрирование хранилища

На закладке [Пользователи](#) расположены элементы управления списком пользователей – разработчиков конфигурации (далее в этой главе – пользователи). Всем пользователям доступен режим изменения собственных параметров (имя, пароль), кроме смены прав. Жирным шрифтом выделены те пользователи, который в данный момент подключены к хранилищу. На [рис. 474](#) это пользователь [Администратор](#).

ВНИМАНИЕ! Добавление, удаление и смена прав доступны только пользователям, имеющим административные права.

На закладке [Подключения](#) выводится список пользователей, подключенных к хранилищу.

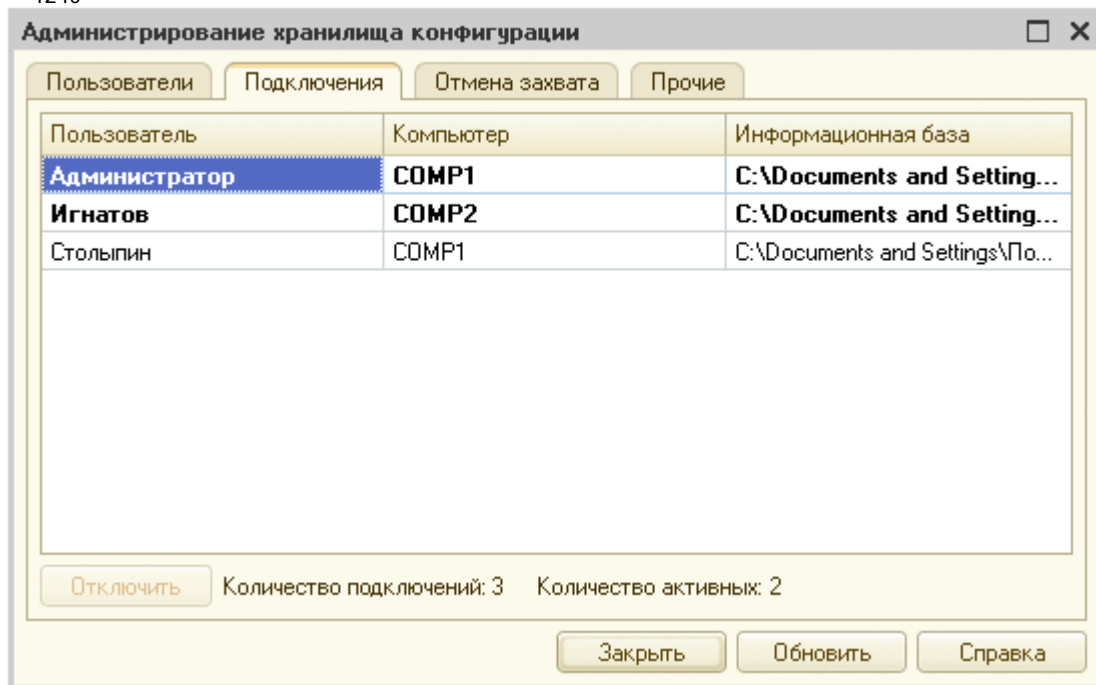


Рис. 475. Подключенные пользователи

С помощью кнопки **Отключить** можно отключить от хранилища любого из указанных пользователей, не работающих в данный момент с хранилищем.

ВНИМАНИЕ! Кнопка **Отключить** доступна пользователям, имеющим административные права.

На закладке **Отмена захвата** производится отмена захвата объектов для указанного пользователя.

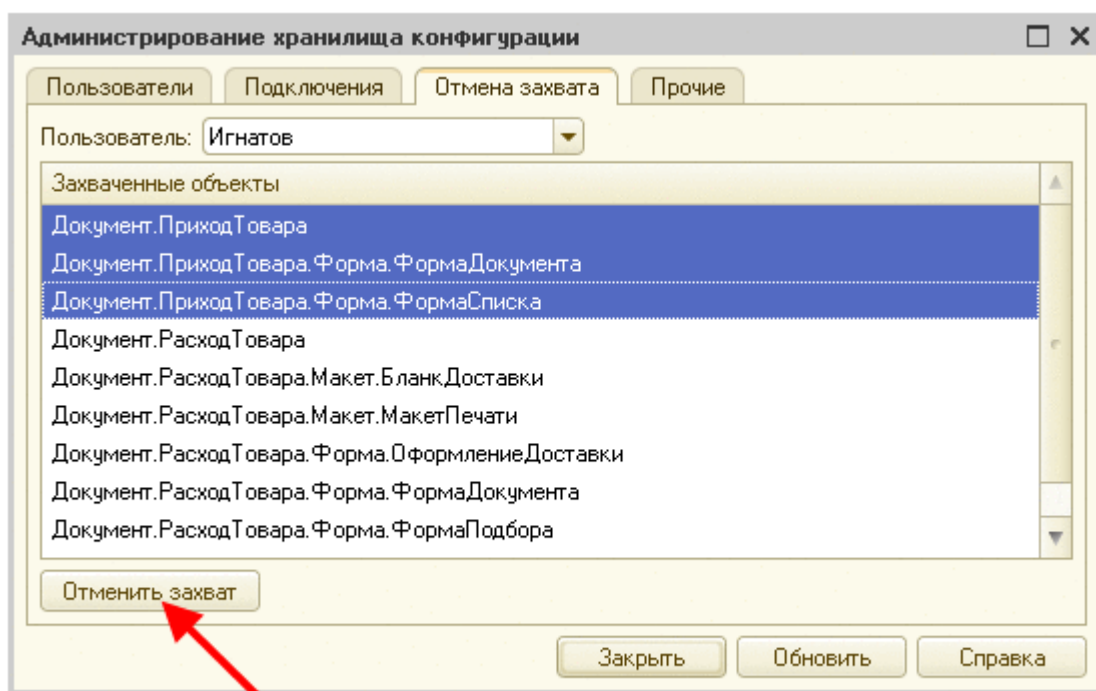


Рис. 476. Отмена захваченных объектов

При этом пользователь, захвативший объекты, лишается возможности поместить изменения этих объектов в хранилище.

ВНИМАНИЕ! Данная кнопка доступна пользователям, имеющим административные права.

На закладке **Прочие** расположены параметры, управляющие работой самого хранилища.

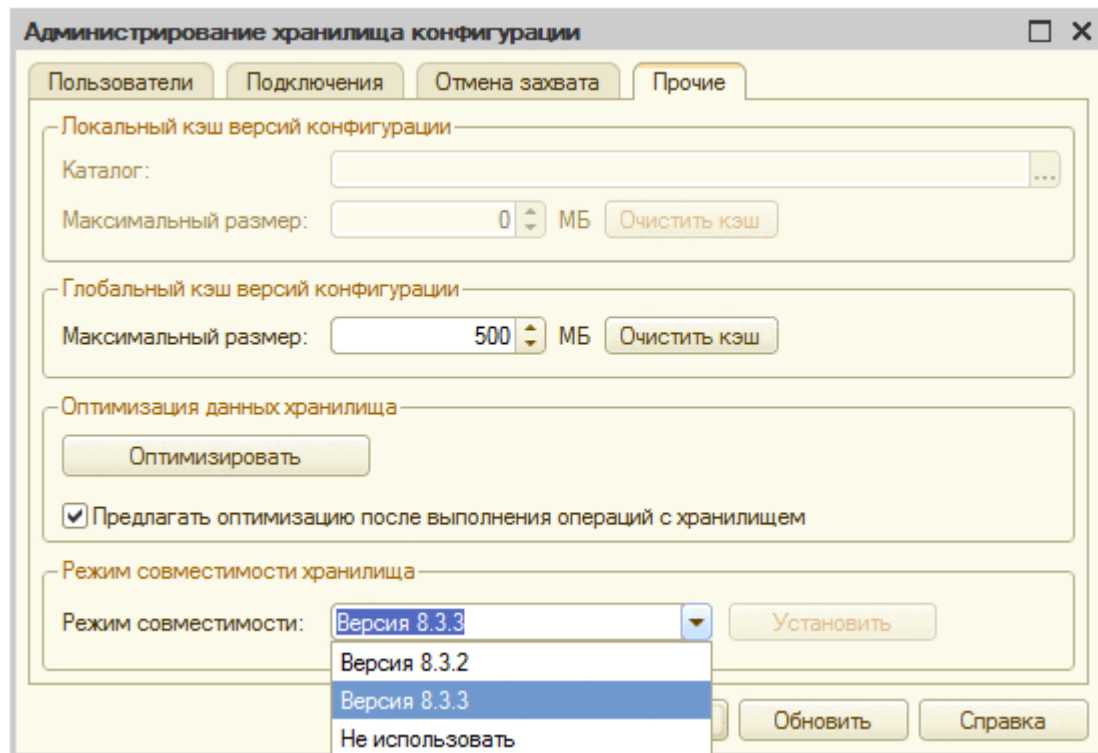


Рис. 477. Параметры хранилища

Группа параметров **Локальный кэш версий конфигурации** описывает характеристики локального кэша версий. Локальный кэш формируется в том случае, когда хранилище конфигурации расположено на сетевом диске или работа с хранилищем выполняется с помощью сервера хранилища. Свойство **Каталог** служит для указания каталога в локальной файловой системе, где будет располагаться кэш. По умолчанию, кэш располагается в каталоге:

- Для ОС Windows: `%APPDATA%\1C\1Cv8\<Уникальный идентификатор информационной базы>\cfgrepo\cache`.
- Для ОС Linux: `~/1cv8/1C/1cv8/<Уникальный идентификатор информационной базы>/cfgrepo/cache`.

Параметр **Максимальный размер** определяет размер локального кэша в мегабайтах. Значение по умолчанию – 500 Мбайт. Если в процессе работы размер локального кэша превысит установленный размер, будет автоматически удалены самые старые версии в количестве 50% от размера кэша. Кнопка **Очистить кэш** приводит к полной очистке локального кэша.

Для управления параметрами локального кэша версий служит файл `cfgrepo.conf`, описание которого приведено в книге «1С:Предприятие 8.3. Руководство администратора»

Группа параметров **Глобальный кэш версий конфигурации** служит для задания размера глобального кэша версий конфигурации. Алгоритм очистки кэша совпадает с алгоритмом очистки локального кэша версий. Кнопка **Очистить кэш** приводит к полной очистке кэша.

Для управления параметрами глобального кэша версий служит файл `cfgrepo.conf`, описание которого приведено в книге «1С:Предприятие 8.3. Руководство администратора».

Группа параметров **Режим совместимости хранилища** управляет форматом хранилища и способом взаимодействия с сервером хранилища.

В следующей таблице отражено, какие версии системы могут работать с хранилищем, для которого установлен тот или иной режим совместимости:

Режим совместимости	Реализован в версии	Может работать
Версия 8.3.2	8.3.3	Любая версия
Версия 8.3.3	8.3.6	8.3.3 и старше

Если свойство [Режим совместимости](#) установлено в значение [Версия 8.3.2](#), то с хранилищем могут работать любые версии «1С:Предприятия». Если установить это свойство в значение [Версия 8.3.3](#) и выше (включая [Не использовать](#)), то после нажатия кнопки [Установить](#), хранилище будет преобразовано в оптимизированный формат. После выполнения оптимизации работа с хранилищем будет выполняться быстрее (в том числе и при использовании сервера хранилища). Операцию изменения режима совместимости может выполнять только пользователь с административными правами.

ВНИМАНИЕ! Преобразование хранилища может занимать существенное время.

При необходимости, можно обратно преобразовать хранилище конфигурации в формат, пригодный для использования любыми версиями «1С:Предприятия». Для этого необходимо установить свойство [Режим совместимости](#) в значение [Версия 8.3.2](#) и нажать кнопку [Установить](#).

Для нового хранилища конфигурации свойство [Режим совместимости](#) устанавливается в значение [Не использовать](#).

Группа [Оптимизация данных хранилища](#) позволяет выполнить оптимизацию хранилища, которое не находится в режиме совместимости с версией 8.3.2. В результате оптимизации будет повышена скорость работы с хранилищем. Оптимизацию требуется проводить только после того, как система предложит это сделать.

Для пользователя с административными правами можно включить рекомендацию выполнить оптимизацию после проведения операции хранилищем (помещение объектов, откат версий, слияние версий). Для этого предназначен флажок [Предлагать оптимизацию после выполнения операций с хранилищем](#) на закладке [Прочие](#) диалога администрирования хранилища. В этом случае, после завершения операции с хранилищем, пользователю может быть показан диалог:

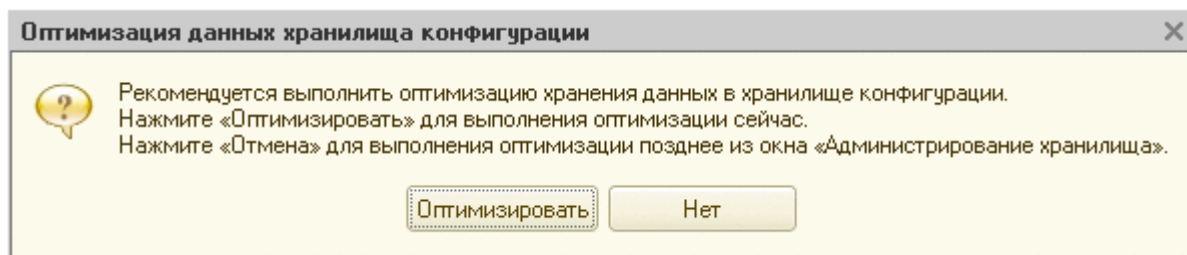


Рис. 478. Предложение оптимизировать хранилище

Если пользователь (администратор хранилища) отказывается от проведения оптимизации непосредственно при появлении рекомендации, то оптимизацию можно выполнить позднее. В этом случае следует зайти в диалог администрирования хранилища и на закладке [Прочие](#) нажать кнопку [Оптимизировать](#). Кнопку стоит нажимать в том случае, если справа от нее отображается рекомендация выполнить оптимизацию. Оптимизация хранилища может занимать существенное время.

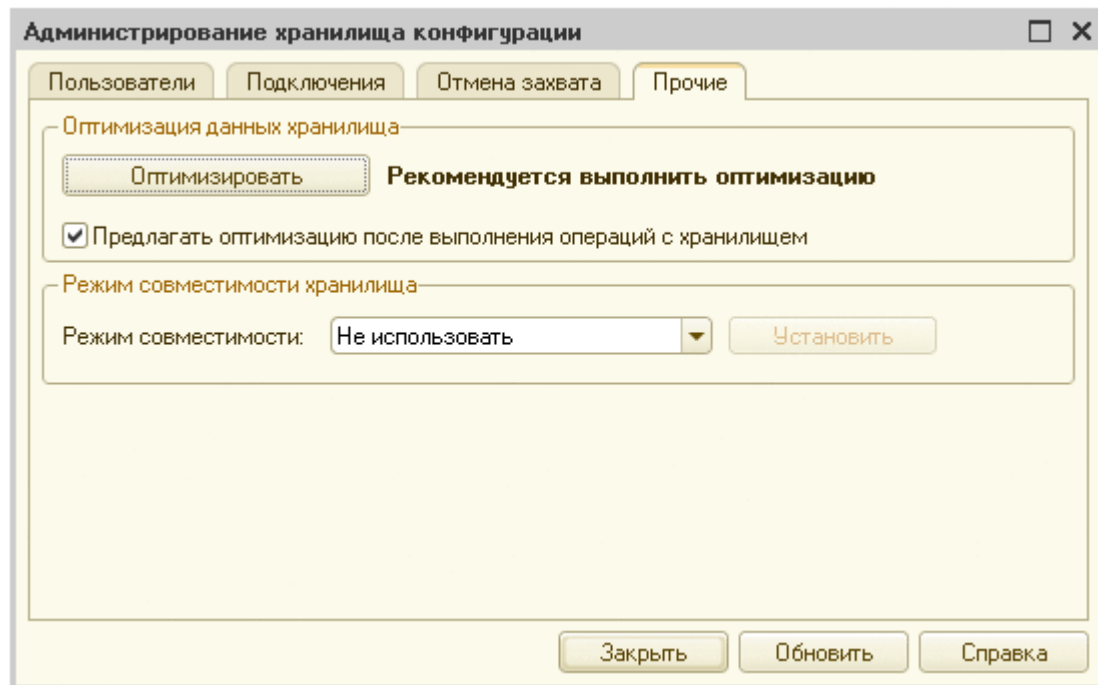


Рис. 479. Рекомендуется выполнить оптимизацию

Во время выполнения оптимизации работать с хранилищем не рекомендуется.

Оптимизацию можно также выполнить с помощью параметра командной строки запуска конфигуратора `/ConfigurationRepositoryOptimizeData`.

30.2.6. Резервное копирование хранилища

В зависимости от режима совместимости хранилища, резервное копирование выполняется по-разному.

ВНИМАНИЕ! Во время выполнения операции резервного копирования хранилища конфигурации, к нему не должно быть ни одного подключения.

Если хранилище работает в режиме совместимости (свойство [Режим совместимости](#) находится в значении [Версия 8.3.2](#)), то для выполнения резервной копии хранилища необходимо скопировать файл `1Cv8ddb.1CD`.

Если хранилище не использует режим совместимости (свойство [Режим совместимости](#) находится в значении [Не использовать](#)), то для выполнения резервной копии хранилища следует скопировать файл `1Cv8ddb.1CD` и **полностью каталог `data`**, расположенный в том же каталоге, что и файл `1Cv8ddb.1CD`.

30.3. Работа с хранилищем конфигурации

Захват, получение, помещение объектов в хранилище и другие действия с объектами хранилища удобнее выполнять в окне [Хранилище конфигурации](#). Для его открытия нужно выбрать пункт [Конфигурация – Хранилище конфигурации – Хранилище](#).

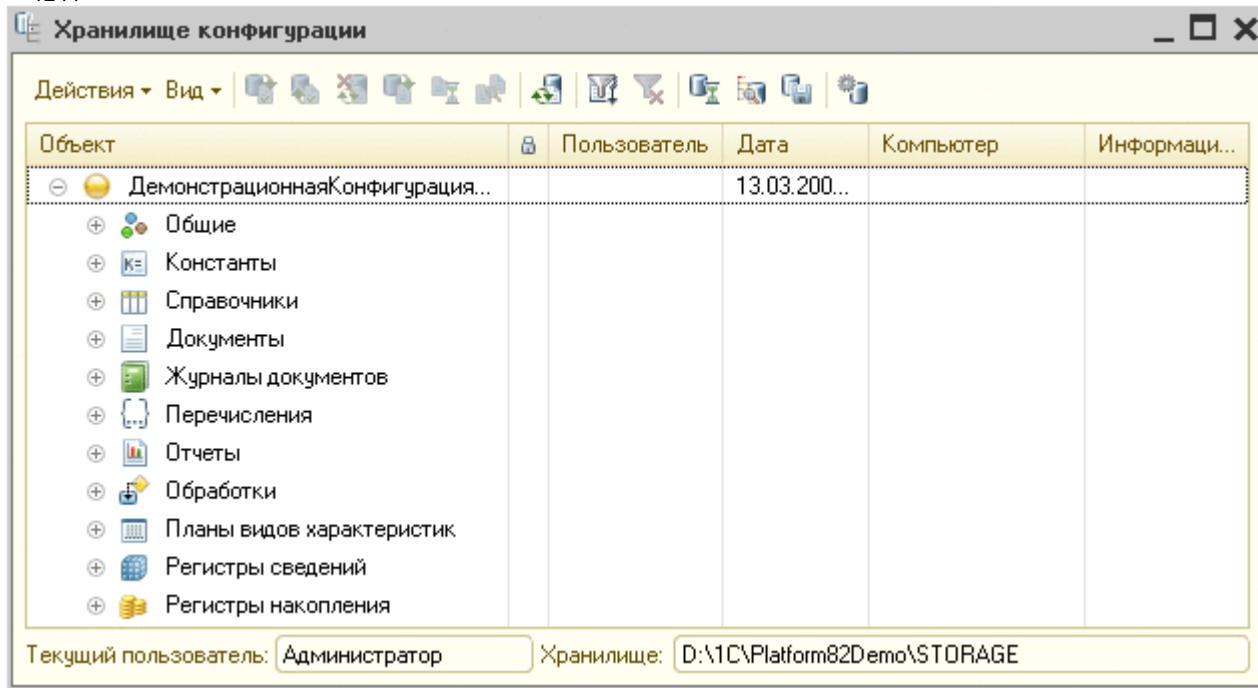


Рис. 480. Хранилище конфигурации

Окно [Хранилище конфигурации](#) состоит из панели инструментов и табличного поля, в первой колонке которого показывается иерархическая структура объектов хранилища или список объектов в зависимости от режима просмотра (меню [Вид](#)).

Если режим показа удаленных объектов установлен (подменю [Вид](#)), то в списке удаленные объекты показываются более светлым шрифтом.

Во второй колонке пиктограммами показывается состояние захваченных объектов. Захваченные текущим пользователем объекты показываются пиктограммой, содержащей красную галочку. В колонке [Пользователь](#) указывается пользователь, захвативший объект. В колонке [Дата](#) показывается дата захвата (если объект на данный момент захвачен) или дата последнего обновления объекта. В колонке [Компьютер](#) указывается имя компьютера пользователя, а в колонке [Информационная база](#) – имя локального каталога информационной базы на компьютере пользователя.

ВНИМАНИЕ! В окне хранилища не показываются созданные пользователем новые объекты (еще не помещенные в хранилище).

30.3.1. Фильтр хранилища

Для удобства работы с конфигурацией можно настроить показ объектов в графе [Объект](#) окна [Хранилище конфигурации](#). Фильтр вызывается пунктом меню [Действия – Отбор](#). На экран выводится окно:

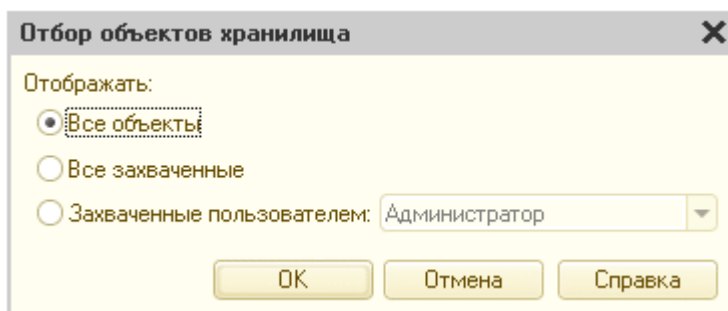


Рис. 481. Отбор объектов хранилища

Смысл элементов управления понятен из их названия.

Также имеется возможность быстро отобразить только те объекты, которые захвачены в хранилище текущим пользователем.

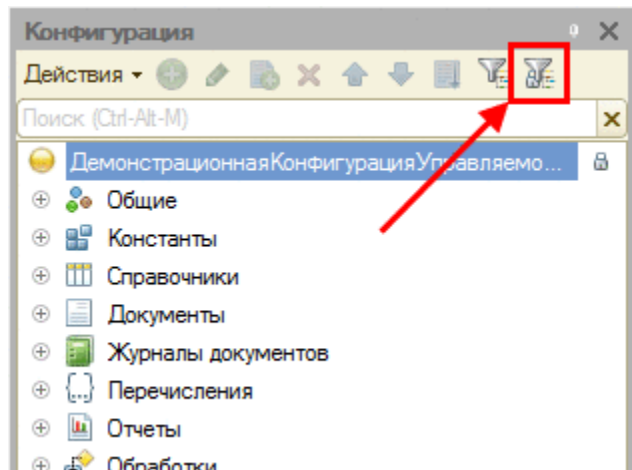


Рис. 482. Отбор захваченных объектов

Для этого необходимо воспользоваться командой **Действия – Захваченные объекты** или нажать специальную кнопку на командной панели окна **Конфигурация**. Кнопка отображается только в том случае, если конфигурация подключена к хранилищу конфигурации. Повторное выполнение команды отключает отбор. Отбор работает совместно с отбором по подсистемам (см. [здесь](#)) и поиском по дереву конфигурации (см. [здесь](#)).

30.3.2. Захват объектов хранилища

Захват объекта предназначен для монопольного владения объектом хранилища с целью его редактирования. Захваченный объект для других пользователей становится недоступным для изменений (возможен только просмотр и получение измененного объекта для использования).

Для захвата в окне **Хранилище конфигурации** следует указать нужный объект (допускается множественный выбор) и выполнить команду **Действия – Захватить в хранилище...**

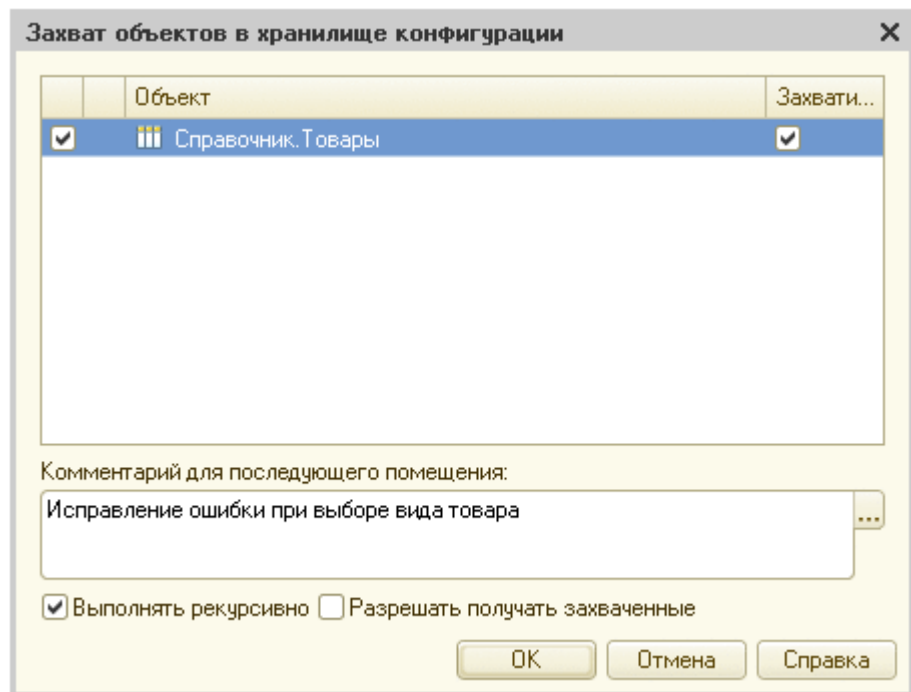


Рис. 483. Захват объектов в хранилище

Если объект связан с другими объектами, то в список будут также включены эти объекты. В этом случае выбранный объект показывается жирным шрифтом. Захват выбранного объекта возможен только в случае получения этих объектов из хранилища.

При этом в список включаются также те объекты, которые не связаны непосредственно с выбранным объектом, а имеют косвенную связь.

Для получения этих объектов необходимо установить флажок в первой колонке списка. Если также требуется захватить объекты, то для них нужно установить флажок в колонке [Захватить](#). Если не все требуемые для захвата объекты отмечены для получения, конфигуратор выдает сообщение [Для выполнения операции требуется получение объектов и перечисляет требуемые объекты](#). Захват в данном случае не производится.

Если объекты содержат подчиненные объекты и требуется захват подчиненных объектов, нужно установить флажок [Выполнять рекурсивно](#).

Если требуется получить данные уже захваченных объектов, то установить флажок [Разрешить получать захваченные](#).

ВНИМАНИЕ! Если флажок [Разрешить получать захваченные](#) установлен, изменения, сделанные в захваченных и повторно получаемых объектах, будут потеряны.

После захвата указанные объекты будут помечены специальным значком.

При захвате имеется возможность заранее внести комментарий, который может быть использован при помещении объекта в хранилище (см. [здесь](#)). Для этого нужно ввести текст в поле [Комментарий для последующего помещения](#) или выбрать такой комментарий из списка ранее внесенных комментариев (кнопка ... в правой части поля). В этом списке содержится 20 последних комментариев, которые вводились при захвате или помещении объектов.

Для добавления новых объектов в конфигурацию нужно захватить родительский объект. Например, для создания нового справочника необходимо захватить корневой объект конфигурации; для создания новой формы или реквизита справочника – захватить сам объект-справочник.

После захвата родительского объекта можно создавать подчиненные объекты. Затем все новые и захваченный родительский объекты нужно поместить в хранилище.

30.3.3. Помещение объекта в хранилище

Модифицированный объект можно поместить в хранилище, чтобы другие пользователи могли обновить этот объект в своих конфигурациях. Для этого нужно выполнить команду [Действия – Поместить в хранилище](#). На экран выводится диалог:

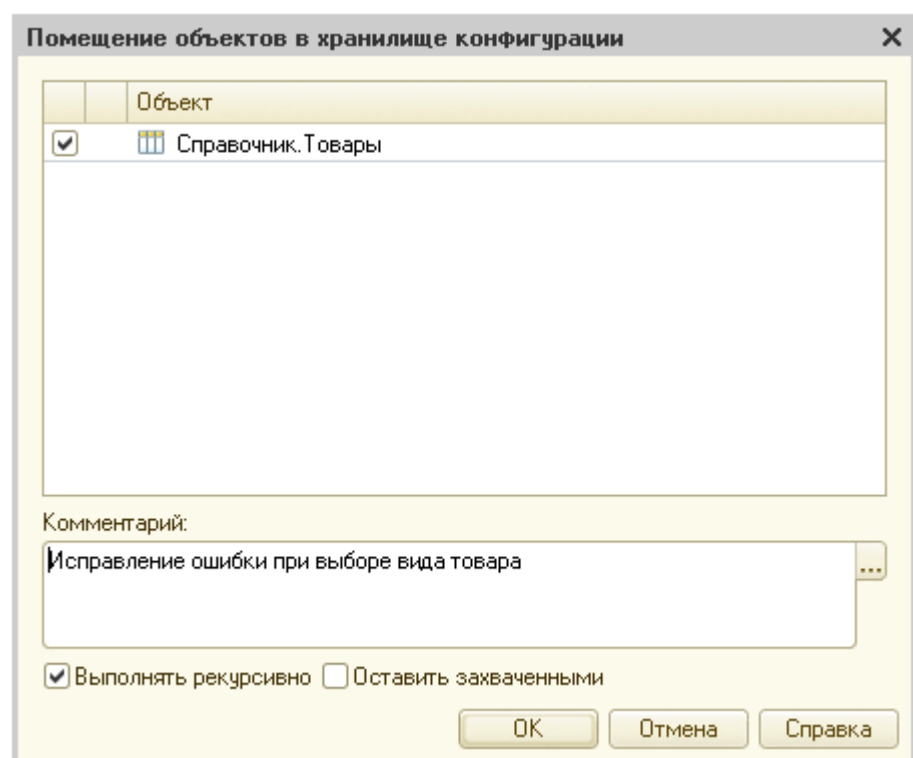


Рис. 484. Помещение объектов в хранилище

Необходимо установить флажки для тех объектов, которые будут помещены в хранилище.

Для описания произведенных изменений ввести их в многострочное поле [Комментарий](#). Если комментарий вводился при захвате объекта (объектов) в хранилище (см. [здесь](#)) или ранее использовался при помещении объекта в хранилище, этот комментарий можно выбрать для повторного использования из списка последних комментариев. Для открытия списка необходимо нажать кнопку "...", расположенную справа от поля [Комментарий](#). В этом списке содержится 20 последних комментариев, которые вводились при помещении или захвате объектов.

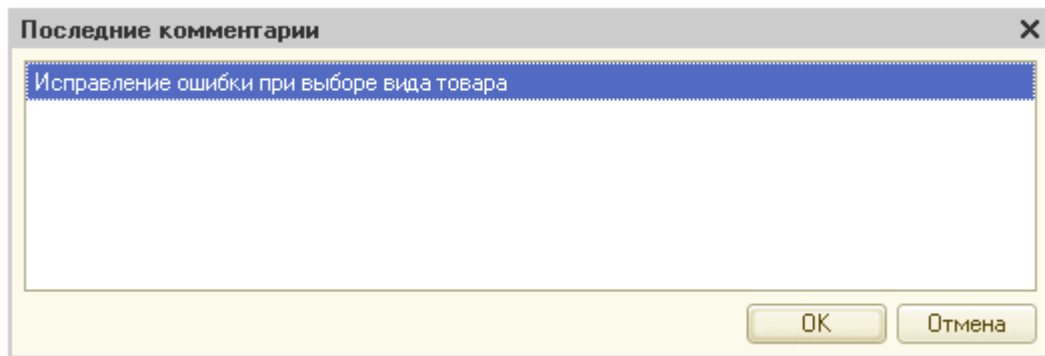


Рис. 485. Список последних комментариев

Если объекты содержат подчиненные объекты и требуется их поместить в хранилище, нужно установить флажок [Выполнять рекурсивно](#).

Если требуется, чтобы указанные объекты оставались захваченными, установить флажок [Оставить захваченными](#).

Если в хранилище помещаются объекты, связанные с другими объектами конфигурации, в диалог выводится полный список таких объектов.

Для того чтобы поместить в хранилище новые объекты, в окне [Конфигурация](#) нужно выбрать объект и в контекстном меню выбрать пункт [Поместить в хранилище...](#) На экран выводится окно помещения объектов в хранилище.

Помимо этого объекта в список могут быть включены и другие объекты, связанные с данным объектом. В этом случае в списке объектов выбранный объект выделен жирным шрифтом. Поместить объект в хранилище без помещения (обновления) этих объектов нельзя.

30.3.4. Получение объектов из хранилища

Чтобы получить объекты, измененные другими пользователями, в списке объектов хранилища следует указать нужные объекты и выполнить команду [Действия – Получить из хранилища](#). На экран выводится диалог:

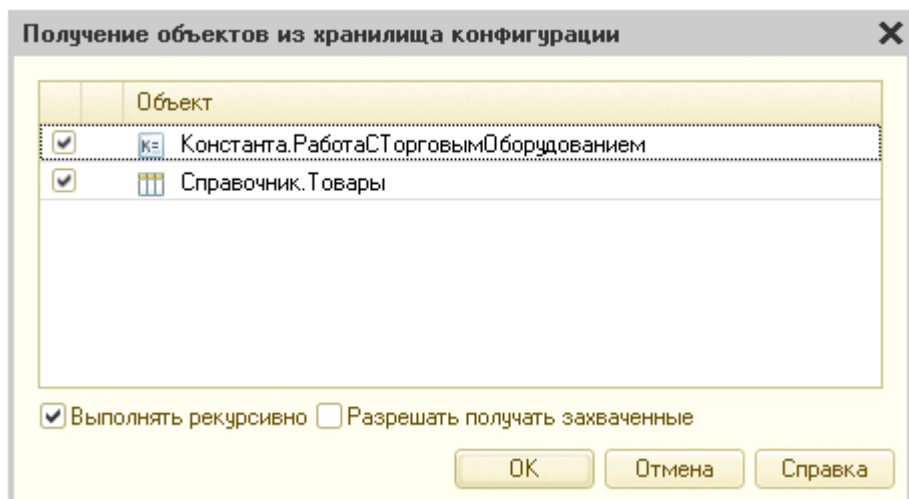


Рис. 486. Получение объектов из хранилища

В строках списка объектов нужно установить флажки для тех объектов, которые требуется получить из хранилища.

Если объекты содержат подчиненные объекты и требуется их получение, установить флажок [Получать рекурсивно](#).

Если выбранный объект уже захвачен, то установка флажка [Получать захваченные](#) приведет к потере произведенных изменений в данном объекте. В этом случае производится восстановление данных объекта данными из хранилища.

30.3.5. Отмена захвата объектов

Если не требуется сохранять произведенные в объекте хранилища изменения или нужно предоставить данный объект для модификации другому пользователю, данный объект можно отпустить. Для этого следует выполнить команду [Действия – Отменить захват в хранилище...](#)

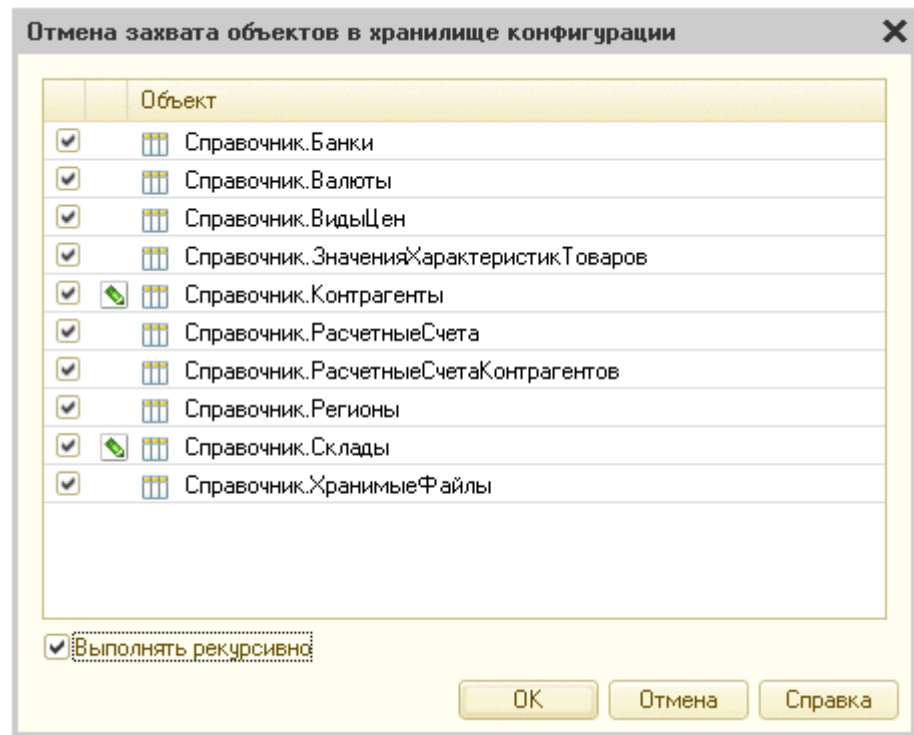


Рис. 487. Отменить захват объектов

Во второй колонке списка пиктограмма показывает отличие объекта в конфигурации и хранилище. Если пиктограмма отсутствует, то объекты совпадают.

Если объекты содержат подчиненные объекты и требуется отмена их захвата, нужно установить флажок [Выполнять рекурсивно](#).

После отмены захвата объекты будут получены из хранилища.

30.3.6. Обновление статуса объектов хранилища

Если требуется обновить информацию об объектах, размещенных в хранилище, нужно выполнить команду [Обновить статус](#). По этой команде производится обновление информации о захвате или отмене захвата, появлении новых объектов по всему хранилищу.

30.3.7. Сравнение хранилища с текущей конфигурацией

В любой момент времени можно произвести сравнение текущей конфигурации с хранилищем. Для этого нужно выполнить команду [Конфигурация – Хранилище конфигураций – Сравнить/объединить конфигурацию с хранилищем](#). На экран выводится окно сравнения (см. [здесь](#)).

30.3.8. Сравнение объекта хранилища с объектом текущей конфигурации

Иногда нужно произвести сравнение конкретного объекта в хранилище и в текущей конфигурации. Для этого в окне [Хранилище](#) конфигурации нужно выбрать объект и выполнить команду [Действия – Сравнить](#)

с объектом конфигурации.

На экран выводится вопрос: **Отображать сравнение конфигураций?** В случае утвердительного ответа производится сравнение конфигураций, выводится окно сравнения конфигураций, в окне сравнения активизируется строка с выбранным объектом, и на экран выводится диалог настройки отчета. В случае отрицательного ответа окно сравнения конфигураций на экран не выводится и сравнение не выполняется.

Если в хранилище отключен режим совместимости с версией 8.3.2 (см. [здесь](#)), то имеется возможность выполнить выборочное сравнение двух объектов конфигурации, не выполняя сравнение все конфигурации. При этом для выборочного сравнения будут доступны только версии объектов, которые помещались в хранилище с помощью «1С:Предприятия» версии 8.2.9 и старше. Для выборочного сравнения следует использовать команду контекстного меню **Сравнить выборочно с объектом хранилища**, **Сравнить выборочно с объектом конфигурации** или **Сравнить выборочно версии объекта**, в зависимости от того, откуда требуется выполнить сравнение: из дерева конфигурации, дерева хранилища или из окна истории хранилища. При выборе выборочного сравнения из дерева конфигурации, сравнение будет выполнено с последней версией объекта, расположенной в хранилище конфигурации. Если выборочное сравнение будет недоступно – платформа сообщит об этом с помощью диагностического сообщения.

Возможность проводить выборочное сравнение реализована для следующих объектов конфигурации: модуль, макет, картинка, WS-ссылка, расписание, агрегаты регистра, состав плана обмена, стиль, рабочая область начальной страницы, карта маршрута, форма (без учета модуля формы), модуль формы, справочная информация, предопределенные элементы, фрагмент командного интерфейса, права роли.

30.3.9. Сохранение хранилища как конфигурации

Сохранение хранилища как конфигурации производится с помощью команды **Конфигурация – Хранилище конфигурации – Сохранить конфигурацию хранилища в файл**.

30.3.10. Отключение от хранилища

Для отключения от хранилища нужно выбрать пункт **Конфигурация – Хранилище конфигурации – Отключиться от хранилища**.

30.3.11. История хранилища

Система групповой разработки конфигурации поддерживает ведение истории изменения информации об объектах конфигурации. В истории записываются результаты помещения объектов.

Каждое изменение объекта можно просмотреть, сравнить с текущим состоянием, сравнить версии конфигурации, а также получить конфигурацию указанной версии из хранилища.

Для открытия окна истории хранилища следует выбрать пункт **Конфигурация – Хранилище – История хранилища**.

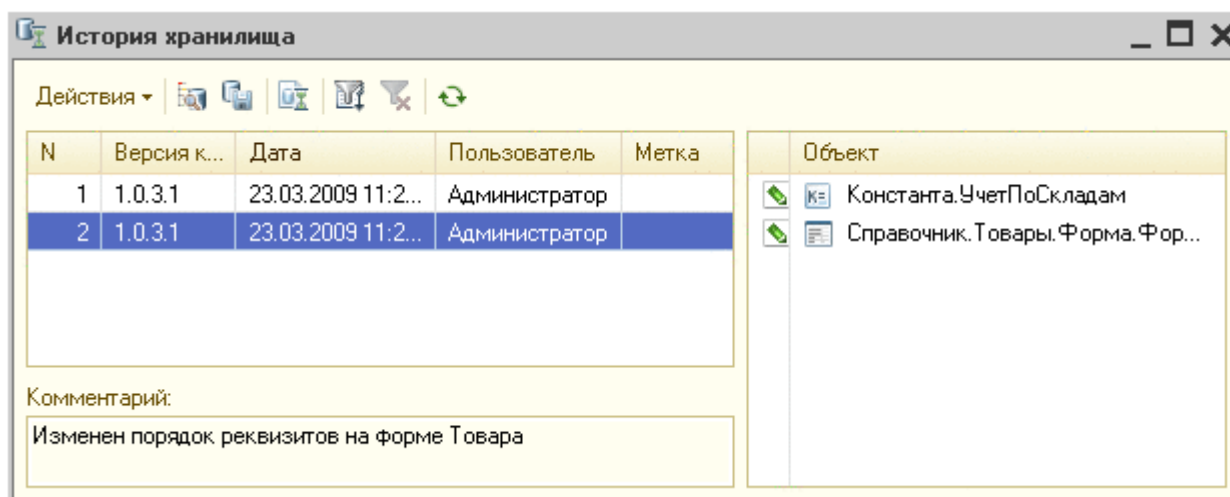


Рис. 488. История хранилища

В табличном поле, расположенном в левой части окна, выводится хронологический список версий конфигурации.

Ниже списка версий расположено поле для показа комментария по выбранной версии.

В табличном поле, расположенном в правой части окна, выводится список объектов, которые были изменены, добавлены или удалены в выбранной версии.

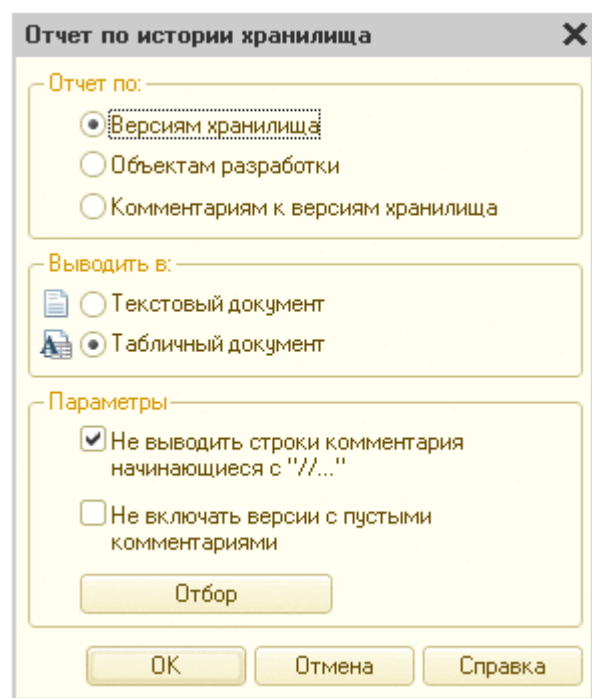
В списке версий допускается множественный выбор версий. В этом случае в поле **Комментарий** выводятся комментарии всех выбранных версий, а в списке объектов – список всех объектов по данным версиям.

Каждый объект списка объектов окна **История хранилища** можно открыть для просмотра, посмотреть историю изменений данного объекта и сравнить с объектом текущей конфигурации. Эти действия выполняются с помощью команд контекстного меню списка объектов.

30.3.11.1. Работа с версией хранилища

Каждую версию хранилища конфигурации можно открыть для просмотра, загрузить вместо текущей конфигурации, сравнить с текущей конфигурацией, сохранить в файл на диске. Также предоставлена возможность сравнить версии хранилища конфигурации с конфигурацией, хранящейся в файле. Все действия выполняются с помощью команд меню **Действия** окна **История хранилища**.

Чтобы вывести информацию по истории хранилища, можно использовать специальный отчет. Для запуска отчета в окне **История хранилища** следует выбрать пункт **Действия – Отчет по истории хранилища**. На экран выводится окно для указания параметров отчета.

**Рис. 489. Параметры отчета по истории хранилища**

В группе **Отчет по** выбирают вид группировки информации в отчете. В каждой группе информация сортируется по алфавиту.

Если выбран вид **Версиям хранилища**, то информация группируется по номерам версий. Если выбран вид **Объектам разработки**, то информация группируется по объектам. Если выбран вид **Комментариям к версиям**, то информация группируется по комментариям в истории хранилища.

В группе **Выводить в** выбирают вид представления отчета (текстовый или табличный документ).

В группе **Параметры** указываются дополнительные параметры отчета.

Для исключения строк комментария, начинающихся с «//», следует установить соответствующий флажок.

Чтобы указать объекты, по которым требуется выполнить отчет, и интервал дат, следует нажать кнопку **Отбор**. В открытом окне задания отбора установить параметры отбора.

Для начала построения отчета нажать кнопку **ОК**.

30.3.11.2. Возврат к предыдущим версиям

Если в хранилище конфигурации были опубликованы ненужные версии, конфигуратор предоставляет возможность отката до нужной версии хранилища. Для этого в списке версий нужно выбрать ту, до которой следует откатиться, и выбрать пункт **Действия – Откатить до версии**. На экран выводится вопрос-предупреждение: **При выполнении отката информация об откатываемых версиях будет удалена без возможности восстановления. Продолжить?** В случае утвердительного ответа будет произведен откат, при котором из хранилища удаляются все версии, помещенные после указанной.

30.3.11.3. Сокращение версий

Если ранние версии конфигураций становятся ненужными, их можно удалить из хранилища. Для этого в списке версий следует выбрать ту, до которой нужно сократить, и выбрать пункт **Действия – Сократить до версии**. На экран выводится вопрос-предупреждение: **При выполнении информация об удаляемых версиях будет уничтожена без возможности восстановления. Продолжить?** В случае утвердительного ответа будет произведено сокращение, при котором из хранилища удаляются все версии, помещенные до указанной.

30.3.11.4. Объединение версий

Несколько последовательных версий хранилища можно объединять. Для этого в окне **История хранилища** нужно выделить объединяемые версии (при этом в разделе **Комментарий** формируется сводный комментарий, состоящий из комментариев версий, а в списке объектов показываются все объекты, над которыми были выполнены любые действия, выбранных версий) и выбрать пункт **Действия – Объединить версии**.

В результате в истории хранилища будет сформирована одна версия, включающая сводные изменения указанных версий.

30.3.11.5. Установка меток версий

Для каждой версии конфигурации в окне **История хранилища** можно установить метку. Для этого следует выбрать нужную версию и в контекстном меню выбрать пункт **Изменить**. На экран выводится диалог.

Версия хранилища

Номер версии: 2
Версия конфигурации: 1.0.3.1
Дата: 23.03.2009 11:29:59
Пользователь: Администратор
Метка: Релиз 1

Комментарий:
Изменен порядок реквизитов на форме Товара

Комментарий метки:
Конфигурация подготовлена к выпуску

OK Отмена Справка

Рис. 490. Версия хранилища

Нужно ввести значение метки. При вводе первого символа становится доступным поле [Комментарий метки](#), в которое записывается текст, описывающий назначение данной метки. Метки в основном играют роль отметок завершения определенных логических этапов или действий (выпуск новых версий конфигураций, подготовка поставки и др.).

30.3.11.6. Отбор

Диалог отбора списка [История хранилища](#) отличается от стандартного.

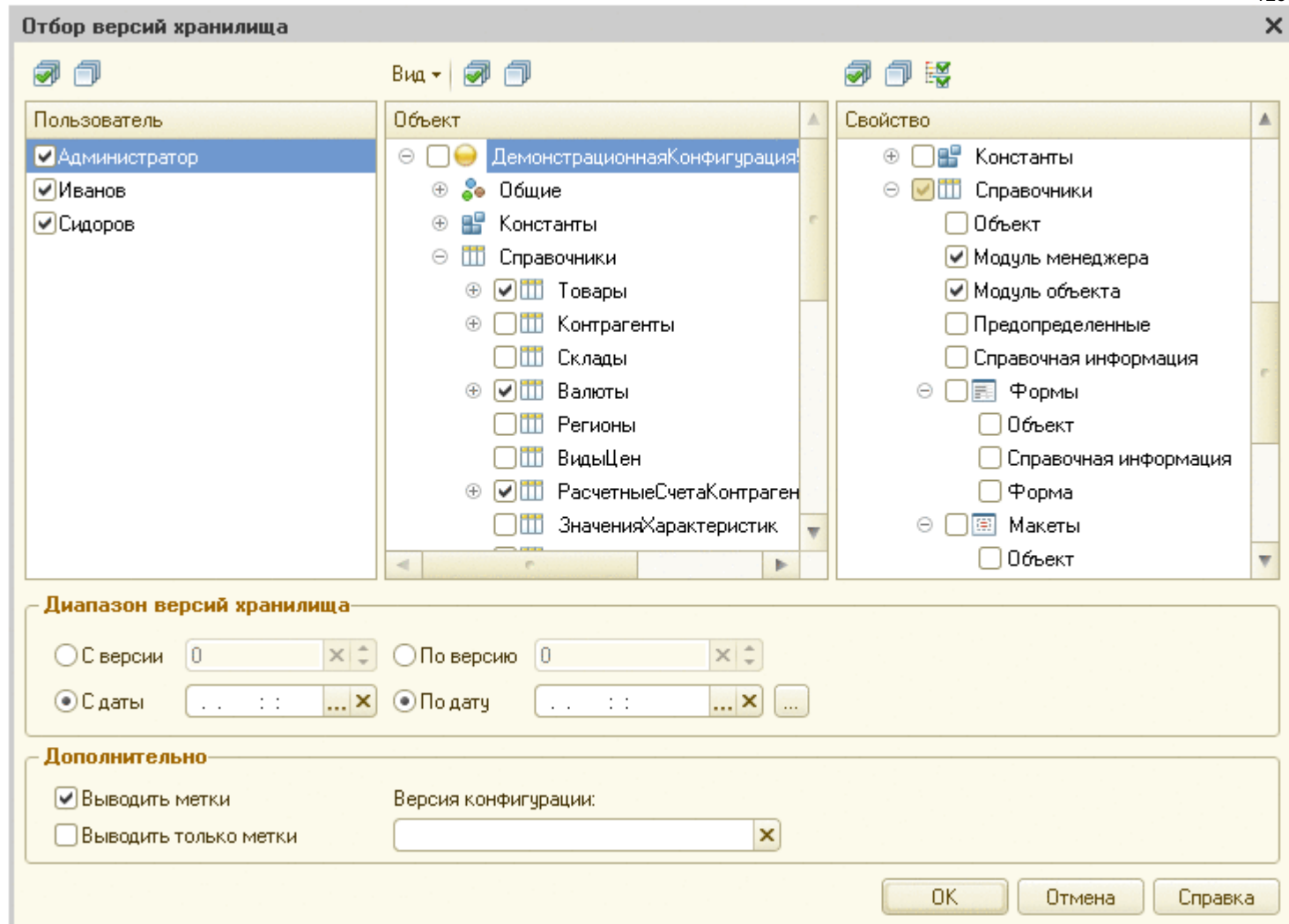


Рис. 491. Отбор версий

Отбор версий выполняется в интервале времени, в интервале версий хранилища (или в комбинации версии и даты), по указанному списку пользователей хранилища, выполнивших размещение версий, по определенным объектам конфигурации, по определенной версии конфигурации. Также имеется возможность устанавливать отбор по конкретным свойствам объектов конфигурации. Для этого предназначено дерево свойств в окне установки отбора. Так, в примере на [рис. 491](#), для всех выбранных справочников (*Товары*, *Валюты*, *РасчетныеСчетаКонтрагентов* в дереве *Объект*) будут отображены только те версии, где у этих справочников изменялись модули менеджеров и модули объектов (в дереве *Свойство*). Для упрощения выбора свойств в командной панели дерева *Свойство* присутствует кнопка **Отметить все свойства выбранных объектов**. С помощью этой кнопки выбираются все свойства для тех объектов конфигурации, которые отмечены в дереве объектов. Возможность выполнять отбор истории реализован по следующим свойствам объектов конфигурации: модуль, макет, картинка, WS-ссылка, расписание регламентного задания, агрегаты оборотного регистра накопления, состав плана обмена, стиль, рабочая область начальной страницы, карта маршрута, форма, справочная информация, предопределенные элементы, фрагмент командного интерфейса, права роли.

При выборе в дереве свойств элемента *Объект*, в историю попадут те версии, где изменялись свойства соответствующего объекта конфигурации, для которых недоступно выборочное сравнение.

Кроме того, можно указать необходимость вывода версий, помеченных метками, а также включения в выборку только тех версий, которые помечены метками.

По каждой выбранной версии списка версий в левой части окна *История хранилища* содержится список объектов, которые были добавлены, изменены или удалены. Этот список всегда полный для данной версии (не зависит от условий отбора). Если в условиях отбора указываются конкретные объекты конфигурации (не вся конфигурация), то в списке объекты, удовлетворяющие условию отбора, выделяются другим цветом фона.

30.3.11.7. Сравнение версий хранилища

Иногда нужно произвести сравнение версий конфигурации хранилища. Для этого в окне [История хранилища](#) нужно выбрать две сравниваемые версии (выбор производится с нажатой клавишей **Ctrl**) и выполнить команду [Действия – Сравнить версии конфигурации](#) или [Действия – Сравнить выборочно версии объекта](#). Подробнее о выборочном сравнении см. [здесь](#).

Производится сравнение указанных версий конфигураций друг с другом, и выводится окно сравнения конфигураций.

30.3.12. История объекта

Для просмотра истории изменения объекта в списке объектов хранилища или в окне [Конфигурация](#) нужно указать объект и выбрать пункт [Действия – История объекта](#) или с помощью меню [Действия – Выборочная история объекта](#). Для каждого объекта доступен собственный набор свойств, по которым можно просматривать историю.

После выбора вида просмотра, на экран выводится диалог отбора истории.

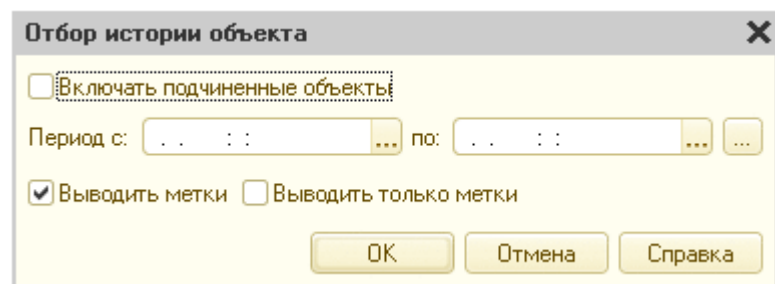


Рис. 492. Отбор истории объекта

Следует указать нужные параметры отбора и нажать кнопку **OK**.

На экран выводится окно истории объекта.

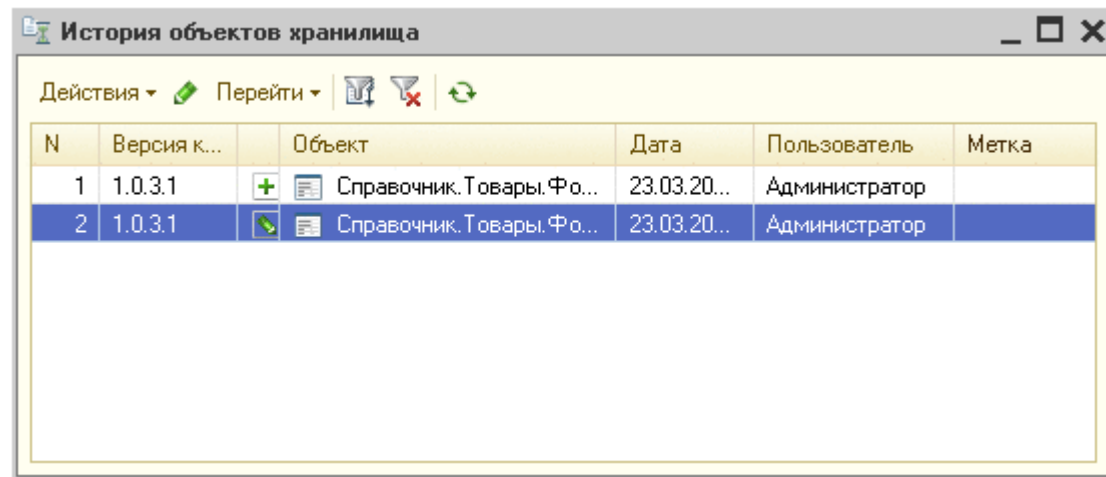


Рис. 493. История объекта

Каждое изменение объекта можно просмотреть, сравнить с текущим состоянием, сравнить версии (для этого необходимо в списке выбрать две версии).

По каждой версии объекта можно просмотреть комментарии и метки аналогично версиям конфигурации (см. выше), а также получить конфигурацию указанной версии из хранилища.

В списке версий объекта можно выполнить отбор. Отбор выполняется аналогично описанному выше отбору в истории хранилища. При этом в окне отбора версий будет выбран текущий объект и соответствующие свойства: все, если был выбран просмотр истории объекта или указанное, если был выбран просмотр выборочной истории.

Если требуется перейти в окно истории хранилища, то в подменю [Перейти](#) окна истории объектов нужно выбрать пункт [История хранилища](#). Если требуется открыть окно истории хранилища с одновременной

установкой отбора по текущему объекту, то следует выбрать пункт [История хранилища по объекту](#).

30.3.13. Выполнение действий с хранилищем без подключения

Некоторые действия с хранилищем можно выполнять без подключения.

Если конфигурация не подключена к хранилищу, то предварительно нужно установить соединение с хранилищем. Для этого следует выбрать любой доступный пункт из группы [Конфигурация – Хранилище конфигурации](#). На экран выводится диалог соединения с хранилищем.

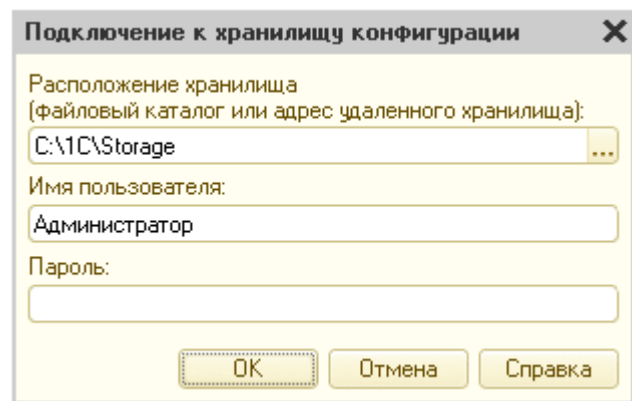


Рис. 494. Подключение к хранилищу

Нужно указать те же данные, что и при подключении, и нажать кнопку **ОК**.

После установки соединения при выборе других доступных пунктов меню диалог соединения не выводится.

В режиме соединения доступны все действия, связанные с просмотром данных хранилища, сравнением объектов и конфигураций, а также администрирование хранилища в полном объеме (для пользователей, имеющих административные права). Недоступны действия, связанные с получением объектов из хранилища.

В режиме соединения пользователи, имеющие административные права, могут выполнять все действия, связанные с историей хранилища.

30.4. Удаленная работа с хранилищем конфигурации

Механизм удаленной работы с хранилищем конфигурации расширяет возможности групповой разработки конфигурации:

- хранилище конфигурации может быть расположено на компьютере, работающем под управлением операционных систем Windows или Linux;
- доступ к хранилищу конфигурации могут иметь пользователи локальной сети (по протоколу tcp) и пользователи сети Интернет (по протоколу http).

Для прикладного разработчика удаленная работа с хранилищем конфигурации практически ничем не отличается от работы с хранилищем, расположенным на общем сетевом ресурсе, за исключением некоторых особенностей, описание которых см. [здесь](#).

ПРИМЕЧАНИЕ. Механизм удаленной работы с хранилищем ориентирован в основном на работу с соединением с высокой пропускной способностью. Кроме этого, следует учитывать, что при использовании удаленной работы может существенно возрасти время операций, требующих обмена большим количеством информации, например таких, как получение полной версии хранилища.

30.4.1. Обзор архитектуры

Удаленную работу с хранилищем конфигурации обеспечивает специальное приложение – **сервер хранилища конфигурации**. Компьютер, на котором функционирует приложение сервера хранилища

конфигурации (**crserver.exe** для Windows или **crserver** для Linux), также называется **сервер хранилища конфигурации**. В зависимости от протокола работы (TCP или HTTP) используются различные схемы удаленного взаимодействия клиентского приложения и хранилища конфигурации. При этом обеспечивается одновременная работа с хранилищем конфигурации как удаленно, так и «локально», как с общим файловым ресурсом.

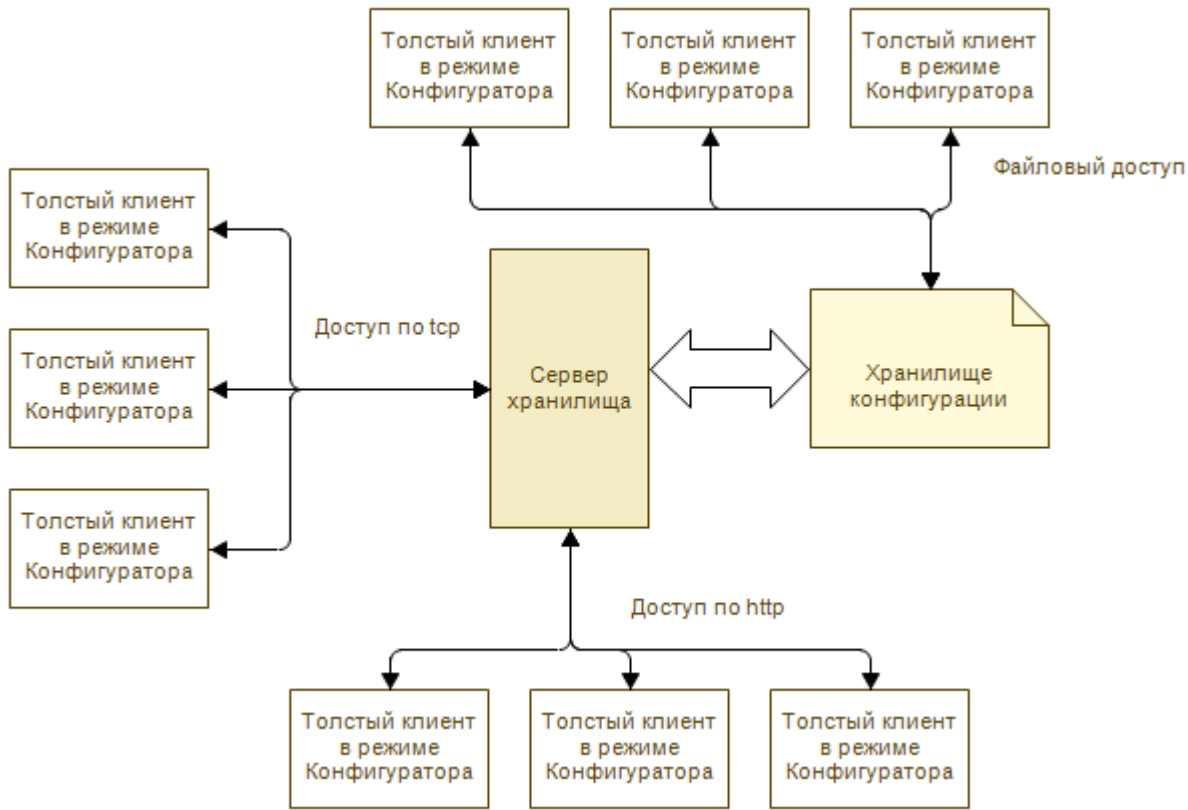


Рис. 495. Удаленное хранилище

30.4.1.1. Общая схема работы с хранилищем конфигурации

В общем случае система «1С:Предприятие» обеспечивает одновременную работу с хранилищем конфигурации с использованием всех протоколов доступа к хранилищу: файловый доступ, протокол TCP и протокол HTTP.

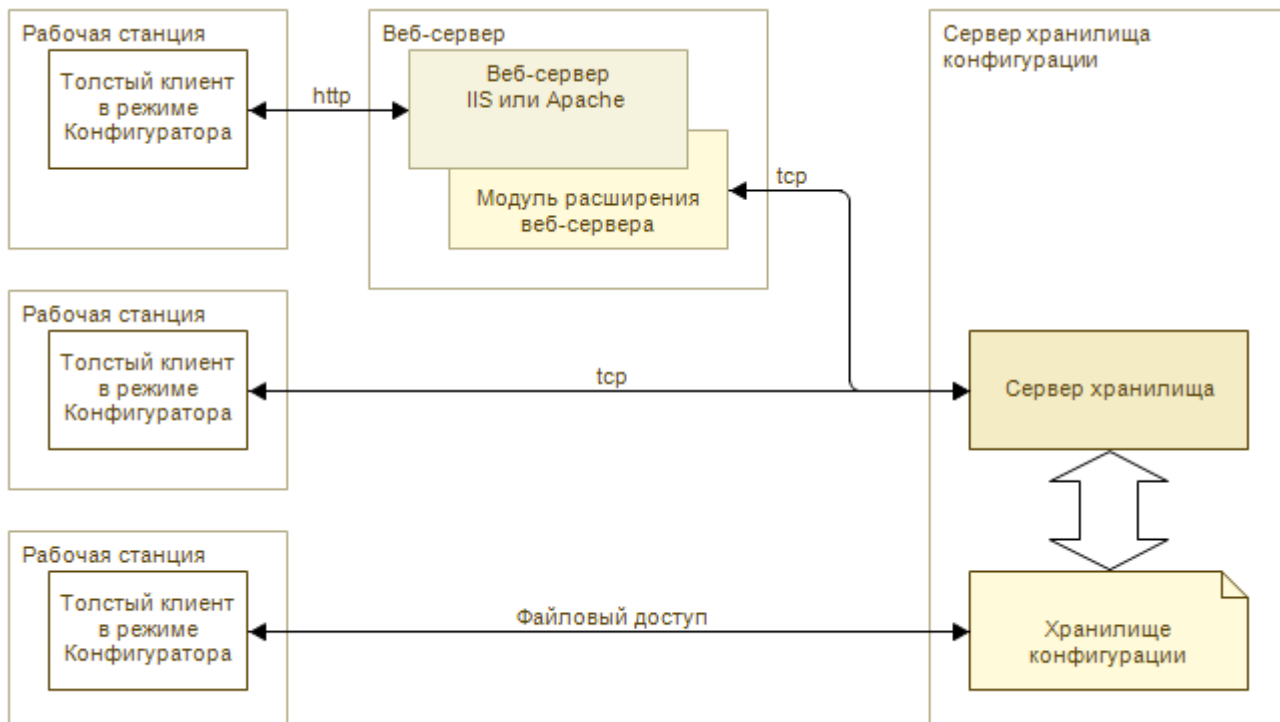
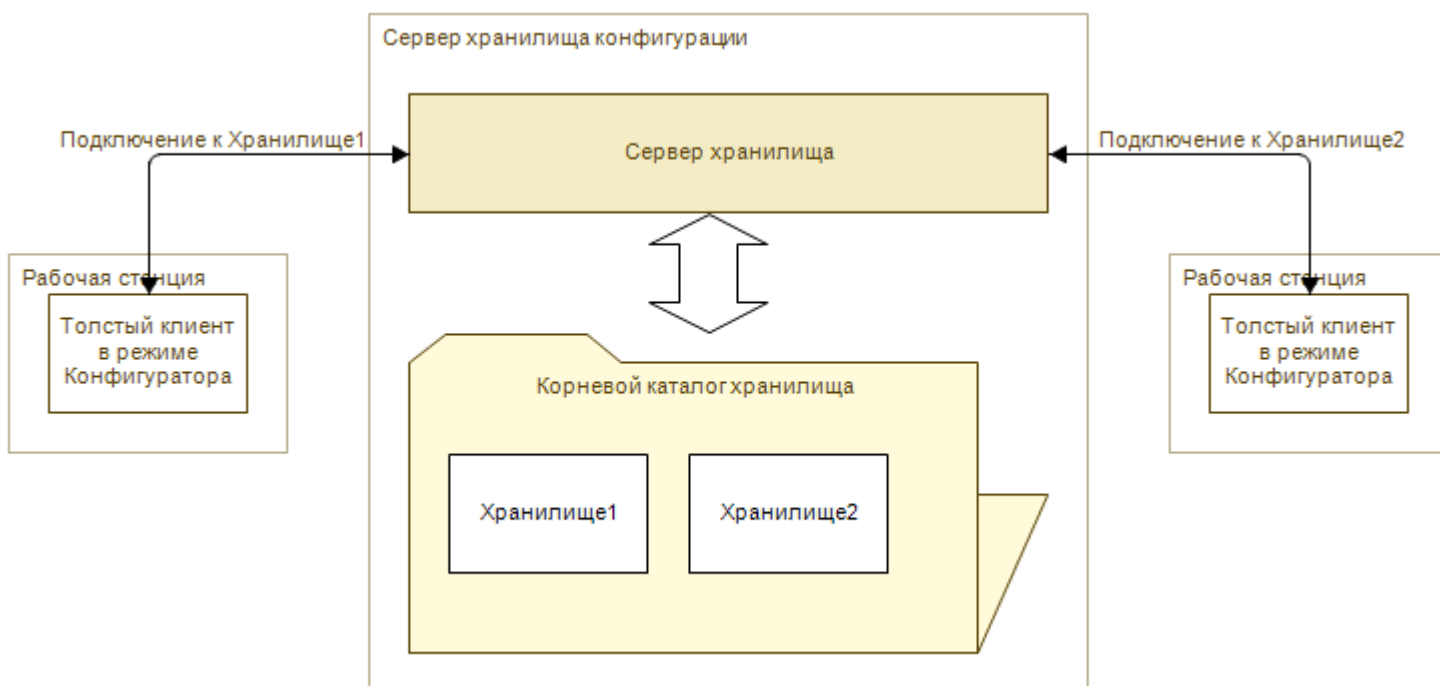


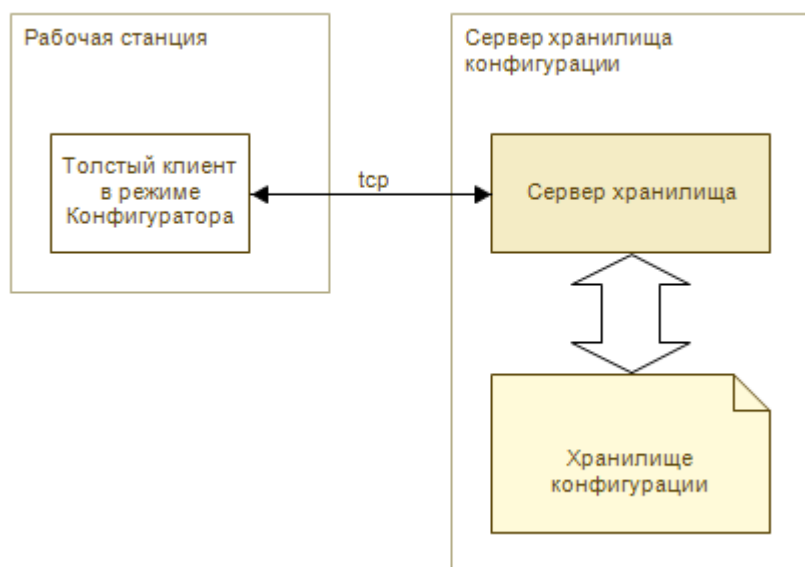
Рис. 496. Взаимодействие с удаленным хранилищем

При этом один и тот же сервер хранилища конфигурации позволяет работать клиентам с различными хранилищами конфигурации. Единственное ограничение – все хранилища должны быть расположены в одном корневом каталоге.

**Рис. 497. Расположение различных файлов хранилищ**

30.4.1.2. Взаимодействие по протоколу TCP

Когда используется протокол TCP, клиентское приложение взаимодействует с сервером хранилища конфигурации, который, в свою очередь, взаимодействует непосредственно с самим хранилищем конфигурации.

**Рис. 498. Взаимодействие по протоколу TCP**

30.4.1.3. Взаимодействие по протоколу HTTP

Когда используется протокол HTTP, взаимодействие осуществляется по следующей цепочке: клиентское приложение – веб-сервер – сервер хранилища конфигурации – хранилище конфигурации.

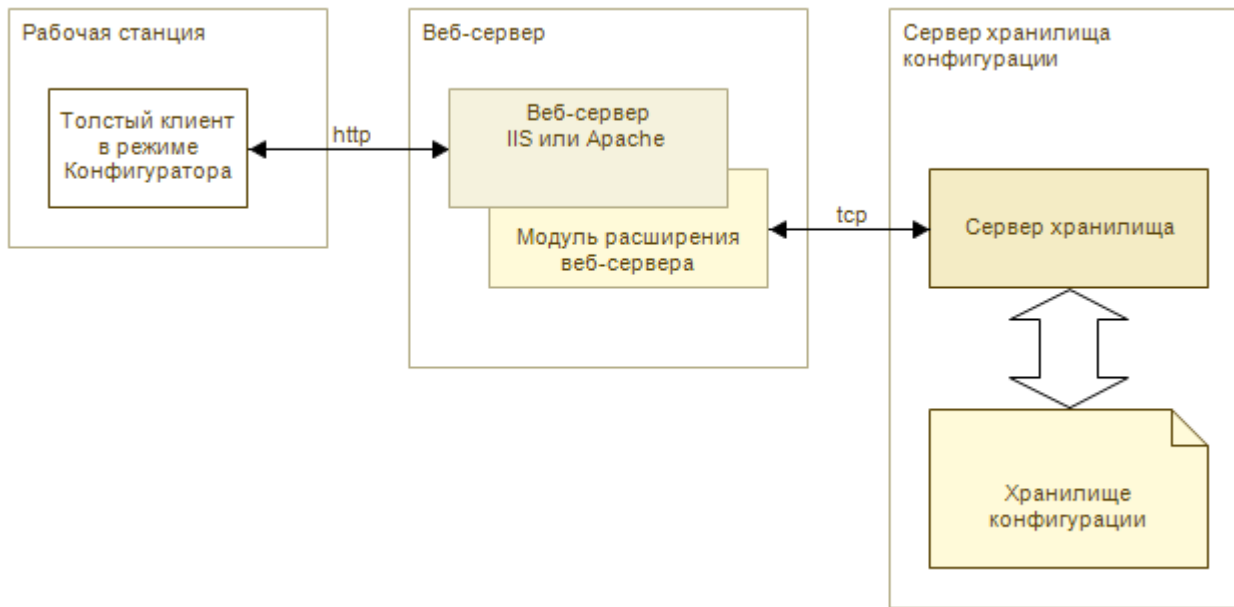


Рис. 499. Взаимодействие по протоколу HTTP

30.4.2. Установка сервера хранилища

Установка сервера хранилища конфигурации описана в книге «1С:Предприятие 8.3. Руководство администратора».

После установки сервера хранилища конфигурации необходимо для пользователя, от имени которого работает сервер хранилища конфигурации, права записи и чтения на каталог, в котором расположено хранилище конфигурации (файл [1Cv8DDB.1CD](#) и каталог [data](#)).

30.4.3. Настройка веб-сервера для работы с хранилищем

Если планируется удаленная работа с хранилищем по протоколу HTTP, то кроме настройки сервера хранилища конфигурации необходимо настроить веб-сервер на работу с системой «1С:Предприятие» и сервером хранилища конфигурации. Настройка заключается в следующих действиях:

- Зарегистрировать модуль расширения веб-сервера;
- Зарегистрировать виртуальное приложение на веб-сервере;
- Опубликовать конфигурационный файл работы с сервером хранилища конфигурации.

ПРИМЕЧАНИЕ. На веб-сервере **обязательно** должна быть установлена поддержка **ISAPI расширений**.

Публикация конфигурационного файла работы с сервером хранилища конфигурации сводится к размещению файла с расширением [1ccr](#) в виртуальном каталоге веб-сервера. Например, такой конфигурационный файл может иметь имя [storage.1ccr](#) (описание файла см. книгу «1С:Предприятие 8.3. Руководство администратора») и следующее содержание:

[Копировать в буфер обмена](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<storage connectString="tcp://RepServ"/>
```

Для настройки веб-сервера необходимы права администратора.

ПРИМЕЧАНИЕ. Сервер хранилища является 32-разрядным приложением, поэтому при работе через веб-сервер следует использовать или 32-разрядные операционные системы и веб-серверы или разрешить веб-серверу Microsoft Internet Information Services исполнять 32-разрядные модули.

30.4.3.1. Для веб-сервера Microsoft Internet Information Services

Данный раздел содержит настройки веб-сервера Microsoft Internet Information Services (далее IIS) в зависимости от версии веб-сервера.

30.4.3.1.1. Для IIS версий 5.1, 6.0

Необходимо создать виртуальный каталог. Для этого необходимо использовать мастер создания виртуального каталога из контекстного меню пункта **Веб-узел по умолчанию (Internet Information Services – <Имя компьютера с веб-сервером> – Веб-узлы – Веб-узел по умолчанию)**. В контекстном меню необходимо выбрать пункт **Создать – Виртуальный каталог ...**.

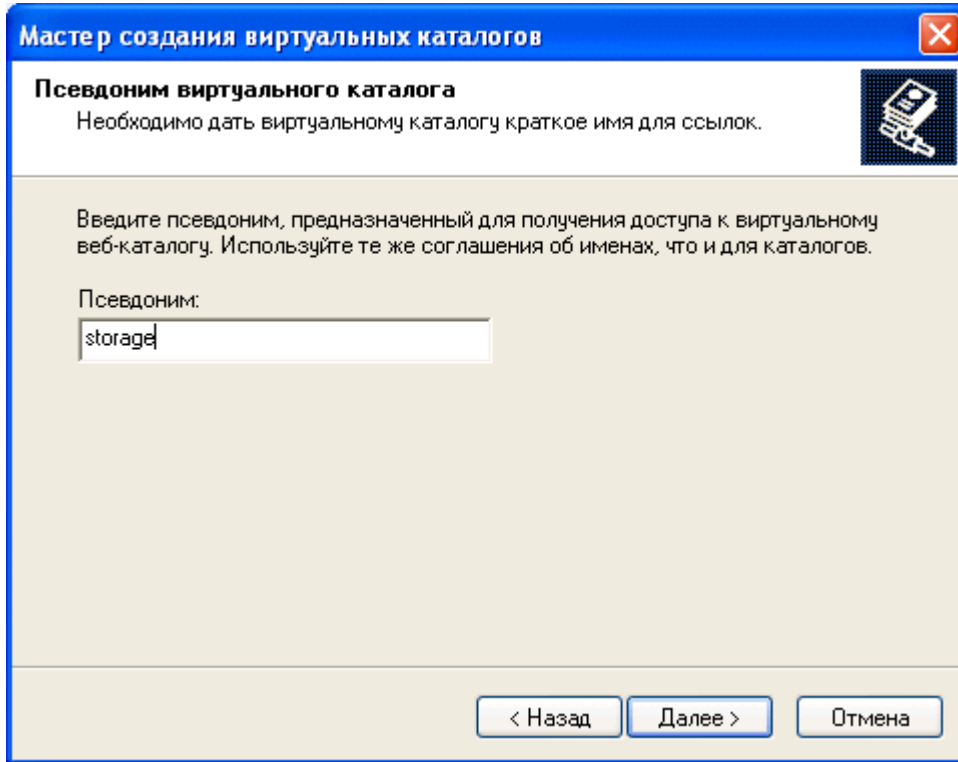


Рис. 500. Задание имени виртуального каталога

Имя, указанное в поле **Псевдоним**, будет использоваться при формировании URL доступа с хранилищу. Затем необходимо указать путь к папке, содержащей файлы создаваемого каталога.

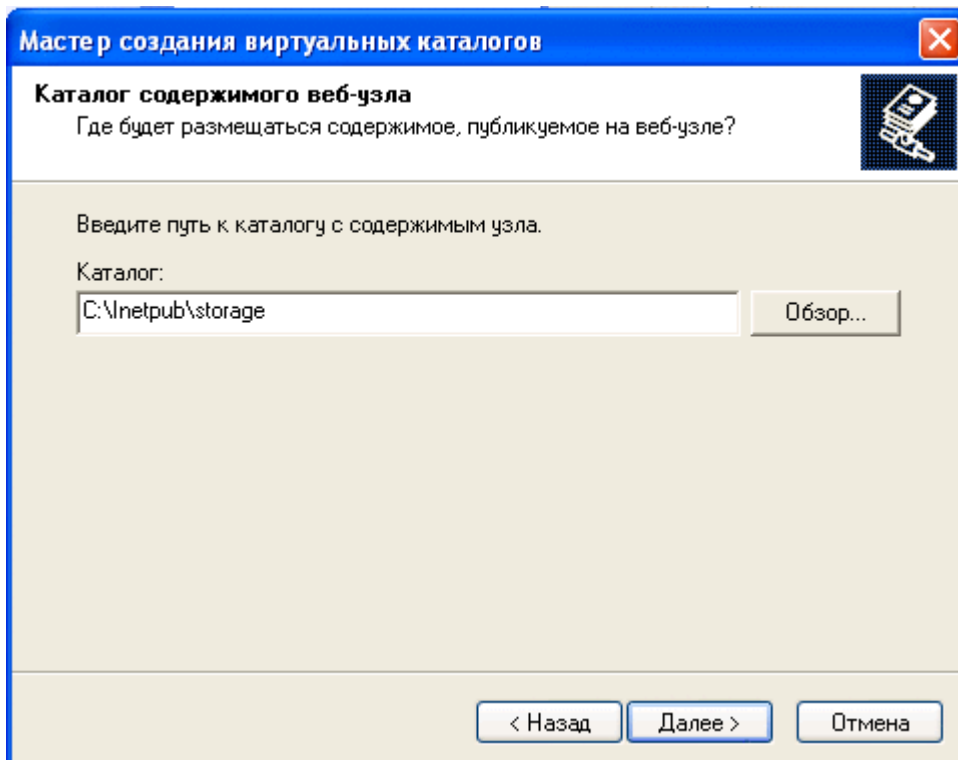


Рис. 501. Указание каталога

На следующем шаге необходимо указать, какие действия необходимо разрешить при работе с каталогом. Следует указать возможность выполнять **чтение** и **выполнение** (например приложений **ISAPI** или **CGI**). Остальные права следует выключить.

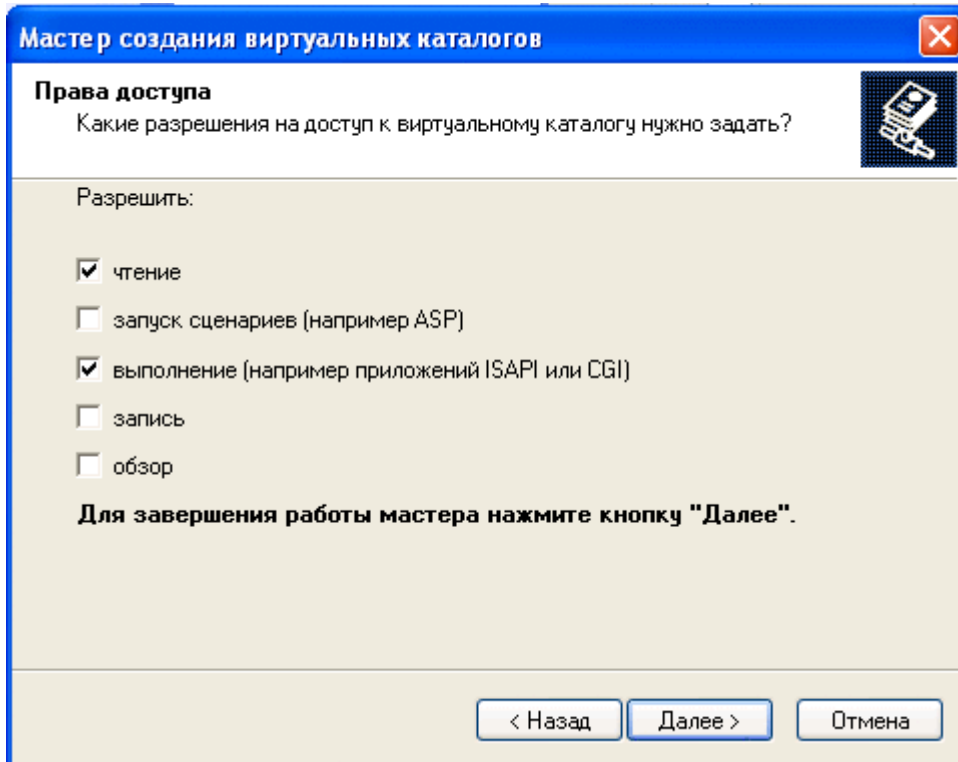


Рис. 502. Установка разрешений

На следующем шаге мастера необходимо нажать кнопку **Готово**. На этом создание виртуального каталога завершено. Следующим шагом необходимо настроить созданный каталог и зарегистрировать для него расширение веб-сервера. Для этого надо открыть свойства созданного виртуального каталога и в открывшемся диалоге нажать кнопку **Настройка**.

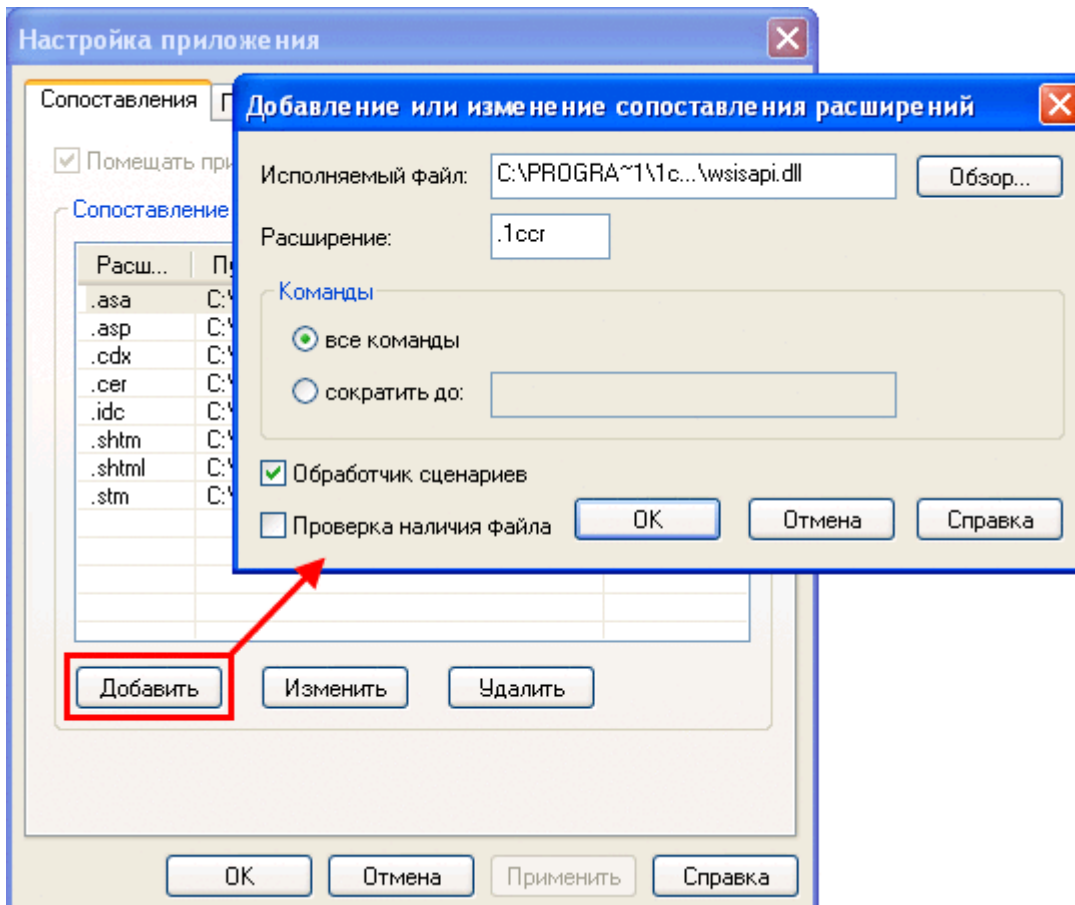


Рис. 503. Регистрация расширения веб-сервера

В качестве расширения веб-сервера IIS выступает динамическая библиотека [wsisapi.dll](#), расположенная в каталоге [bin](#) конкретной версии «1С:Предприятия».

СОВЕТ. Для регистрации расширения веб-сервера используйте или короткий путь («формат 8.3») или устанавливайте «1С:Предприятие» в каталог, не содержащий пробелов и имен каталогов, длиннее 8 символов.

В качестве расширения (поле **Расширение:** в диалоге на [рис. 503](#)) можно указать как конкретное расширение ([.1ccr](#)), так и маску всех файлов (*).

Последним шагом необходимо перезапустить веб-сервер.

30.4.3.1.2. Для IIS версий 7.0, 7.5

Необходимо создать виртуальное приложение. Для этого необходимо использовать мастер создания виртуального приложения из контекстного меню пункта [Default Web Site](#) (<Имя компьютера с веб-сервером> – сайты – [Default Web Site](#)). В контекстном меню необходимо выбрать пункт [Добавить приложение ...](#).

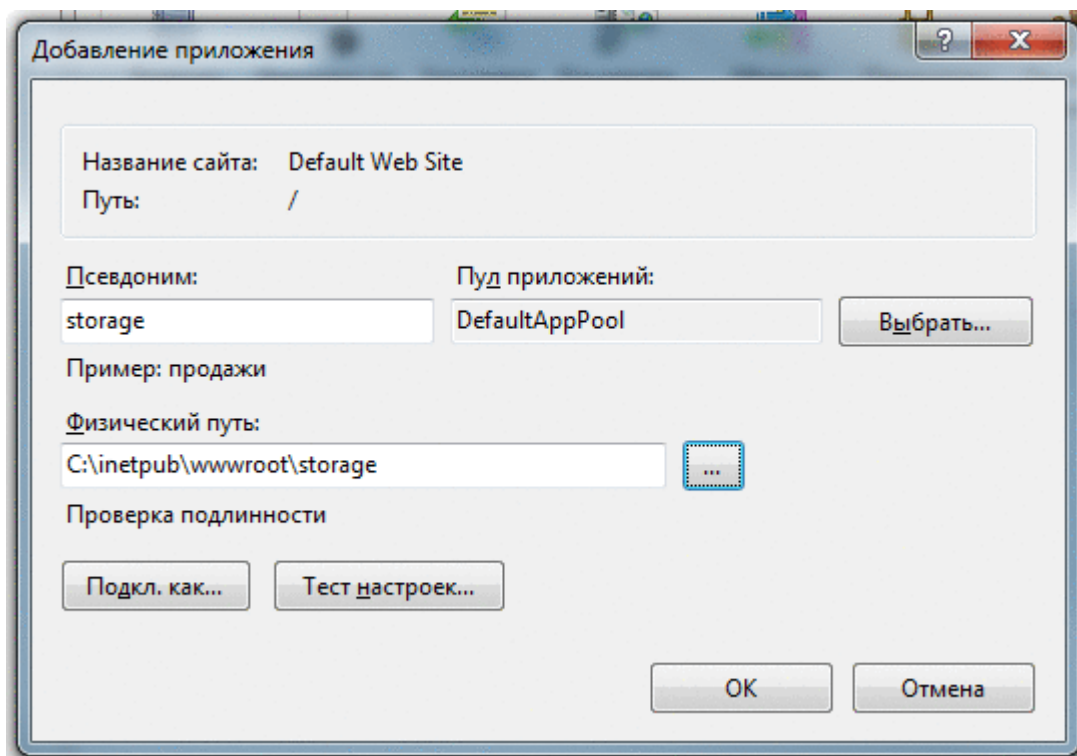


Рис. 504. Создание виртуального приложения

Имя, указанное в поле **Псевдоним**, будет использоваться при формировании URL доступа с хранилищу.

Затем необходимо добавить обработчик для файлов [.1ccr](#). Для этого необходимо выбрать команду <Имя компьютера с веб-сервером> – сайты – [Default Web Site](#) – [storage](#) – [Сопоставления обработчиков](#) – [Добавить сопоставление сценария ...](#)

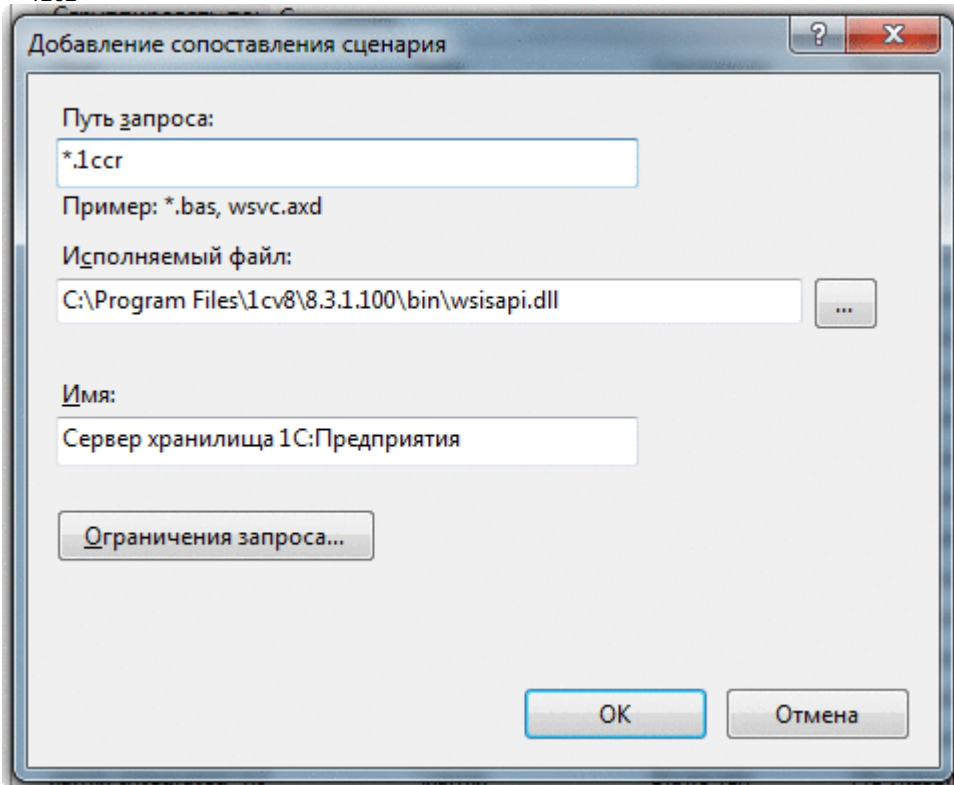


Рис. 505. Добавление обработчика

В качестве расширения веб-сервера IIS выступает динамическая библиотека [wsisapi.dll](#), расположенная в каталоге [bin](#) конкретной версии «1С:Предприятия».

В качестве расширения (поле [Путь запроса:](#) в диалоге на [рис. 505](#)) можно указать как конкретное расширение ([*.*scr](#)), так и маску всех файлов ([*](#)).

Последним шагом необходимо перезапустить веб-сервер.

30.4.3.2. Для веб-сервера Apache

Для того чтобы выполнить настройку веб-сервера Apache на работу с хранилищем конфигурации, необходимо выполнить следующие действия:

- Включить в конфигурационный файл веб-сервера ([httpd.conf](#)) загрузку модуля расширения веб-сервера. В зависимости от версии веб-сервера и используемой операционной системы, модули расширения имеют разное имя:

- Для ОС Windows:
 - Для Apache 2.0: [wsapch2.dll](#).
 - Для Apache 2.2: [wsap22.dll](#).
- Для ОС Linux:
 - Для Apache 2.0: [wsapch2.so](#).
 - Для Apache 2.2: [wsap22.so](#).

Модуль расширения веб-сервера имеет идентификатор `_lcws_module`. Для загрузки модуля необходимо внести в конфигурационный файл следующую строку вида `LoadModule _lcws_module <путь к модулю расширения веб-сервера>`.

Пример для ОС Windows:

```
LoadModule _lcs_module "C:\Program Files\lcv8\8.3.1.100\bin\wsap22.dll"
```

Пример для ОС Linux:

[Копировать в буфер обмена](#)

```
LoadModule _lcs_module /opt/1C/v8.2/i386/wsap22.so
```

- Создать на веб-сервере виртуальный каталог, через который будет осуществляться доступ к хранилищу конфигурации и зарегистрировать обработчик файлов для каталога виртуального приложения. Идентификатор регистрируемого обработчика: `1c-application`.

Пример виртуального каталога для ОС Windows:

[Копировать в буфер обмена](#)

```
Alias "/repos" "C:\www\repos\"
<Directory "C:\www\repos\">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
    SetHandler 1c-application
</Directory>
```

Пример виртуального каталога для ОС Linux:

[Копировать в буфер обмена](#)

```
Alias "/repos" "/var/1C/www/repos"
<Directory "/var/1C/www/repos">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
    SetHandler 1c-application
</Directory>
```

- Разместить в папке с файлами виртуального каталога файл `*.1csr`, в котором указать адрес сервера хранилища конфигурации (в схеме TCP).

Пример файла .1csr:

[Копировать в буфер обмена](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<repository connectString="tcp://192.168.0.12"/>
```

- Перезапустить веб-сервер.

30.4.4. Запуск сервера хранилища

В операционной системе Windows сервер хранилища конфигурации может быть запущен как приложение или установлен как сервис.

В операционной системе Linux сервер хранилища конфигурации может быть запущен как процесс или как демон.

30.4.4.1. Для ОС Windows

Запуск сервера хранилища выполняется из командной строки:

[Копировать в буфер обмена](#)

```
crserver.exe -instsrv | -rmsrv -usr <пользователь> -pwd <пароль>
                -start | -stop
                -port <порт> -d <каталог>
```

В случае установки в качестве сервиса сервер хранилища имеет имя **1C:Enterprise 8 Configuration Repository Server**.

ВНИМАНИЕ! Название и значение параметра должны разделяться символом пробел.

-start

Запуск сервиса сервера хранилища.

-stop

Остановка сервиса сервера хранилища.

-instsrvc

Регистрация сервера хранилища как сервиса.

-rmsrvc

Удаление регистрации сервера хранилища как сервиса.

-usr

Имя пользователя, от имени которого будет зарегистрирован сервис. Этот пользователь должен обладать правом [Вход в систему как сервис \(Log on as a service\)](#). Кроме этого, он должен обладать правами на чтение каталога исполняемых файлов соответствующей версии системы «1С:Предприятие» и полными правами на корневой каталог хранилища конфигурации (каталог `%APPDATA%\1C\1Cv8\` или тот каталог, который указан в параметре `-d`).

-pwd

Пароль пользователя, от имени которого будет зарегистрирован сервис.

-port

Рабочий порт сервера хранилища. По умолчанию используется порт [1542](#).

-d

Корневой каталог для хранилищ конфигурации. По умолчанию используется каталог `%APPDATA%\1C\1Cv8\`.

30.4.4.2. Для ОС Linux

Запуск сервера хранилища выполняется из командной строки:

```
./crserver -daemon -port <порт> -d <каталог>
```

[Копировать в буфер обмена](#)

ВНИМАНИЕ! Название и значение параметра должны разделяться символом пробел.

-daemon

Запуск сервера как демона.

-port

Рабочий порт сервера хранилища. По умолчанию используется порт [1542](#).

-d

Корневой каталог для хранилищ конфигурации. По умолчанию используется каталог `/home/usr1cv8/.1cv8` или `~/.1cv8`.

30.4.5. Подключение к серверу хранилища конфигурации

Подключение клиента к хранилищу конфигурации при удаленной работе осуществляется так же, как и при работе с хранилищем, расположенным на общем сетевом ресурсе. Отличие заключается лишь в способе указания каталога хранилища. В зависимости от используемого протокола каталог хранилища может быть указан двумя способами.

Для протокола *TCP*:

```
tcp://<имя сервера хранилища>/<относительный путь к хранилищу>
```

[Копировать в буфер обмена](#)

Для протокола *HTTP*:

```
<адрес файла описания (см. стр. )>/<относительный путь к хранилищу>
```

[Копировать в буфер обмена](#)

Например:

```
tcp://RepServ/MyConfRep
```

[Копировать в буфер обмена](#)

Или:

```
http://www.MyCompany.ru/VirtualFolder/storage.1ccr/MyConfRep
```

[Копировать в буфер обмена](#)

- **RepServ** – имя сервера хранилища конфигурации;
- <http://www.MyCompany.ru> – адрес веб-сервера, на котором опубликован файл описания Web-сервиса;
- <http://www.MyCompany.ru/VirtualFolder/storage.1ccr> – полный адрес файла описания см. в схеме HTTP;
- **MyConfRep** – относительный путь к хранилищу конфигурации.

Относительный путь указывается от каталога по умолчанию (**%APPDATA%\1C\1Cv8**), если каталог не был задан в явном виде, или от каталога, указанного в параметре **-d** при запуске сервера хранилища. Если относительный путь к хранилищу не указан, будет выполнено подключение к хранилищу по умолчанию (каталог **maincr**).

При работе с сервером хранилища конфигурации, версии сервера хранилища и Конфигуратора должны совпадать. В случае несовпадения версий Конфигуратора и сервера хранилища конфигураций работа с хранилищем невозможна.

30.4.6. Создание хранилища конфигурации

При первом запуске сервера хранилища конфигурации на компьютере будет создан каталог по умолчанию для размещения хранилища конфигурации. Такой каталог имеет имя **maincr**. Если при запуске сервера не был указан ключ **-d**, то хранилище конфигурации по умолчанию будет располагаться в каталоге **%APPDATA%\1C\1Cv8\maincr**.

Для создания нового хранилища конфигурации следует из клиентского приложения в режиме Конфигуратор выполнить создание хранилища, указав в качестве каталога хранилища требуемый новый каталог.

Чтобы обеспечить максимальную производительность, рекомендуется размещать корневой каталог хранилища конфигурации на компьютере сервера хранилища конфигурации. Система «1С:Предприятие» поддерживает размещение каталогов хранилища на сетевых ресурсах, доступных серверу хранилища, однако такой вариант работы может приводить к снижению производительности.

30.4.7. Особенности удаленной работы с хранилищем конфигурации

При удаленной работе с хранилищем конфигурации не выполняется автоматическое обновление состояния объектов в хранилище. Для обновления информации об объектах, размещенных в хранилище, следует выполнить команду [Действия – Обновить статусы](#).

Активные пользователи, работающие с хранилищем конфигурации удаленно, не выделяются жирным шрифтом в списке пользователей хранилища.

30.5. Рекомендации по использованию хранилища

Для обеспечения группы разработчиков информацией о внесенных изменениях каждый разработчик одновременно с помещением в хранилище измененных объектов конфигурации составляет описание внесенных изменений.

Чтобы описать изменения, нужно сформировать текстовый отчет о сравнении рабочей конфигурации с конфигурацией хранилища (только по помещаемым в хранилище объектам конфигурации) и поместить его в доступное место.

Администратор хранилища периодически (период определяется интенсивностью работы) должен производить архивирование хранилища и отчетов об изменениях, внесенных разработчиками.

Не рекомендуется выполнять групповые операции с хранилищем (получение объектов, помещение объектов и т. д.), а также выполнять снятие конфигурации с поддержки при открытом окне сравнения конфигураций.

Глава 31. Поставка и поддержка конфигурации

31.1. Общая информация

Типовые конфигурации регулярно меняются. Это может быть связано с изменением законодательства, добавлением новых функциональных возможностей или внесением исправлений. Поэтому большое значение для пользователей типовых конфигураций представляет механизм поддержки этих конфигураций.

Для описания механизмов поддержки используются термины «поставка» и «поддержка конфигурации», а также «комплект поставки».

Поставка. Различают полную поставку и поставку обновлений. Полная поставка представляет файл конфигурации формата *.cf, сформированный специальным образом. Поставка обновлений представляет файл обновлений формата *.cfu.

Поддержка. Поддержка конфигурации заключается в способности конфигурации быть обновляемой средствами конфигуратора с использованием файлов поставки. Назначение механизма поддержки – в защите логической целостности конфигурации. В описании также применяется термин «конфигурация, находящаяся на поддержке».

Комплект поставки. Комплект поставки представляет собой дистрибутив, в состав которого входит программа установки [setup.exe](#) и файлы поставки, сжатые в файл-архив [1cv8.efd](#). Язык получаемого дистрибутива соответствует языку конфигуратора (локализованные ресурсы дистрибутива находятся в каталоге дистрибутива, в подкаталоге с именем, соответствующим имени языка).

Конфигуратор позволяет разработчикам типовых конфигураций осуществлять поставки не только новых версий конфигураций, но и установить режим поддержки этих конфигураций с учетом произведенных изменений в пользовательских конфигурациях.

31.2. Поставка конфигурации

31.2.1. Настройка поставки

Настройка поставки заключается в указании правил поставщика на изменение объектов конфигурации разработчиками, осуществляющими поддержку конфигураций конечных пользователей.

Для настройки прав нужно выбрать пункт [Конфигурация – Поставка конфигураций – Настройка поставки](#). Открывается диалог настройки поставки.

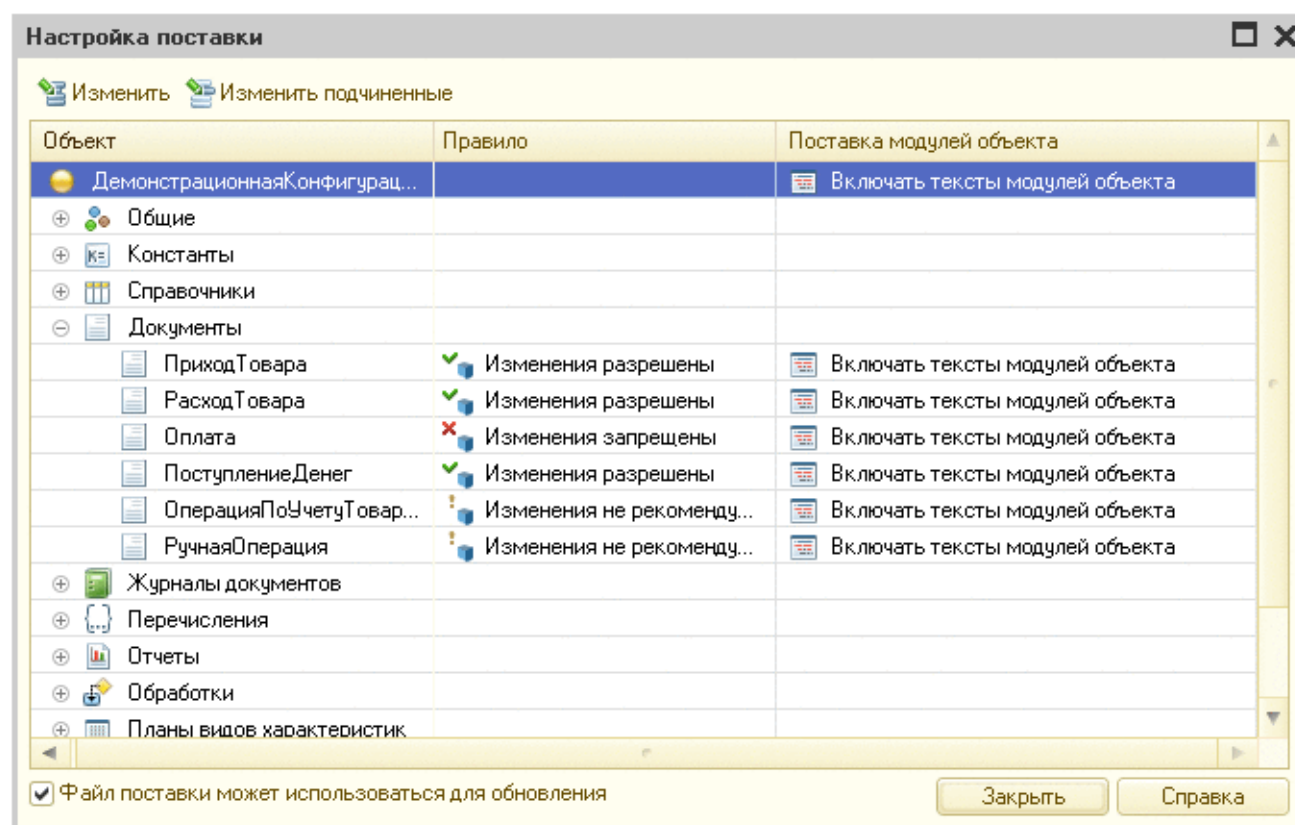
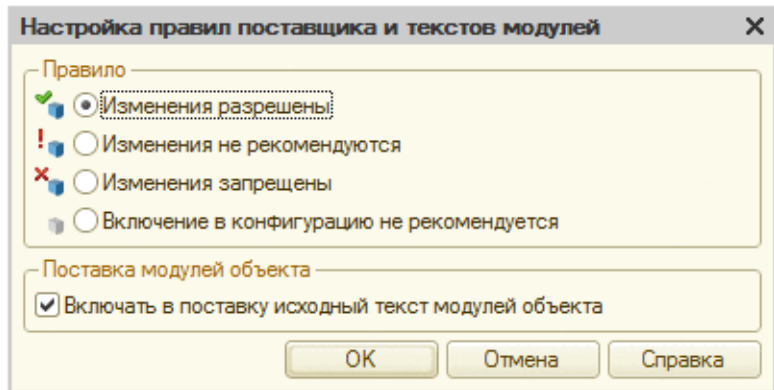


Рис. 506. Настройка поставки

В иерархическом списке объектов конфигурации доступны только объекты первого уровня.

По каждому объекту следует указать правило изменения. Для этого нужно указать объект и нажать кнопку [Изменить](#), а для изменения правил для группы объектов – выбрать группу и нажать кнопку [Изменить подчиненные](#).

**Рис. 507. Настройка правил поставщика**

В дереве объектов допускается множественный выбор. В этом случае настройка выполняется для выбранных объектов.

Допускается указание следующих правил:

- [Изменения разрешены](#) – допускается любое изменение в пользовательской конфигурации.
- [Изменения не рекомендуются](#) – изменения в пользовательской конфигурации не рекомендуются.
- [Изменения запрещены](#) – изменения полностью запрещены. Данное право не позволяет изменять объекты, находящиеся на поддержке. Объект с таким правилом нельзя снять с поддержки. Для изменения такого объекта нужно снимать с поддержки всю конфигурацию.
- [Включение в конфигурацию не рекомендуется](#) – при постановке на поддержку путем объединения с конфигурацией поставщика или при обновлении новой версией конфигурации поставщика объект с таким правилом по умолчанию не предлагается к включению в пользовательскую конфигурацию.

По умолчанию конфигуратор устанавливает правило [Изменения разрешены](#).

Если в поставку включается только скомпилированный текст модуля, нужно снять флажок [Включать в поставку исходный текст модулей объекта](#). В этом случае функциональность не нарушается, но текст модуля недоступен для просмотра. Не поддерживается исключение исходных текстов для следующих модулей:

- модуль управляемого приложения;
- модули форм;
- модули команд;
- модули, включающие директивы препроцессора;
- клиентские общие модули, работающие в управляемом режиме (тонкий клиент, веб-клиент и управляемый режим толстого клиента).

ПРИМЕЧАНИЕ. Если флажок [Включать в поставку исходный текст модулей объекта](#) снят для корневого элемента конфигурации, то без исходных текстов будут поставляться модули обычного приложения, внешнего соединения и сеанса.

Для настройки файла поставки флажок [Файл поставки может использоваться для обновления](#) следует установить, если файл поставки предназначается для обычного обновления конфигурации. Снять флажок нужно в том случае, когда предполагается использовать полученный файл поставки как промежуточный, когда требуется произвести ряд последовательных обновлений.

Например, смена типа реквизита при обычной технологии может привести к потерям введенных данных пользователей. Поэтому часто используют промежуточную версию, в которой вводят новый реквизит с нужным типом, а прежний реквизит остается. Обычно подготавливается только файл обновления. Специальная обработка выполняет преобразование данных. Затем используют следующую версию конфигурации, в которой имя вспомогательного реквизита, имеющего нужный тип, заменяется прежним. Назначение второй версии – только в выполнении переходной функции, поэтому его нельзя использовать как отдельное обновление.

После указания прав нужно нажать кнопку [Заккрыть](#).

31.2.2. Создание файлов поставки

Для создания файлов поставки и обновлений следует выбрать пункт [Конфигурация – Поставка конфигурации – Создать файлы поставки и обновления конфигурации](#).

Предварительно в категории свойств [Разработка](#) должны быть указаны свойства [Поставщик](#) и [Версия](#).

[Поставщик](#) – наименование поставщика конфигурации.

[Версия](#) – версия конфигурации (строка).

СОВЕТ. Рекомендуемая структура номера версии: [<версия>.<подверсия>.<релиз>.<сборка>](#), где элементы структуры представляют собой десятичные числа. Такая структура упрощает идентификацию версии конфигурации при последовательной разработке и построении файлов (комплектов) поставки.

В списке шаблонов и обновлений конфигураций конфигурации сортируются по номерам версий с учетом структуры номера.

Если конфигурация или база данных модифицирована, то конфигуратор просит произвести сохранение конфигурации и обновить конфигурацию базы данных.

На экран выводится окно формирования поставки (см. [рис. 508](#)).

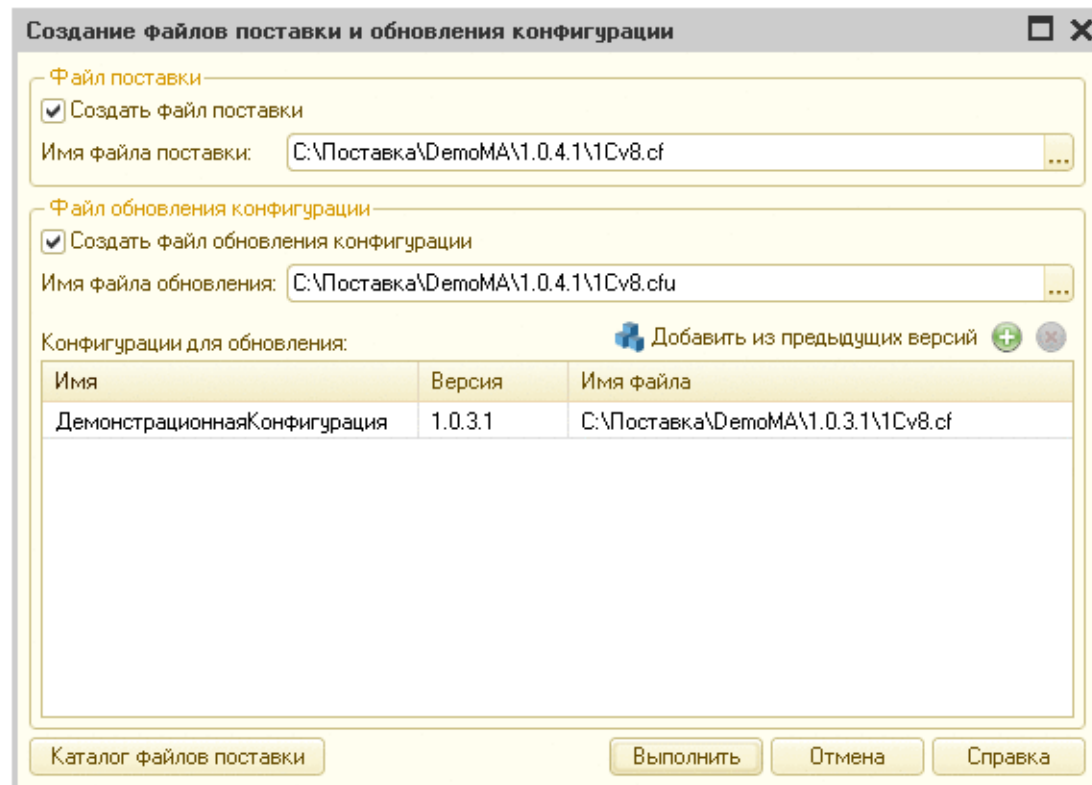


Рис. 508. Формирование поставки

По кнопке [Каталог файлов поставки](#) формируется каталог, который будет использован для создания каталогов поставки новых версий. Имя каталога версии совпадает с номером версии.

Для создания файла поставки нужно установить флажок [Создать файл поставки](#). В поле задания имени по умолчанию будет предложено имя файла поставки.

Для создания файла обновления установить флажок [Создать файл обновления конфигурации](#). В поле задания имени по умолчанию будет предложено имя файла обновления.

Если флажок [Создать файл обновления конфигурации](#) установлен, то требуется включить в файл обновления прежние обновления. Для этого нужно выполнить команду [Добавить из предыдущих версий](#) и выбрать файлы поставки прежних версий.

Для формирования указанных файлов нужно нажать кнопку [Выполнить](#).

Комплект поставки, сформированный в ОС Windows, будет не полностью работоспособен в ОС Linux, т. к. файловая система ОС Windows не позволяет выставить атрибут [Выполнение \(X\)](#) у файла [setup](#), который предназначен для работы в ОС Linux. Поэтому для того, чтобы иметь возможность запуска приложения [setup](#), нужно установить этот атрибут для файла [setup](#) после копирования установочного комплекта на компьютер ОС Linux или при создании образа установочного диска. Установку атрибута исполнения необходимо выполнять из ОС Linux. Установочная программа для ОС Linux ([setup](#)) может исполняться и в среде 32-разрядной версии ОС, так и в среде 64-разрядной версии.

Текстовые файлы комплекта поставки нужно подготавливать в кодировке UTF-8.

Полученные файлы передаются разработчикам, осуществляющим поддержку конфигураций конечных пользователей.

31.2.3. Подготовка комплекта поставки тиражных конфигураций

Для создания комплекта поставки сначала необходимо сформировать описание комплекта (хранится в файле), а затем в соответствии с этим описанием создать комплект поставки.

31.2.3.1. Описание комплекта поставки

Для формирования описания комплекта поставки нужно выбрать пункт [Конфигурация – Поставка конфигурации – Комплект поставки](#).

На экран выводится диалог (см. [рис. 509](#)).

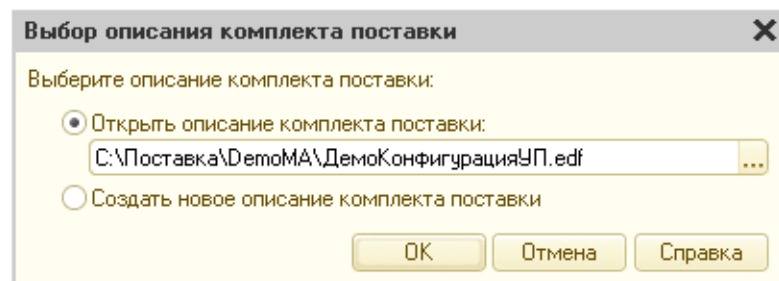


Рис. 509. Выбор описания комплекта поставки

В диалоге выбрать режим работы с описанием.

Затем установить переключатель [Открыть описание комплекта поставки](#), если требуется внести изменения в имеющееся описание. Файл описания указывается в поле ввода.

Для создания нового описания следует установить переключатель [Создать новое описание](#).

31.2.3.1.1. Создание нового описания комплекта поставки

На экран выводится окно помощника создания описания.

Рис. 510. Помощник создания комплекта поставки

Нужно указать наименование комплекта поставки и поставщика. Нажать кнопку [Далее >](#).

На следующем шаге указываются параметры шаблона.

Рис. 511. Создание описания комплекта поставки

По умолчанию (установлен флажок [Определять параметры по текущей конфигурации](#)) параметры выбираются из соответствующих свойств конфигурации. Если флажок не установлен, эти параметры можно изменить.

В параметре [Путь](#) указывается каталог, в котором будет создан файл-манифест.

Параметры шаблона записываются в файл манифеста, который входит в комплект поставки. О структуре файла-манифеста написано в книге «1С:Предприятие 8.3. Руководство администратора».

На следующем шаге указываются параметры шаблона.

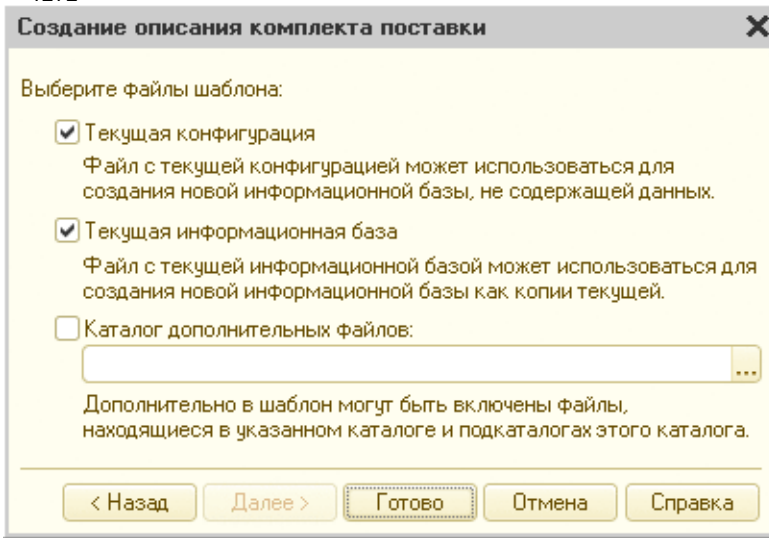


Рис. 512. Задание параметров шаблонов

Следует отметить файлы шаблона, которые должны войти в описание комплекта поставки. Для дополнительных файлов, включаемых в комплект, указать каталог их расположения. Все файлы этого каталога, в том числе и подкаталоги, будут включены в описание комплекта поставки.

Для создания описания нажать кнопку **Готово**. На экран выводится окно редактирования описания комплекта поставки.

31.2.3.1.2. Редактирование описания комплекта поставки

Окно редактирования описания комплекта поставки открывается после создания описания (см. раздел выше) или при выборе режима редактирования (см. [здесь](#)).

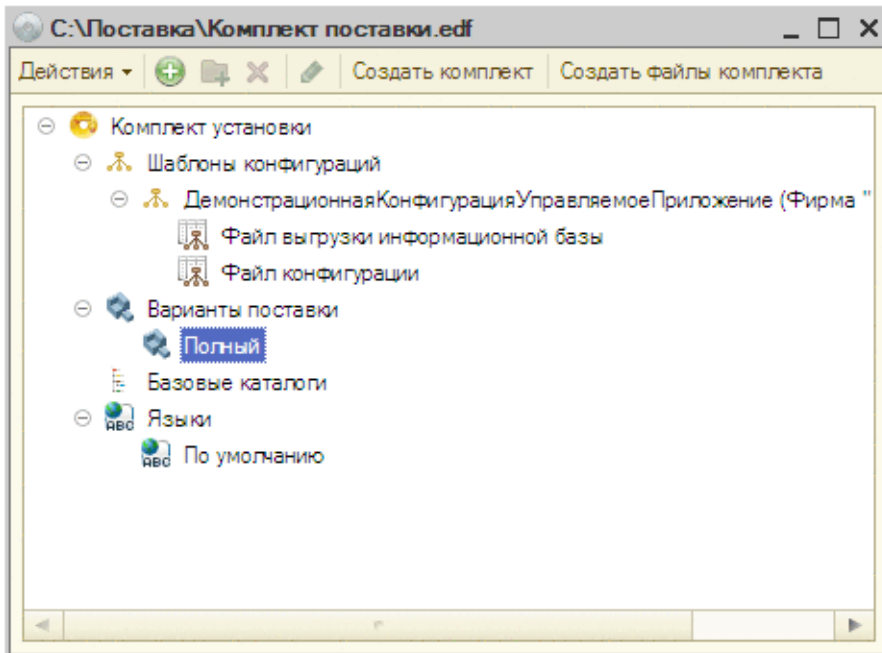


Рис. 513. Редактор описания комплекта поставки

В окне редактирования описания поставки выполняется настройка для создания комплекта поставки.

Описание представляет собой дерево, состоящее из четырех основных ветвей:

- шаблоны конфигураций,
- варианты поставки,
- базовые каталоги,
- языки.

По любой ветке возможно добавление новых данных, удаление ненужных и изменение существующих свойств. Все действия по составу выполняются с помощью команд меню **Действие**. Редактирование свойств производится в палитре свойств.

Шаблоны конфигураций

Шаблоны конфигураций представляют собой основные элементы комплекта поставки. В состав шаблонов конфигураций входят файлы и группы файлов.

Файлы могут быть как произвольными (или наборами файлов, определяемыми по маске), так и файлами конфигурации или выгрузки данных, получаемыми из текущей информационной базы.

Каждый шаблон имеет свое размещение в каталоге шаблонов. Соответственно, файлы и группы файлов размещаются в каталоге шаблонов, исходя из адреса файла-манифеста (группы файлов представляют при этом подкаталоги).

В файл-манифест включаются только файлы конфигурации и выгрузки данных. Остальные файлы просто размещаются в каталоге шаблона конфигурации.

Шаблоны

Для настройки свойств шаблона нужно выбрать шаблон и указать его свойства.

Текущая конфигурация – если флажок установлен, то данные (поставщик, имя, версия) выбираются из текущей информационной базы. Если флажок снят, то допускается редактирование этих свойств:

- **Поставщик** – поставщик конфигурации (соответствует свойству конфигурации **Поставщик**);
- **Имя** – имя конфигурации (соответствует свойству конфигурации **Имя**);
- **Версия** – версия конфигурации (соответствует свойству конфигурации **Версия**);
- **Размещение манифеста** – относительный путь к файлу-манифесту шаблона в каталоге шаблонов.

Файлы и группы файлов

Для добавления файла нужно выбрать шаблон и пункт **Действия – Добавить**. На экран выводится диалог выбора файла.

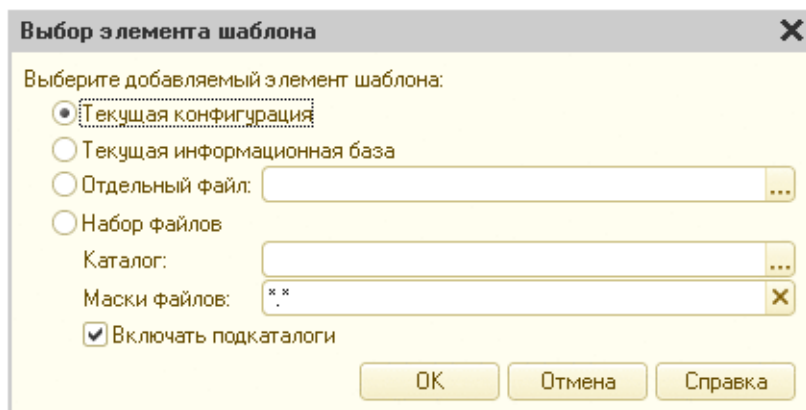


Рис. 514. Добавление файла в шаблон

В диалоге указывается местонахождение добавляемого элемента шаблона.

Файл может располагаться в текущей конфигурации, текущей информационной базе, в отдельном файле, а также представлять набор файлов указанного каталога, отбираемый по маске файлов (допускается включение файлов подкаталогов).

Для настройки свойств файла нужно выбрать файл и указать его свойства:

- **Наименование** – наименование файла.

- **Данные** – источник данных для файла (файл, текущая конфигурация или текущая информационная база).
- **Базовый каталог** – базовый каталог для ссылки на файл.
- **Файл/каталог** – относительный (относительно базового каталога) или абсолютный путь к файлу.
- **Поставить на поддержку** – для файла конфигурации или выгрузки данных – поставить файл на поддержку.
- **Включать в манифест** – для файла конфигурации или выгрузки данных включать файл в манифест.
- **Наименование в шаблоне** – наименование в каталоге шаблонов. Наименование является локализуемым в соответствии с языками комплекта установки.
- **Размещение** – каталог для информационной базы, предлагаемый по умолчанию.
- **Наименование для обновлений** – использовать наименование файла в каталоге шаблонов для наименования обновлений.

Для добавления группы файлов нужно выбрать шаблон и пункт **Действия – Новая группа**. В палитре свойств ввести наименование группы.

Если в списке указать группу и выбрать пункт **Действия – Добавить**, то в данную группу добавляется файл. Число файлов в группе неограниченно.

Допускается создание вложенных групп файлов.

Варианты поставки

Варианты поставки представляют собой различные комбинации файлов поставки. Такие комбинации могут использоваться для различной комплектации поставки (например, при полной поставке или же при поставке только обновлений):

- **Наименование** – наименование варианта поставки.
- **Поставляемые файлы** – набор поставляемых файлов. По умолчанию создается полный вариант, в который включены все файлы. Если указано несколько вариантов построения, то состав файлов можно сформировать, нажав для него в палитре свойств ссылку **Редактировать**. Нужно установить отметки для тех файлов, которые будут включены в выбранный вариант построения.
- **Файл описания поставки** – описание, показываемое пользователю после проведения установки продукта.
- **Базовый каталог** и **Каталог** – используются для указания на место создания комплекта поставки или набора файлов поставки. Если эти каталоги указаны, то при запросе места создания они будут автоматически подставлены в диалог запроса.

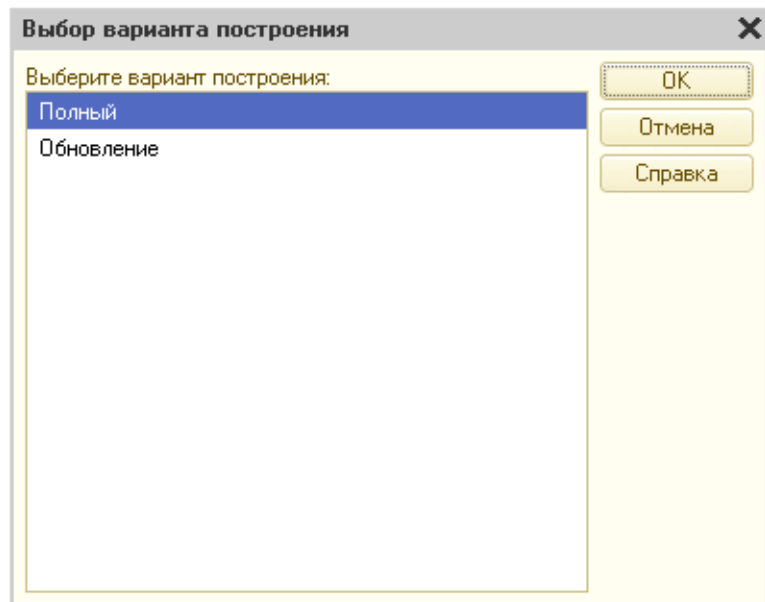


Рис. 515. Вариант построения

Базовые каталоги

Базовые каталоги используются для переносимого между различными компьютерами указания файлов поставки.

В качестве базового каталога может быть указано значение переменной среды или имя каталога. После этого имя файла поставки может быть указано как комбинация базового каталога и относительного пути от этого каталога:

- **Наименование** – наименование базового каталога.
- **База** – может принимать два значения:
 - **Переменная среды**,
 - **Каталог**.
- **Переменная среды/Каталог** – содержит имя переменной среды (переменной окружения) или абсолютный адрес каталога в зависимости от значения свойства **База**.

Языки

Языки предназначены для представления локализуемых наименований элементов шаблонов.

Для корректного отображения наименований следует создавать языки в соответствии с наименованиями каталогов платформы системы «1С:Предприятие», содержащими соответствующие локализованные интерфейсы. Один из языков является языком по умолчанию (локализованная по умолчанию строка наименования используется при отсутствии строки, соответствующей текущему языку платформы):

- **Наименование** – наименование языка;
- **Каталог ресурсов** – имя каталога языковых ресурсов, соответствующее этому языку.

31.2.3.2. Создание комплекта

Для создания комплекта поставки нужно выбрать пункт **Действия – Создать комплект**. При этом будет создан комплект поставки, соответствующий указанному варианту поставки.

Чтобы создать комплект файлов поставки, нужно выбрать пункт **Действия – Создать комплект файлов поставки**. При этом будут созданы файлы поставки (не сжатые в пакет установки) без программы установки, соответствующие указанному варианту поставки.

Если при этом описание комплекта поставки не было сохранено, конфигуратор предлагает его сохранить.

Если количество вариантов больше одного, то на экран выводится диалог выбора варианта построения. В диалоге нужно выбрать требуемый вариант.

Создаваемые файлы комплекта обновления следует размещать в каталоге ресурсов, указанном в свойстве [Адрес каталога обновлений](#) (см. [здесь](#)).

Помимо лицензирования с использованием аппаратного ключа, ограничивающего общее количество рабочих мест и не контролирующего используемые конфигурации, допускается использование электронных лицензий, которые выдаются на конкретное рабочее место и контролируют использование на нем определенных конфигураций, записанных в лицензии. Программные лицензии выдаются Центром лицензирования фирмы «1С» и сохраняются на компьютере пользователя.

ПРИМЕЧАНИЕ. Для создания комплекта поставки конфигурации, распространяемой по схеме лицензирования рабочего места пользователя, реализован параметр **-DigiSign** ключей командной строки **/CreateDistributionFiles** и **/CreateDistributive**, позволяющий задать файл с параметрами лицензирования. Возможность создания и распространения конфигураций, поддерживающих схему лицензирования рабочего места пользователя, может быть получена партнером на основании отдельного договора с фирмой «1С».

31.3. Поддержка конфигурации

31.3.1. Общая информация

Новую версию конфигурации можно получить в виде поставки (файл поставки конфигурации новой версии) и/или в виде обновления.

Конфигурации пользователей, которые могут быть обновлены с учетом полученных файлов поставки, должны находиться на поддержке.

Режим поддержки объектов можно менять, предварительно отказавшись от полной поддержки. Возможность устанавливать правила пользователя (изменение объектов) зависит от правил, заданных поставщиком, и текущих блокировок пользовательской конфигурации.

Для конфигураций, находящихся на поддержке, предусмотрен режим сравнения с поставкой, сравнения новой поставки с прежней, а также сравнение текущего состояния конфигурации с первоначальным.

Если конфигурация находится на поддержке, то в окне [Конфигурация](#) объекты помечаются специальными пиктограммами, располагающимися справа от объекта, вдоль границы окна.

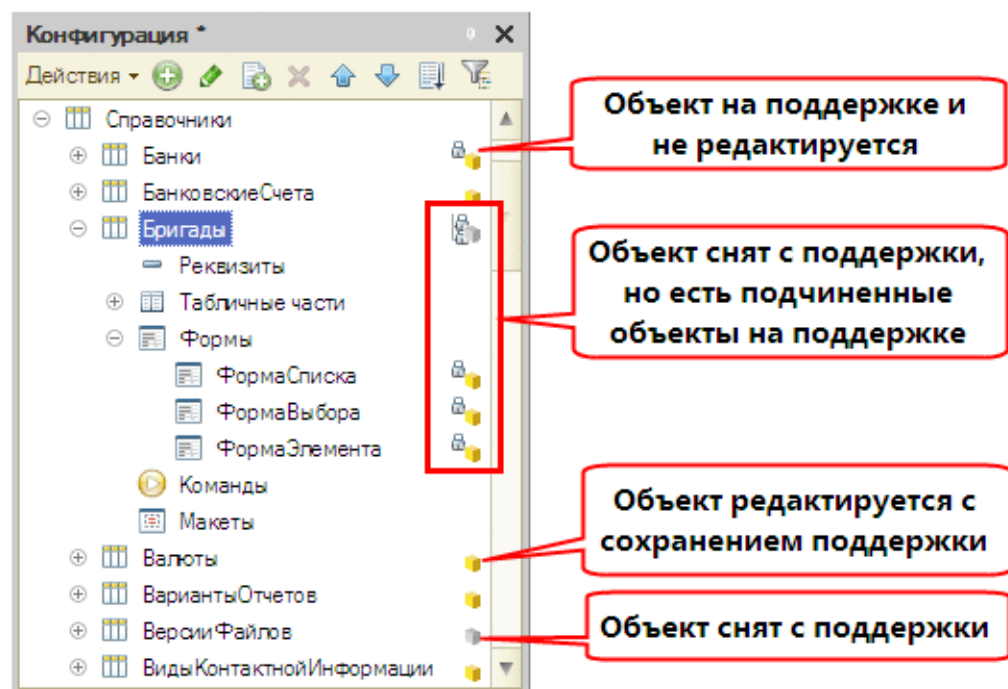


Рис. 516. Пиктограммы режимов поддержки

Описание пиктограмм, обозначающих различные режимы поддержки, приведено на [рис. 516](#).

31.3.2. Постановка конфигурации на поддержку

После установки типовой конфигурации она автоматически устанавливается на поддержку.

Если конфигурация не находится на поддержке, то для постановки ее на поддержку нужно выбрать пункт [Конфигурация – Сравнить, объединить с конфигурацией из файла](#), а в качестве сравниваемого файла выбрать файл поставки.

В этом случае конфигуратор предлагает произвести постановку конфигурации на поддержку. На экран выводится диалог, в котором помещен текст [Обнаружена возможность объединения с постановкой на поддержку](#) и сообщаются основные параметры конфигурации поставщика. Производится запрос на постановку.

При нажатии кнопки [Да](#) будет открыто окно сравнения и объединения конфигураций.

После указания условий и режимов объединения нужно нажать кнопку [Выполнить](#). Будет произведен пообъектный анализ конфигураций на соответствие правил поставщика правилам объединения.

Если обнаруживаются объекты, по которым правила поставщика вступают в противоречие с правилами объединения, то на экран выводится окно со списком этих объектов. В окне сравнения следует выполнить нужные настройки, чтобы снять противоречия, и повторить попытку объединения. Если противоречий в правилах нет, то производится объединение конфигураций с постановкой текущей конфигурации на поддержку.

При включении возможности изменений конфигурации поставщика, при постановке на поддержку и при обновлении конфигурации поставщика с включенной возможностью изменений имеется возможность детально настроить правила поддержки по умолчанию для объектов с разными правилами поставки.

Если текущая конфигурация подключена к хранилищу конфигураций, то при постановке конфигурации на поддержку требуется, чтобы все объекты конфигурации, которые должны быть полностью взяты из конфигурации поставщика, а также корневой объект конфигурации должны быть захвачены в хранилище.

31.3.3. Настройка поддержки

Настройка поддержки заключается в установке правил пользователя для каждого объекта конфигурации.

Для настройки поддержки нужно выбрать пункт [Конфигурация – Поддержка – Настройка поддержки](#). На экран выводится окно настройки правил поддержки.

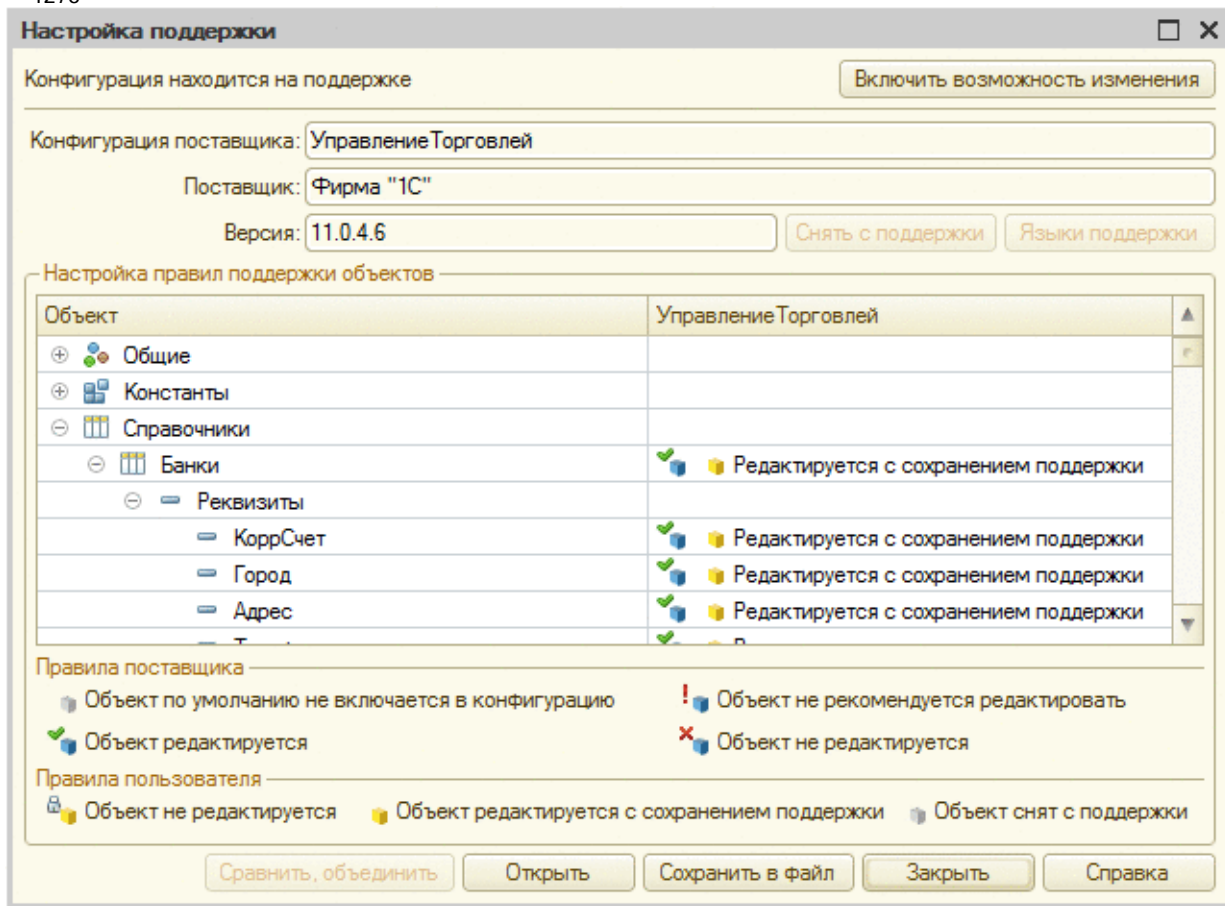


Рис. 517. Настройка поддержки

В верхней строке окна настройки конфигуратор сообщает, в каком состоянии поддержки находится данная конфигурация.

Ниже указывается имя конфигурации, поставщик и текущая версия конфигурации.

В разделе [Настройка правил поддержки объектов](#) размещено дерево объектов и показываются правила поддержки поставщика и правила пользователя по каждому объекту.

Правила поддержки поставщика не могут быть изменены. Можно менять только правила пользователя (при условии, что конфигурация снята с полной поддержки). В окне настройки допускается использование множественного выбора для настройки одинаковых правил поддержки выбранным объектам. В этом случае для изменения правил поддержки нужно использовать контекстное меню в колонке правил поддержки.

Комплект поставки может содержать данные на нескольких языках. С помощью выбора языков поддержки можно указать те из них, которые используются данной конфигурацией. Для этого следует нажать кнопку [Языки поддержки](#) и в открывшемся окне указать эти языки.

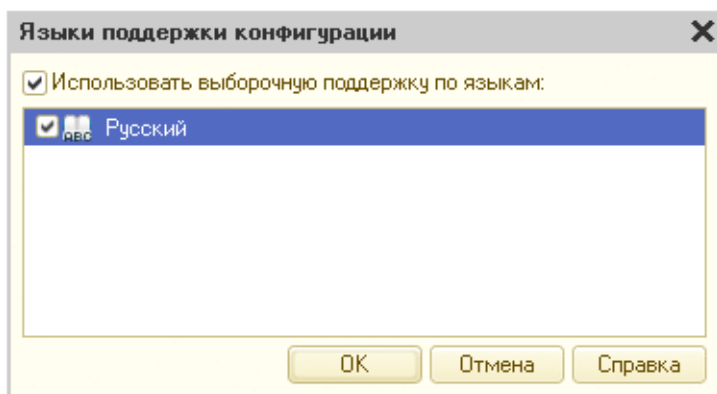


Рис. 518. Выбор языков поддержки конфигурации

31.3.3.1. Полная (автоматическая) поддержка

Если в строке состояния окна настройки поддержки указано [Конфигурация находится на поддержке](#), то это

означает, что все объекты данной конфигурации находятся на поддержке и не могут быть изменены (отредактированы). В этом случае конфигурацию можно обновить автоматически.

Для выполнения автоматического обновления текущей версии конфигурации:

- нужно выбрать пункт [Конфигурация – Поддержка – Обновить конфигурацию](#);
- выбрать файл поддержки.

Обновление производится без вывода окна [Сравнение и объединение](#).

В данном состоянии изменение правил пользователя для любого объекта конфигурации недоступно независимо от правил поддержки поставщика.

31.3.3.2. Поддержка с возможностью обновления

Часто требуется изменить типовую конфигурацию, чтобы учесть требования конкретного пользователя. В таких случаях необходимо отказаться полностью (объект поставщика снят с поддержки) или частично (объект поставщика редактируется с возможностью поддержки) от поддержки некоторых объектов.

Для получения доступа к установке правил поддержки пользователя в окне настройки поддержки нужно нажать кнопку [Включить возможность изменения](#).

После нажатия кнопки на экран выводится предупреждение: [Изменение режима приведет к невозможности выполнять обновление конфигурации полностью автоматически. Продолжить?](#) Если выбрать [Да](#), то будет предложено выбрать правила поддержки для объектов с разными правилами поставки.

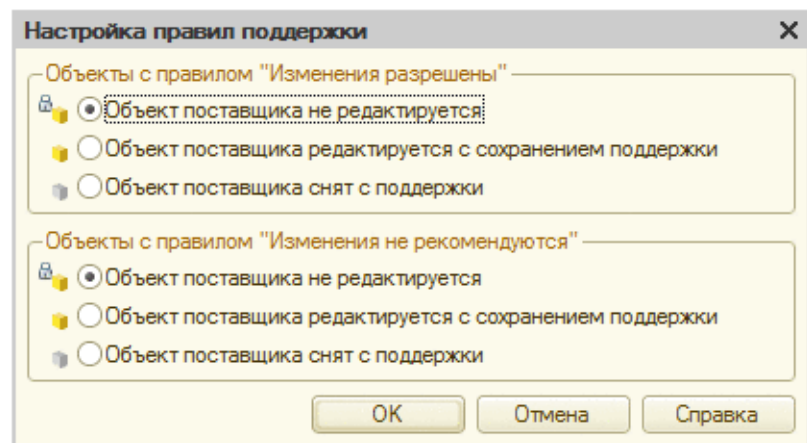


Рис. 519. Настройка правил поддержки

После указания правил поддержки и нажатия кнопки [ОК](#) произойдет изменение статуса конфигурации, и она будет находиться на поддержке с возможностью изменения. В результате такой настройки все объекты конфигурации окажутся заблокированными от случайных изменений, независимо от правил поставки, установленных поставщиком. Теперь для изменений объекта необходимо явно включать возможность изменения объекта или его подчиненных объектов (см. [здесь](#)).

Если не требуется менять режим поддержки (действия были выполнены по ошибке), нужно закрыть конфигурацию без сохранения.

После отказа от полной поддержки заново установить поддержку для всей конфигурации возможно только при последовательном выполнении следующих действий:

- снять конфигурацию с поддержки (в окне настройки нужно указать корневой объект конфигурации и выбрать пункт [Действия – Снять с поддержки](#));
- выбрать пункт [Конфигурация – Загрузить конфигурацию из файла](#);
- выбрать файл поставки.

31.3.3.2.1. Изменение правил поддержки объектов

Чтобы изменить правила поддержки, в дереве объектов окна настройки поддержки следует выбрать нужный

объект и в контекстном меню выбрать [Установить правило поддержки](#). В диалоге выбрать нужное правило пользователя из трех возможных:

- объект поставщика не редактируется;
- объект поставщика редактируется с сохранением поддержки;
- объект поставщика снят с поддержки.

Доступность правил определяется установленным правилом поддержки поставщика.

Если установлено правило поддержки поставщика [Изменения разрешены](#), то доступна установка любого правила поставщика.

Если установлено правило поддержки поставщика [Изменения не рекомендуются](#), то также доступна установка любого правила поставщика, но при изменении данного правила будет выдано предупреждающее сообщение.

Если установлено правило поддержки поставщика [Изменения запрещены](#), то установка правил поставщика невозможна.

ВНИМАНИЕ! После установки правила пользователя [Объект поставщика снят с поддержки](#) нельзя «вернуть» правило [Объект поставщика не редактируется](#).

Если выбранный объект содержит подчиненные объекты, то для изменения правил поддержки одновременно и для этих объектов нужно установить флажок [Установить для подчиненных объектов](#) (по умолчанию установлен).

Полный отказ от поддержки. Для отказа от поддержки всех объектов конфигурации следует выбрать пункт [Действия – Снять с поддержки](#). На экран выводится предупреждение: [Снятие с поддержки приведет к невозможности получать обновления от поставщика. Продолжить?](#) Для снятия с поддержки необходимо выбрать кнопку [Да](#). Конфигурация полностью снимается с поддержки. Если данные действия выполнены по ошибке, нужно закрыть конфигурацию без сохранения.

ВНИМАНИЕ! Отказ от поддержки не означает, что данную конфигурацию невозможно обновлять методом сравнения и объединения.

31.3.3.2.2. Сравнение и объединение с конфигурацией поставщика

Из окна настройки поддержки можно выполнить сравнение и объединение с первоначальным (предыдущим) файлом поставки. Для этого нужно нажать кнопку [Сравнить, объединить](#). На экран выводится окно сравнения и объединения.

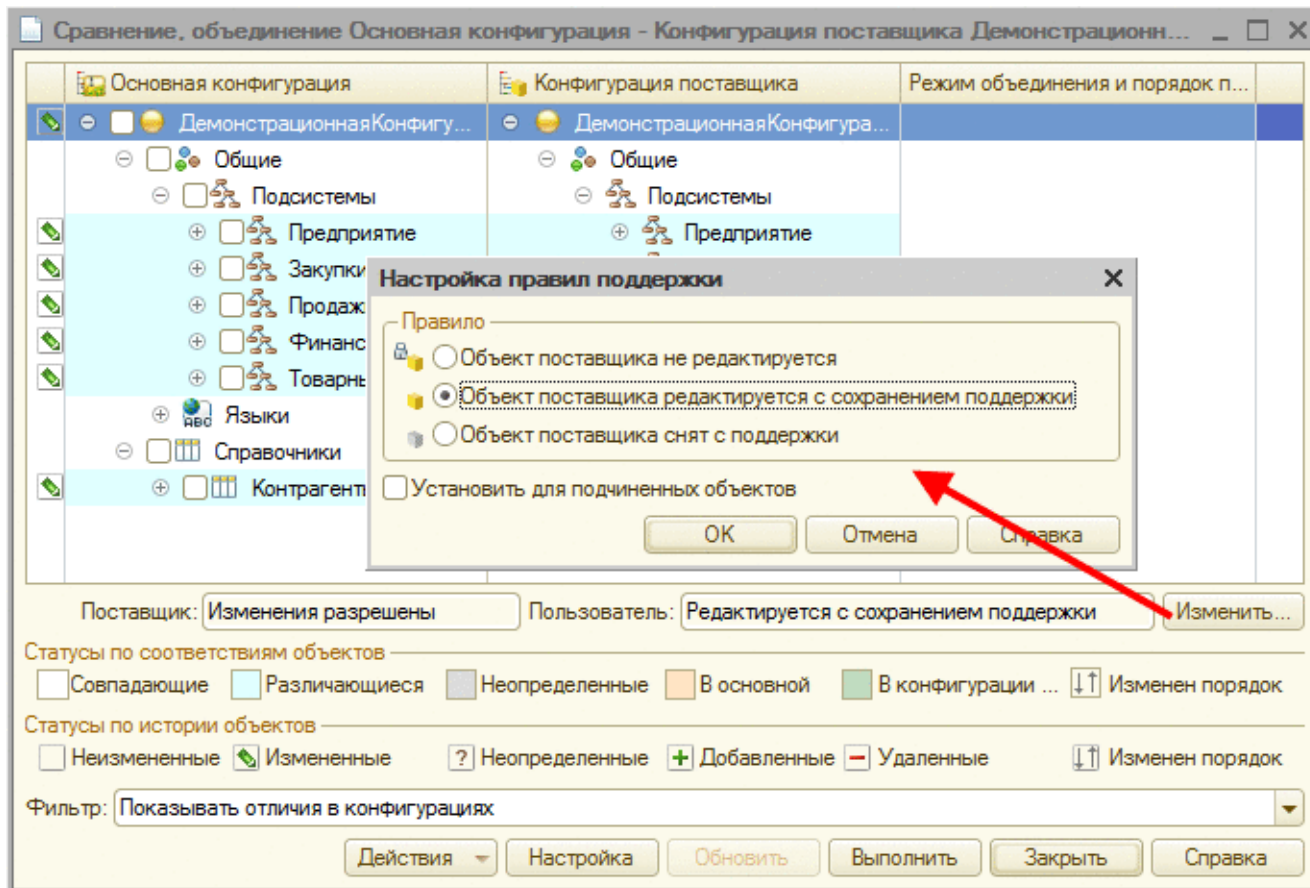


Рис. 520. Сравнение и объединение с конфигурацией поставщика

Основные приемы сравнения и объединения см. [здесь](#).

В отличие от «обычного» режима сравнения и объединения конфигураций, в окно сравнения добавлена строка показа правил поставщика и пользователя, а также кнопка **Изменить** для изменения правила пользователя. В этом случае открывается диалог с возможностью указания правила поддержки и возможностью иерархического указания этого правила (флажок **Установить для подчиненных объектов**).

При сравнении с файлом поставки текущая конфигурация всегда является потомком конфигурации-поставщика, поэтому кнопка выбора режима настройки конфигураций отсутствует.

31.3.4. Обновление конфигурации

Для выполнения обновления конфигурации, находящейся на поддержке, нужно выбрать пункт **Конфигурация – Поддержка – Обновить конфигурацию**.

На экран выводится помощник обновления.

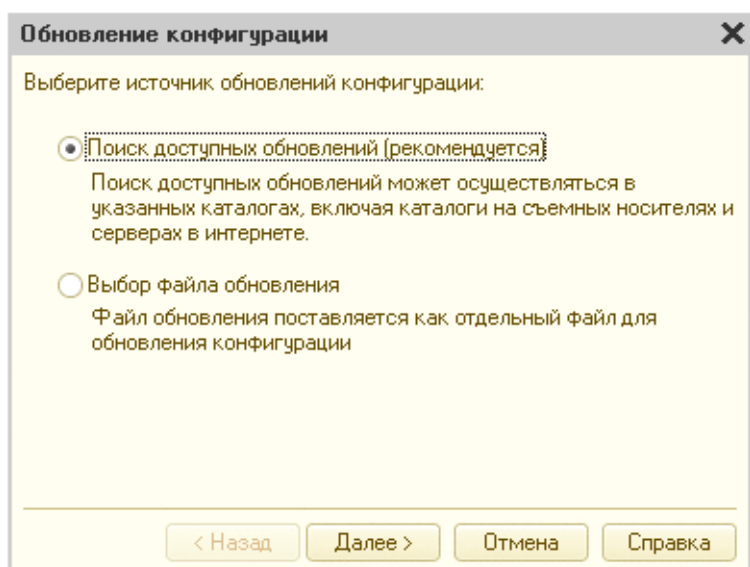
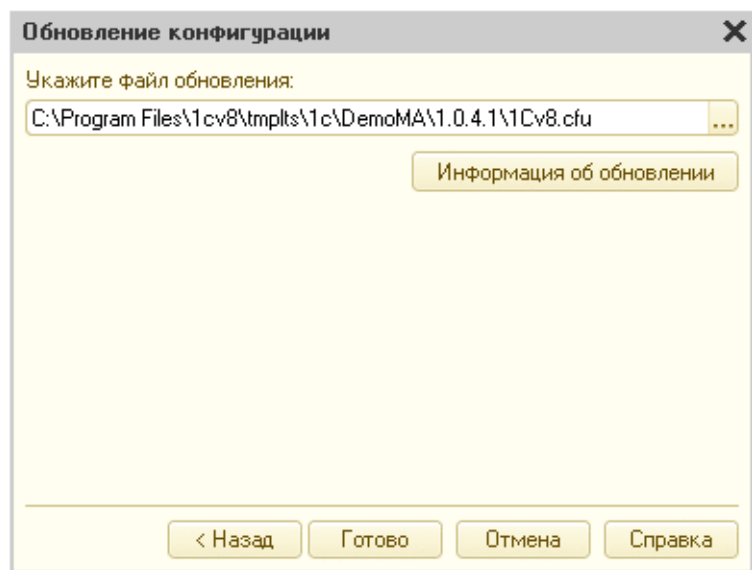


Рис. 521. Помощник обновления

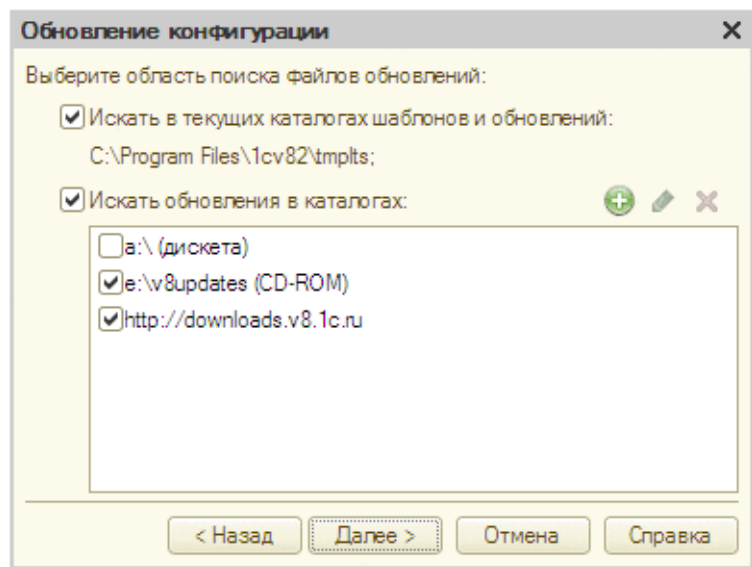
На первом шаге пользователь для обновления конфигурации может выбрать отдельный файл обновления или начать поиск по возможным местам расположения файлов обновлений.

Если выбран отдельный файл обновления, то на следующем шаге этот файл выбирается.

**Рис. 522. Выбор файла обновления**

При нажатии кнопки **Готово** выводится диалог, в котором указываются основные параметры текущей конфигурации и конфигурации поставщика (см. ниже).

Если выбран поиск отдельных обновлений, то после нажатия кнопки **Далее >** помощник обновления переходит к выбору возможных мест расположения обновлений.

**Рис. 523. Места расположения обновлений**

В качестве возможных мест расположения приводятся:

- текущий каталог шаблонов;
- съемные диски (при этом для дисков CR-ROM добавляется каталог **v8updates**);
- адрес, указанный в конфигурации как адрес каталога обновления.

По указанным каталогам производится поиск обновлений следующим образом:

- для локальных каталогов производится поиск файлов обновлений и файлов списка шаблонов в

указанных каталогах и в подкаталогах. Найденные файлы списка шаблонов должны описывать каталоги шаблонов и находиться в корне этих каталогов;

- для удаленных каталогов производится поиск файлов списка шаблонов только в указанных каталогах.

Если в свойстве конфигурации [Адрес каталога обновлений](#) указан адрес, с которого можно получить обновление конфигурации, то при нажатии кнопки [Далее >](#) выполняется аутентификация пользователя.

Рис. 524. Аутентификация доступа к интернет-ресурсу

Необходимо ввести имя и пароль пользователя для получения доступа к серверу, на котором размещен комплект обновления, а также имя и пароль для доступа через прокси-сервер.

На третьем шаге показываются обновления, для которых найдены соответствующие им шаблоны. В списке этих обновлений жирным шрифтом выделяется обновление, которое наиболее вероятно подходит для обновления конфигурации.

Рис. 525. Выбор обновления

Если выбирается обновление, которое находится в удаленном каталоге, то это обновление копируется в текущий каталог шаблонов, если он является локальным.

Нажатие кнопки [Готово](#) завершает поиск файла обновления.

На экран выводится диалог, в котором указываются основные параметры текущей конфигурации и конфигурации поставщика.

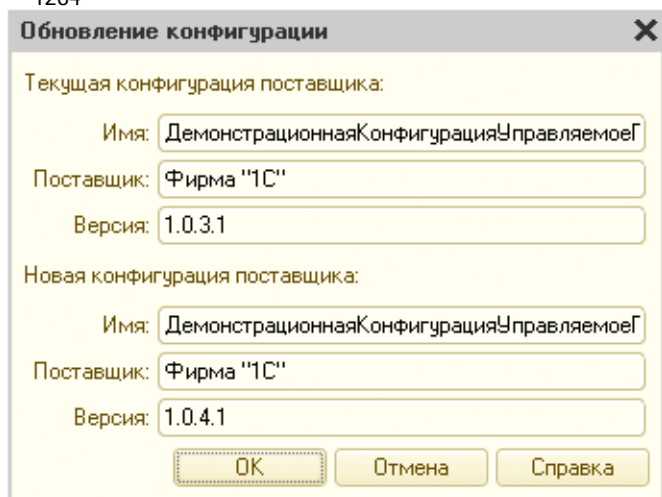


Рис. 526. Параметры обновления

Для продолжения нужно нажать кнопку [OK](#).

Конфигуратор производит сравнение конфигураций и выводит на экран окно [Обновление конфигурации](#).

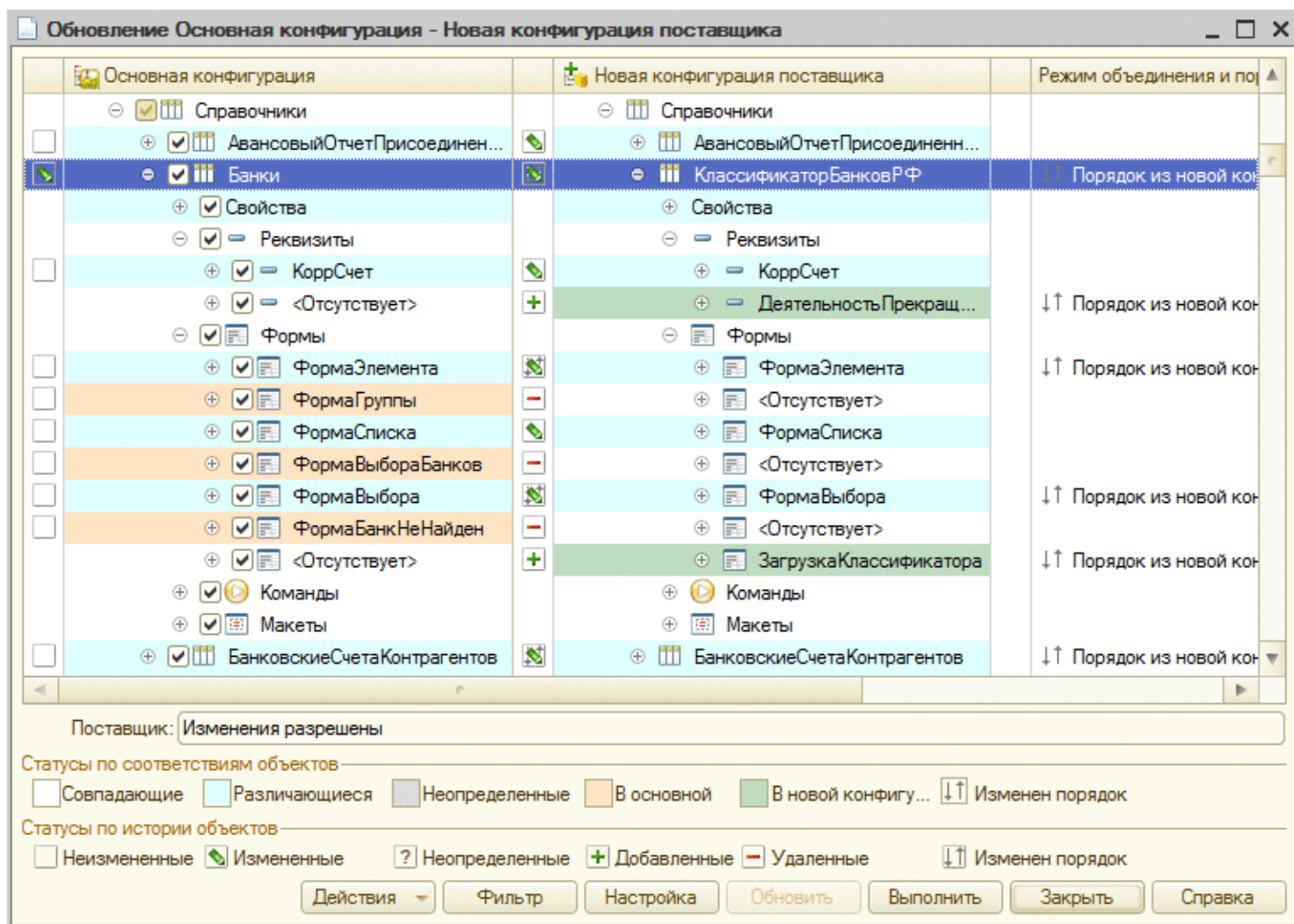


Рис. 527. Обновление конфигурации

Основные приемы сравнения и объединения см. [здесь](#).

Если в настройках конфигуратора выбрано использование внешней программы (см. [здесь](#)), то для программных модулей в список режимов объединения добавляется пункт [Объединить с помощью внешней программы](#). При выборе этого режима для объединения модулей используются возможности внешней программы. Если у выбранной внешней программы не заполнено свойство [Автоматическое трехстороннее объединение](#), то для выполнения объединения будет использоваться интерактивное трехстороннее объединение. Интерактивное трехстороннее объединение будет также использоваться в том случае, если

внешняя программа не смогла автоматически разрешить конфликты при выполнении объединения.

Некоторые действия, которые можно выполнить в окне обновления конфигурации, зависят от настроек конфигуризатора (см. [здесь](#)):

- Команда контекстного меню [Показать различия в модулях](#):
 - Настройка конфигуризатора: [Режимы "Взять из ...", "Объединить с приоритетом ..."](#);
 - Настройка внешней программы: [Объединение двух файлов](#);
 - Результаты работы: игнорируется.
- Команда контекстного меню [Показать различия в модулях без учета структуры](#):
 - Настройка конфигуризатора: [Показать различия без учета структуры](#);
 - Настройка внешней программы: [Сравнение двух файлов](#);
 - Результаты работы: игнорируется.
- Действия, выполняемые при указании режима объединения [Взять из ...](#) или [Объединить с приоритетом ...](#):
 - Настройка конфигуризатора: [Режимы "Взять из ...", "Объединить с приоритетом ..."](#);
 - Настройка внешней программы: [Трехстороннее объединение](#);
 - Результаты работы: используется в качестве результата объединения.
- Действия, выполняемые при указании режима объединения [Объединить с помощью внешней программы](#):
 - Настройка конфигуризатора: [Режим "Объединить с помощью внешней программы"](#);
 - Настройка внешней программы: [Трехстороннее объединение](#);
 - Результаты работы: используется в качестве результата объединения.
- Действия, выполняемые после нажатия кнопки [Выполнить](#):
 - Настройка конфигуризатора: группа [Обновление конфигурации на поддержке](#);
 - Настройка внешней программы: [Автоматическое трехстороннее объединение](#) или [Трехстороннее объединение](#);
 - Результаты работы: используется в качестве результата объединения.

При вызове команды будет использоваться тот инструмент, который указан в соответствующей настройке конфигуризатора, а если в качестве инструмента выбрана какая-либо внешняя программа – для выполнения действия будет использоваться командная строка, заданная в соответствующей настройке внешней программы.

При объединении конфигураций вначале выполняется объединение модулей. Для режима [Объединить с помощью внешней программы](#), запуск внешней программы, поддерживающей автоматическое трехстороннее объединение, выполняется только при обнаружении конфликтов. Если разработчик не сохранил результат объединения, то текст модуля не меняется, и объединение конфигураций продолжается. Если среди всех модулей был хотя бы один не разрешенный конфликт, то после неудачной попытки объединения модулей объединение конфигураций останавливается, и в окно сообщений выдается список модулей, которые не были объединены. Окно объединения конфигураций не закрывается, в дереве объектов обновляется колонка с состоянием настройки объединения. Разработчик должен вручную объединить необъединенные модули и продолжить объединение конфигураций нажатием кнопки [Выполнить](#).

В том случае, если в объединяемых конфигурациях используются объекты типа [ОпределяемыйТип](#), то для свойства этого объекта [Тип](#), а также для свойства [Тип](#) объекта [ПланВидовХарактеристик](#), используются

следующие правила объединения:

- Режим объединения [Взять из файла](#):
 - Описание типа в соответствующем реквизите замещается значением из новой конфигурации поставщика.
- Режим объединения [Объединить с приоритетом основной конфигурации](#):
 - Объединение типов:
 - Если тип присутствует в основной конфигурации, старой конфигурации поставщика и новой конфигурации поставщика – тип попадает в результат объединения;
 - Если тип присутствует в основной конфигурации и старой конфигурации поставщика, но отсутствует в новой конфигурации поставщика – тип не попадает в результат объединения;
 - Если тип присутствует в старой и новой конфигурациях поставщика, но отсутствует в основной конфигурации – тип не попадает в результат объединения;
 - Если тип отсутствует в старой конфигурации поставщика, но присутствует в основной конфигурации или новой конфигурации поставщика – тип попадает в результат объединения.
 - Объединение квалификаторов типов:
 - При конфликте квалификаторов их значения берутся из основной конфигурации. Конфликтом квалификаторов считается ситуация, когда квалификаторы были изменены и в новой конфигурации поставщика и в основной конфигурации.
 - Если есть отличия между основной конфигурацией и старой конфигурацией поставщика, но нет отличий между новой конфигурацией поставщика и старой конфигурацией поставщика – значения квалификаторов в результате объединения не изменяются.
 - Если есть отличия между новой конфигурацией поставщика и старой конфигурацией поставщика, но нет отличий между основной конфигурацией и старой конфигурацией поставщика – в результате объединения значения квалификаторов берутся из новой конфигурации поставщика.
- Режим объединения [Объединить с приоритетом новой конфигурации поставщика](#):
 - Объединение типов:
 - Если тип присутствует в основной конфигурации, старой конфигурации поставщика и новой конфигурации поставщика – тип попадает в результат объединения;
 - Если тип присутствует в основной конфигурации и старой конфигурации поставщика, но отсутствует в новой конфигурации поставщика – тип не попадает в результат объединения.
 - Если тип присутствует в старой и новой конфигурациях поставщика, но отсутствует в основной конфигурации – тип не попадает в результат объединения.
 - Если тип отсутствует в старой конфигурации поставщика, но присутствует в основной конфигурации или новой конфигурации поставщика – тип попадает в результат объединения.
 - Объединение квалификаторов типов:
 - при конфликте квалификаторов их значения берутся из новой конфигурации поставщика. Конфликтом квалификаторов считается ситуация, когда квалификаторы были изменены и в новой конфигурации поставщика и в основной конфигурации.
 - Если есть отличия между основной конфигурацией и старой конфигурацией поставщика, но нет отличий между новой и старой конфигурациями поставщика – значения квалификаторов в результате объединения не изменяются.
 - Если есть отличия между новой и старой конфигурациями поставщика, но нет отличий между основной конфигурацией и старой конфигурацией поставщика – значения квалификаторов берутся

из новой конфигурации поставщика.

По сравнению с обычным режимом сравнения и объединения в табличное поле добавлены две колонки, показывающие статусы по истории объектов. Расшифровка пиктограмм статусов приведена в нижней части окна. С помощью пиктограмм легко понять, какого рода и в какой конфигурации были выполнены изменения.

Так, в примере на [рис. 527](#) видно, что:

- У справочника **Банки** (из основной конфигурации) изменено имя;
- В новой конфигурации поставщика добавлен реквизит **ДеятельностьПрекращена**;
- В основной конфигурации для этого справочника реализовано несколько форм (**ФормаГруппы**, **ФормаБанкНеНайден** и т. д.), которые отсутствуют в новой конфигурации поставщика;
- В новой конфигурации поставщика реализована форма **ЗагрузкаКлассификатора**, которая отсутствует в основной конфигурации.

Для того чтобы посмотреть правила поставщика, нужно указать объект в табличном поле. Правило будет показано в реквизите, расположенном сразу под табличным полем.

Для анализа произведенных изменений в конфигурациях удобно использовать специальный фильтр, настройка которого вызывается по кнопке **Фильтр**. На экран выводится окно настройки:

Рис. 528. Настройка фильтров просмотра

Фильтр позволяет изучить изменения в конфигурациях путем выбора объектов сравнения и установки режимов показа отличий.

В качестве объектов сравнения выступает основная (текущая) конфигурация, новая и старая конфигурации поставщика. Поэтому окно настройки фильтра состоит из трех разделов, схожих по составу реквизитов. Каждый раздел предназначен для настройки режима сравнения любой пары конфигураций.

С помощью настроек фильтра можно производить установки режимов показа отличий сразу по всем

Для удобства сравнения можно использовать шаблон фильтров. В шаблоне сформированы наиболее распространенные варианты сочетаний установок.

На основе данных об изменениях и правилах поддержки пользователь принимает решение по объединению.

31.3.5. Файл описания каталога шаблонов

В тех случаях, когда шаблоны (в том числе и обновления) конфигураций располагаются на [http\(s\)](#) или [ftp](#)-ресурсах, необходимо указать перечень конфигураций, расположенных на том или ином ресурсе. Для этого служит файл описания каталога шаблонов [v8cscdsc.lst](#). Данный файл находится в корневом каталоге шаблонов конфигураций, расположенных на [http\(s\)](#) или [ftp](#)-ресурсах.

Этот файл используется для определения наличия обновлений на том или ином ресурсе в процессе выполнения команды [Конфигурация – Поддержка – Обновить конфигурацию](#).

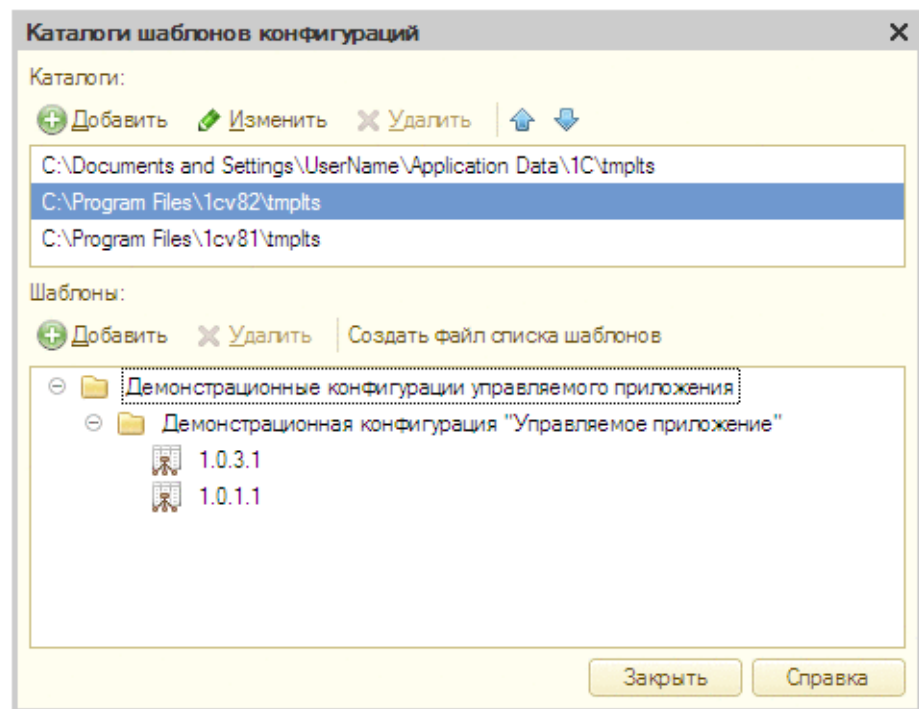


Рис. 529. Создание файла списка шаблонов

Для создания и редактирования файла описания каталога шаблонов нужно выбрать пункт [Конфигурация – Поддержка – Шаблоны конфигураций и обновлений](#).

В таблице [Каталоги](#) приведен перечень каталогов шаблонов, который получается из файла [1CStart.cfg](#) (подробнее про файл [1CStart.cfg](#) можно прочитать в книге «1С:Предприятие 8.3. Руководство администратора»). В таблице [Шаблоны](#) указывается перечень шаблонов, который присутствует в выбранном каталоге шаблонов.

Редактирование каталога шаблонов конфигураций заключается в удалении из него элементов шаблонов (или шаблонов вместе с элементами целиком) и добавлении в него шаблонов конфигураций из других каталогов.

Для добавления шаблонов в каталог нужно выбрать команду [Добавить](#) командной панели таблицы [Шаблоны](#). На экран выводится помощник выбора каталога и элементов шаблонов.

Необходимо выбрать каталог, из которого будет производиться добавление, а затем выбрать элементы, отсутствующие в текущем каталоге.

Для удаления шаблонов из каталога следует выбрать команду [Удалить](#) командной панели таблицы [Шаблоны](#). На экран выводится диалог удаления. Далее выбрать нужный способ удаления (только указанные элементы или шаблоны целиком, в которые входят эти элементы) и нажать кнопку [ОК](#).

Создание файла описания каталога шаблонов выполняется с помощью команды [Создать файл списка шаблонов](#) командной панели таблицы [Шаблоны](#). Данный пункт нужно обязательно выполнить для сохранения внесенных изменений в описание каталога шаблонов.

После этого следует переместить каталог шаблонов (вместе с файлом описания каталога) на необходимый внешний ресурс.

Общий подход к работе с шаблонами, расположенными на [http\(s\)](#) или [ftp](#)-ресурсах, следующий:

1. На диске локального компьютера создается каталог, в котором будут располагаться шаблоны, публикуемые на внешних ресурсах.
2. Выполняется установка всех шаблонов, которые планируется разместить на внешних ресурсах, в этот каталог.
3. В конфигураторе (в окне [Каталоги шаблонов конфигураций](#), см. выше) в список [Каталоги](#): добавляется созданный на шаге 1 каталог с шаблонами.
4. В окне [Каталоги шаблонов конфигураций](#) выполняется создание файла описания каталога шаблонов (кнопка [Создать файл списка шаблонов](#)). Для этого нужно предварительно встать курсором на каталог шаблонов, добавленный на шаге 3.
5. Затем необходимо перенести каталог шаблонов (вместе с файлом описания каталога шаблонов) из каталога локального компьютера на [http\(s\)](#) или [ftp](#)-ресурс.

Глава 32. Расширение конфигурации

32.1. Общая информация

Механизм расширения конфигурации – это специальный механизм, предназначенный для доработки расширяемой конфигурации без изменения этой конфигурации (в том числе без снятия с поддержки).

При рассмотрении механизма расширения конфигурации будут использоваться следующие термины:

- **Расширяемая конфигурация** – основная конфигурация информационной базы, для которой применяется расширение или для которой расширение разрабатывается.
- **Расширение конфигурации** – набор объектов конфигурации, подключаемых к расширяемой конфигурации и содержащий набор объектов, добавляемых к расширяемой конфигурации. Расширение может включать в себя как объекты расширяемой конфигурации, так и объекты, которые отсутствуют в расширяемой конфигурации.
- **Собственный объект** – самостоятельный объект конфигурации, который может находиться как в расширяемой конфигурации, так и в расширении (отчет, обработка или подсистема).
- **Заимствованный объект** – собственный объект, добавленный в расширение конфигурации.
- **Расширяемый объект** – собственный объект, для которого в заимствованном объекте изменены какие-либо параметры (свойства, формы и т. д.).
- **Расширяющий объект** – это заимствованный объект, в который внесены изменения относительно расширяемого объекта. Наличие в заимствованном объекте только контролируемых свойств не делает заимствованный объект расширяющим.
- **Результирующий объект** – это собственный объект плюс объединение всех расширяющих объектов (если расширений несколько). Если для собственного объекта нет расширяющих объектов – он становится результирующим «без изменений». Т.е. в конфигурации, с которой работает пользователь – все объекты являются результирующими, вне зависимости от наличия и количества установленных расширений.
- **Расширяющее свойство** – свойство заимствованного объекта, которое изменяет одноименное свойство расширяемого объекта.
- **Контролируемое свойство** – свойство заимствованного объекта, значение которого проверяется при подключении расширения к расширяемой конфигурации. Если при подключении расширения (в режиме 1С:Предприятие) значение контролируемого свойства в расширении не совпадет со значением этого же свойства в расширяемой конфигурации, расширение не будет подключено.

Основное назначение расширения конфигурации – это доработка прикладного решения при внедрении (или в «облаке») под нужды клиента. При этом дорабатываемую конфигурацию не надо снимать с поддержки. В результате сохраняется простота обновления типового прикладного решения, стоящего на поддержке, с необходимостью выполнять доработки. При разработке расширения следует понимать некоторые особенности функционирования расширения конфигурации. Так, расширяемая конфигурация в любой момент может быть изменена, например, в результате обновления. При этом разработчик расширения не может никак повлиять на возможность или невозможность обновления. Также следует учитывать тот факт, что в системе могут функционировать более одного расширения и автор каждого расширения (в общем случае) не знает, как функционирует другое расширение.

Расширения создаются в конфигураторе и могут быть сохранены в файл. Для добавления (подключения) расширения в прикладное решение конкретного клиента нет необходимости использовать конфигуратор. Подключить расширение можно с помощью специальной стандартной функции ([Все функции – Стандартные – Управление расширениями конфигурации](#)). Подключить расширение также можно с помощью инструментария прикладного решения, которое использует программный интерфейс, предоставляемый платформой.

В качестве расширяемых объектов могут выступать:

- Управляемые формы;
- Роли;
- Подсистемы.

Также могут быть расширены настройки начальной страницы (рабочего стола) прикладного решения.

В качестве собственных объектов расширения могут выступать:

- Отчеты;
- Обработки,
- Подсистемы.

В базовых версиях прикладных решений работа с расширениями не поддерживается.

32.2. Управление расширениями в режиме Конфигуратора

32.2.1.1. Общая информация

Для управления расширениями в конфигураторе предназначен диалог [Расширения конфигурации](#). Доступ к этому диалогу осуществляется через меню [Конфигурация – Расширения конфигурации](#). С помощью данного диалога можно создать новое расширение, удалить существующее, открыть конфигурацию расширения, сохранить расширение в файл, а также выполнять другие операции.

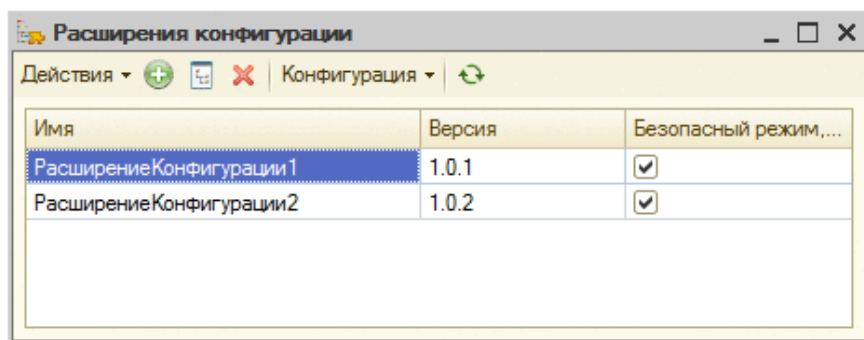


Рис. 530. Управление расширениями в конфигураторе

В общем случае получить доступ к командам работы с расширением конфигурации можно следующими способами:

1. С помощью команд командная панель панели окна [Расширения конфигурации](#).
2. С помощью контекстного меню выбранного расширения в окне [Расширения конфигурации](#).
3. С помощью меню [Действия – Конфигурация](#) окна с расширением конфигурации.

Команды, представленные в этих меню, по смыслу совпадают с командами, применяемыми для основной конфигурации. Подробное описание этих команд см. [здесь](#).

32.2.1.2. Создание расширения

При создании расширения необходимо указать имя расширения, его синоним и префикс имен методов и объектов, создаваемых в расширении. Этот префикс будет автоматически добавляться к новым отчетам, обработкам и подсистемам, которые созданы в расширении, а также к обработчикам событий, которые создаются в расширяющих формах. В дальнейшем этот префикс можно посмотреть в свойстве расширения конфигурации [Префикс имен](#).

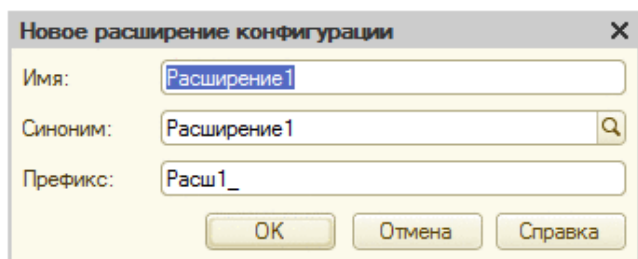


Рис. 531. Новое расширение

При создании нового расширения из расширяемой конфигурации автоматически копируются следующие свойства: [Основной режим запуска](#), [Назначение использования](#), [Основной язык](#), [Режим совместимости интерфейса](#) и [Режим совместимости](#). Эти же свойства помечаются как контролируемые.

32.2.1.3. Сохранение расширения в файл

Для установки расширения в какое-либо прикладное решение, необходим файл с расширением конфигурации. Для выполнения этого действия следует выбрать в списке расширений конфигуратора нужное расширение и использовать команду [Конфигурация – Сохранить конфигурацию в файл](#) командной панели диалога [Расширения конфигурации](#) или контекстного меню этого окна.

В результате выполнения этого действия будет сформирован файл с расширением `.cfe`. Этот файл будет необходимо подключить к прикладному решению. Подробности работы со стандартной функцией управления расширениями можно получить в книге «1С:Предприятие 8.3. Руководство администратора».

32.3. Управление расширениями в режиме 1С:Предприятие

Для управления расширениями в режиме 1С:Предприятия предназначена стандартная функция [Управление расширениями конфигурации](#). Подробности работы с этой стандартной функцией можно получить в книге «1С:Предприятие 8.3. Руководство администратора».

32.4. Работа с расширением

32.4.1.1. Создание заимствованного объекта

Для того, чтобы выполнить заимствование объекта, необходимо воспользоваться командой [Добавить в расширение](#) контекстного меню объекта конфигурации. После выполнения данной команды в расширении создается заимствованный объект. При создании заимствованного объекта заимствуются также все родительские объекты, если они еще не заимствованы в данном расширении. Подчиненные объекты не заимствуются. Поясним поведение на примерах.

Предположим, что в расширяемой конфигурации существует справочник [Товары](#), который представлен некоторой структурой, которая представлена на [рис. 532](#).

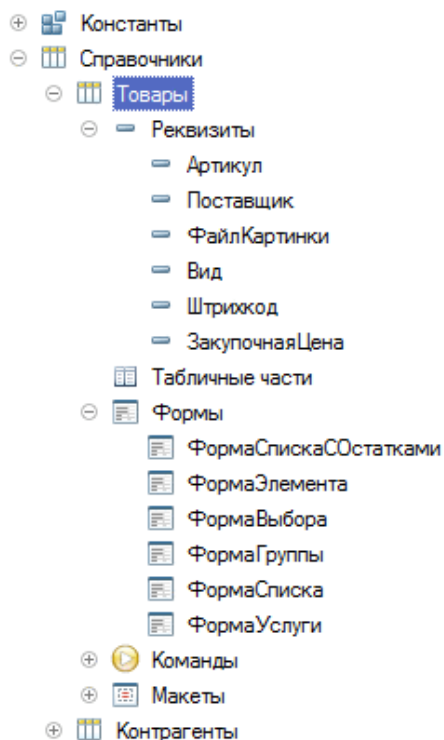


Рис. 532. Структура справочника [Товары](#)

Если добавить в расширение непосредственно сам справочник, то никакие подчиненные объекты добавлены не будут, что видно на [рис. 533](#).

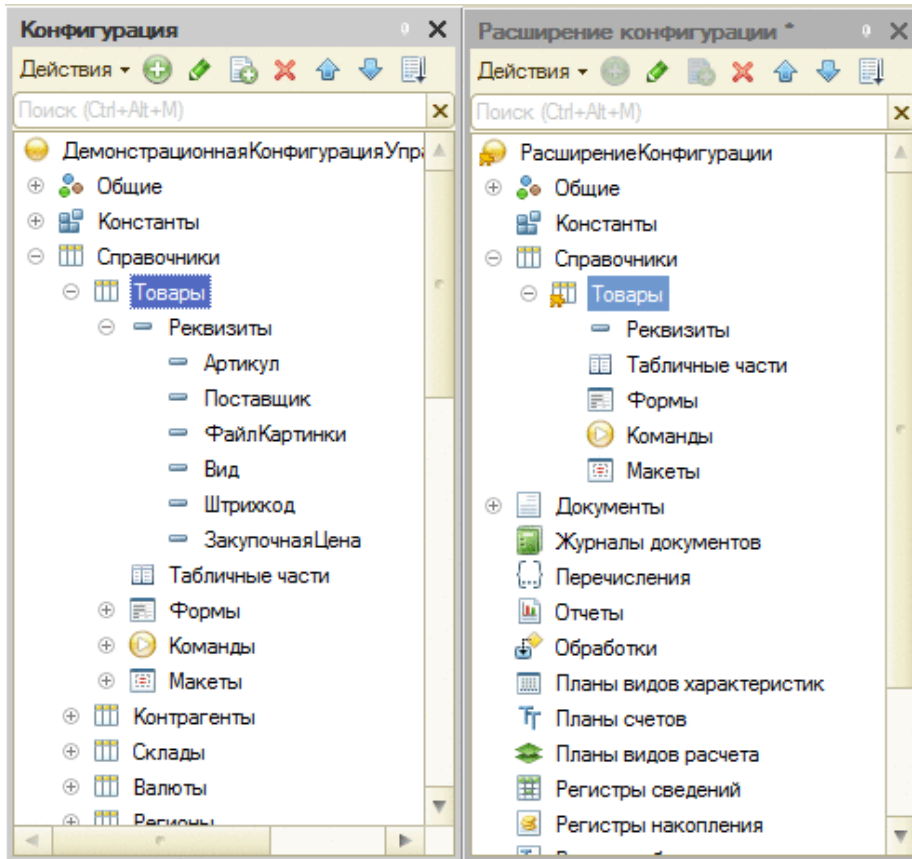


Рис. 533. После добавления объекта

Если теперь добавить форму справочника **Товары**, например форму элемента, то сам справочник уже заимствован и будет заимствована форма и все объекты конфигурации, которые образуют данные, отображаемые на форме. Заимствование выполняется в объеме, достаточном для функционирования редактора формы.

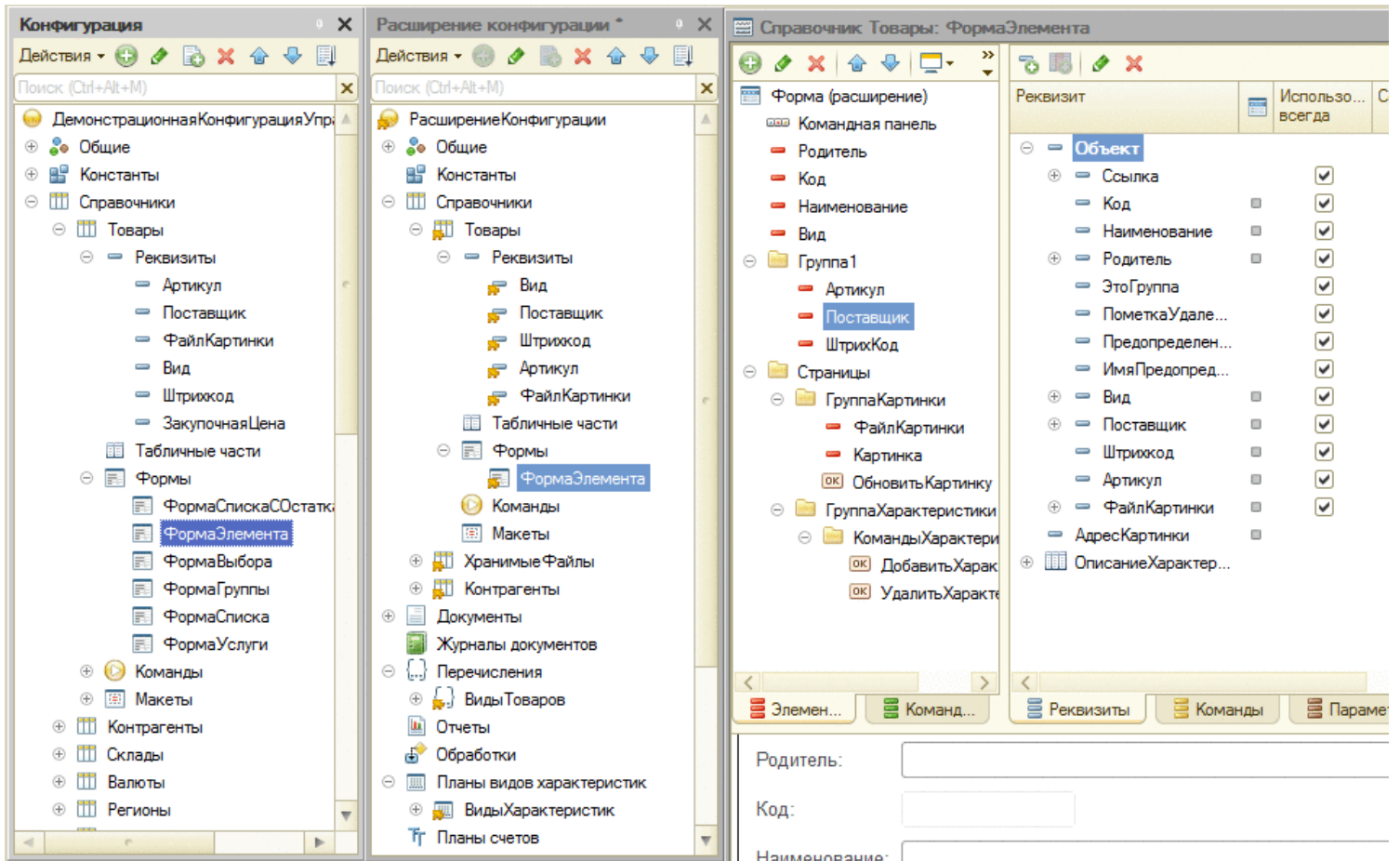


Рис. 534. После добавления объекта с родителями

На [рис. 534](#) видно, что кроме добавляемой формы `ФормаЭлемента` были добавлены также:

- Все реквизиты, которые присутствуют на форме;
- Все объекты, которые образуют типы реквизитов, присутствующих на форме.

При заимствовании составных типов будут заимствованы только примитивные типы (`Строка`, `Число` и т. д.), а если в составном типе таких типов нет, то будет заимствован только один тип (первый по порядку) и связанный с ним объект конфигурации.

32.4.1.2. Работа с заимствованным объектом

Работа с заимствованным объектом имеет некоторые особенности.

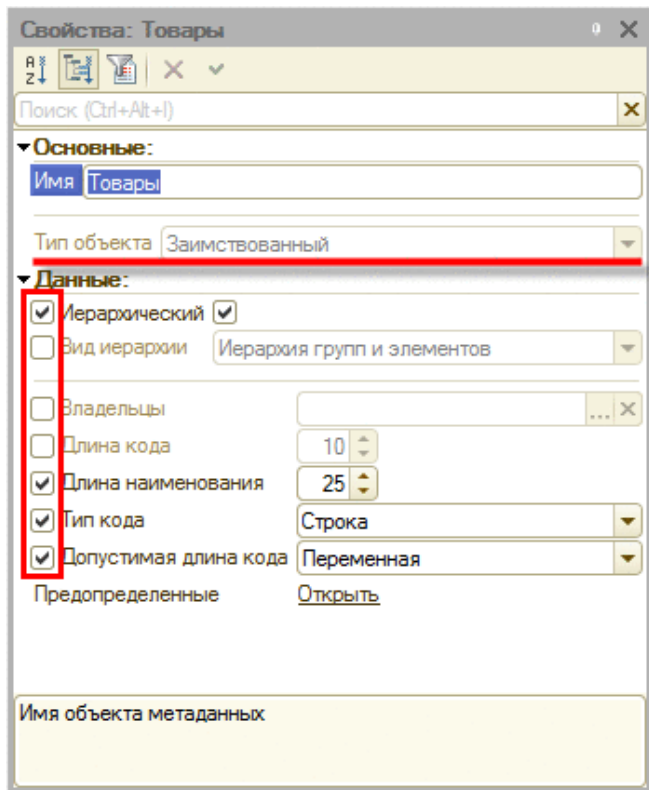


Рис. 535. Свойства заимствованного объекта

Во-первых, у заимствованного объекта присутствуют только те свойства, которые могут быть установлены контролируемыми (на [рис. 535](#) представлен состав свойств заимствованного справочника). Эта возможность (установка контролируемых свойств, подробнее см. [здесь](#)) является второй особенностью панели свойств заимствованного объекта. Для указания контролируемого свойства, следует установить флажок перед именем свойства. На [рис. 535](#) видно, что для справочника `Товары` указаны следующие контролируемые свойства: `Иерархический`, `Длина наименования`, `Тип кода` и `Допустимая длина кода`.

Изменение свойств заимствованного объекта следует делать осторожно. Если какое-либо свойство указано контролируемым, то расширение не будет подключено в том случае, если значение этого свойства в расширяемой конфигурации отличается. Также следует помнить, что `Имя` объекта метаданных всегда является контролируемым реквизитом. Если в расширяемой конфигурации, на момент создания расширения, справочник имел имя `Товары`, а потом был переименован в `Номенклатура`, то расширение перестанет работать до тех пор, пока для заимствованного справочника в расширении также не будет установлено имя `Номенклатура`. То же самое верно для значений других контролируемых свойств.

32.4.1.3. Создание собственного объекта

В расширении можно не только создавать заимствованные объекты, но и собственные. Собственными могут быть следующие объекты: отчет, обработка, подсистема.

Создание и редактирование этих объектов ничем не отличается от аналогичных действий в расширяемой конфигурации. Контроль уникальности имен собственных объектов выполняется:

- При подключении расширения – среди имен расширяемой конфигурации и всех расширений;

- При разработке расширения – среди имен расширяемой конфигурации и того расширения, в котором создается собственный объект.

Если в расширяемой конфигурации, например, существует обработка [ПоискВДанных](#), то ни в одном расширении не может существовать одноименной обработки.

При создании собственного объекта, этот объект получает префикс, который задается при создании расширения.

32.4.1.4. Блокировка расширений

В связи с тем, что список расширений является единым и для доступа из конфигуратора и для доступа из «1С:Предприятия», система не позволяет выполнять взаимоисключающие действия. Другими словами, если в Конфигураторе открыто окно с расширением, то его невозможно удалить из списка расширений в режиме «1С:Предприятия». Расширение, во время записи в информационную базу в режиме «1С:Предприятия», в конфигураторе является недоступным для любых операций с ним.

32.5. Объекты расширения

32.5.1.1. Управляемая форма

После того, как форма добавлена в расширение, визуально она выглядит точно так, как эта же форма выглядит в расширяемой конфигурации в момент добавления ее в расширение. При этом модуль расширяющей формы пустой. В момент добавления конфигуратор «запоминает» добавляемую форму в расширении. В дальнейшем такая форма будет называться **сохраненной**. Расширяемая форма автоматически не синхронизируется с расширением. Как следствие, изменение расширяемой формы никак не отражается в расширяющей форме (в режиме Конфигуратора). В том случае, если прикладному разработчику необходимо обновить форму в расширении, ему следует использовать команду [Расширения – Обновить расширение формы](#) редактора формы. Обновление может потребоваться в том случае, если необходимо перенести в расширения изменения и дополнения, которые выполнены в расширяемой конфигурации после создания расширения формы.

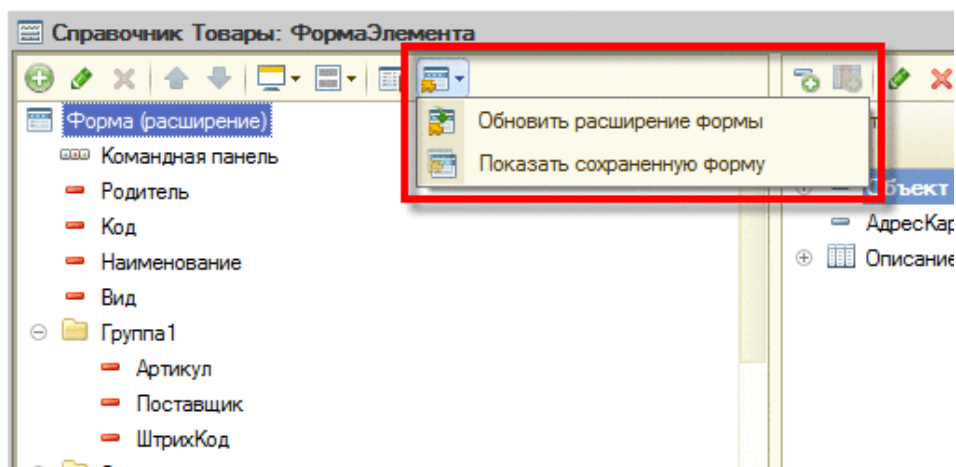


Рис. 536. Команды редактора формы

Если требуется посмотреть, как выглядит сохраненная форма – следует использовать команду [Расширения – Показать сохраненную форму](#) в редакторе заимствованной управляемой формы.

В режиме 1С:Предприятие происходит формирование результирующего объекта объединением расширяемого и всех расширяющих объектов. Процесс объединения работает следующим образом:

1. Выполняется объединение расширяемой конфигурации и первого (по порядку регистрации) расширения.
2. Объединение повторяется для каждого следующего зарегистрированного расширения, при этом в качестве расширяемой конфигурации выступает результат предыдущего объединения. С точки зрения каждого следующего объединения, результат предыдущего объединения будет считаться расширяемым объектом.

В процессе объединения каждой компоненты (элемент, реквизит, параметр или команда) управляемой формы анализируются изменения в расширяемой и расширяющей формах относительно сохраненной. По результатам анализа принимается решение, какая компонента будет отображаться в результирующей форме:

1. Если изменений нет – компонента берется из расширяемого объекта;

2. Если изменение есть только в расширяемой форме – компонента берется из расширяемой формы;
3. Если изменение есть только в расширяющей форме – компонента берется из расширяющей формы;
4. Если изменение есть и в расширяемой форме и в расширяющих формах – компонента берется из расширяющей формы.

Сопоставление компонентов управляемой формы выполняется по имени соответствующего элемента.

Вышеописанный режим не действует:

1. Для условного оформления. В этом случае происходит объединение элементов условного оформления из расширяемой формы и всех расширяющих форм аналогично тому, как это происходит при объединении конфигураций.
2. Командный интерфейс результирующей формы формируется путем добавления элементов командного интерфейса из расширяющей формы к командному интерфейсу расширяемой формы. При этом возможна ситуация, когда на форме будут несколько команд с одинаковым представлением, т. к. они образованы командами из разных расширений, но с одинаковым представлением.

При определении порядка элементов результирующей управляемой формы вначале определяется возможность расположить элементы без конфликтов размещения. Если такая возможность есть, то порядок элементов результирующей формы определяется следующим образом:

- Элементы формы, добавленные в начало списка (не важно, в расширении или в расширяемой конфигурации), останутся в начале списка независимо от изменения порядка остальных элементов. Сначала будут идти элементы формы из расширений (в порядке следования), а потом из расширяемой конфигурации.
- Элементы формы, добавленные в конец списка (не важно, в расширении или в расширяемой конфигурации), останутся в конце списка независимо от изменения порядка остальных элементов. Сначала будут идти элементы формы из расширяемой конфигурации, потом из расширений (в порядке следования).
- Также существует возможность располагать рядом с элементом расширяемой формы другие элементы формы. Для этого следует создать в расширении группу, в которую поместить элемент расширяемой формы, и те элементы, которые необходимо рядом. Тогда при смене местоположения оригинального элемента в расширяемой форме, в результирующей форме на новом месте окажется и вся группа, в которую был помещен этот элемент.

Если без конфликтов порядок определить нельзя, используется следующий алгоритм:

- Исходным выступает список элементов расширения в порядке, который установлен в расширении.
- Для определения положения элемента, добавленного в основной конфигурации, находится элемент, существующий в сохраненной конфигурации и расположенный перед добавленным элементом (в порядке основной конфигурации):
 - В соответствии с порядком элементов в расширении, определяется, какой элемент является следующим – существовавший или добавленный в расширении:
 - Если существовавший, то элемент основной конфигурации добавляется после найденного элемента.
 - Если элемент или элементы, добавленные в расширении, то элемент основной конфигурации добавляется после них.
 - Если в основной конфигурации было добавлено несколько элементов, то последующие элементы добавляются после предшествующих добавленных, даже если предшествующие уже были добавлены расширением.
 - Если элемент, после которого элементы были добавлены, не найден в расширении (например, удален), то ищется предыдущий существующий в порядке основной конфигурации. Если при поиске достигнуто начало списка элементов, то новый элемент (или группа новых элементов) добавляется в начало результирующего списка, но после новых элементов расширения добавленных в начало списка.
- Новые элементы основной конфигурации, добавленные в начало списка, добавляются сразу после новых элементов расширения, добавленных в начало списка.
- Новые элементы основной конфигурации, добавленные в конец списка, добавляются перед новыми элементами расширения, добавленными в конец списка.

- Если при добавлении оказывается, что элемент уже был добавлен расширением, то такой элемент пропускается.
- Удаляются элементы, которые были удалены в основной конфигурации по сравнению с сохраненной версией.

Как уже было сказано выше, модуль расширяемой формы изначально пустой. Результирующий модуль формы получается «добавлением» текстов всех модулей форм из расширений в модуль расширяемой формы. Фактической модификации текстов не выполняется, но с точки зрения понимания такое упрощение вполне допустимо. В результате такого «добавления» могут возникнуть конфликты имен. В качестве примера можно упомянуть обработчики событий элементов формы или методы модуля формы, реализующие какую-либо функциональность. Такая ситуация может возникнуть, когда обновление расширяемой конфигурации привносит в расширяемую форму метод, одноименный аналог которого уже создан разработчиков расширения. Чтобы этого избежать, система при создании обработчиков событий использует при формировании имени обработчика значение свойства **Префикс имен** того расширения, в котором редактируется форма. Прикладному разработчику также рекомендуется использовать префикс для имен методов, добавляемых в расширяющем объекте. Если в результате образования результирующей формы все-таки образуется конфликт имен – возникает стандартная ошибка времени исполнения и результирующая форма оказывается полностью не работоспособной.

Если для какого-то события имеются обработчики в расширяемой форме, а также в расширяющих формах, то при выполнении обработчиков действуют следующие правила:

- Сначала вызываются обработчики из расширений (в порядке, обратном регистрации расширений в списке), последним вызывается обработчик из расширяемой формы. Следствием является тот факт, что смена порядка регистрации расширений приводит к смене последовательности вызова обработчиков.
- Если в любом из обработчиков для параметра **Отказ** (при его наличии среди параметров обработчика) устанавливается значение **Ложь** – дальнейшее выполнение обработчиков прекращается.

Расширение может влиять на исполнение программного кода после окончания работы обработчика. Для этого следует использовать специальные методы управляемой формы. С помощью метода **УстановитьВыполнениеОбработчиковСобытия ()** можно прекратить дальнейшее выполнение обработчиков в цепочке. Например, если расширение «не хочет», чтобы срабатывал обработчик расширяемой формы, то в расширяющей форме, в коде обработчика должен стоять вызов **УстановитьВыполнениеОбработчиковСобытий (Ложь) ;**. Следует пользоваться этой возможностью с осторожностью, т. к. использование данного метода может существенно исказить функционирование дорабатываемого механизма после обновления расширяемой конфигурации или при подключении дополнительных расширений.

Метод **УстановитьВыполнениеПослеОбработчиковСобытия ()** позволяет установить метод обратного вызова, который будет вызван после того, как сработают обработчики расширений и расширяемого объекта. У подключаемого метода должен быть набор параметров, эквивалентный оригинальному обработчику, плюс параметр **ДополнительныйПараметр**. В этом случае подключенные методы будут вызваны в порядке регистрации расширений (в отличие от обработчиков событий).

Прерывание цепочки вызова и подключение методов обратного вызова могут использоваться совместно.

Рассмотрим более подробно порядок срабатывания обработчиков и способы управления этим порядком.

В примере будут рассматриваться:

- Расширяемая конфигурация, в которой существует справочник **Товары**, в которого создана форма элемента.
- **Расширение конфигурации 1** (с префиксом **PK1**), в которой расширяется форма элемента справочника **Товары**.
- **Расширение конфигурации 2** (с префиксом **PK2**), в которой также расширяется форма элемента справочника **Товары**.
- В каждой форме задан обработчик для события **ПриИзменении** элемента формы, связанного с реквизитом справочника **Наименование**.
- Расширения зарегистрированы в следующем порядке:
 - **Расширение конфигурации 1;**
 - **Расширение конфигурации 2.**

В текстах примеров всегда будет показываться общий модуль формы, при этом надо помнить, что методы с префиксами располагаются в соответствующих расширениях.

Вначале существуют только обработчики событий:

[Копировать в буфер обмена](#)

```
// Расширяемая форма
&НаКлиенте
Процедура НаименованиеПриИзменении(Элемент)
    Сообщить ("ПриИзменении расширяемого объекта" );
КонецПроцедуры
// Расширение конфигурации 1
&НаКлиенте
Процедура РК1_НаименованиеПриИзменении(Элемент)
    Сообщить ("ПриИзменении расширения РК1" );
КонецПроцедуры
// Расширение конфигурации 2
&НаКлиенте
Процедура РК2_НаименованиеПриИзменении(Элемент)
    Сообщить ("ПриИзменении расширения РК2" );
КонецПроцедуры
```

В результате изменения поля управляемой формы, связанного с реквизитом **Наименование**, события будут вызваны в следующем порядке:

[Копировать в буфер обмена](#)

```
ПриИзменении расширения РК2
ПриИзменении расширения РК1
ПриИзменении расширяемого объекта
```

Добавим в **Расширение конфигурации 2**, в обработчик `РК2_НаименованиеПриИзменении()`, запрет дальнейшего исполнения обработчиков:

[Копировать в буфер обмена](#)

```
// Расширение конфигурации 2
&НаКлиенте
Процедура РК2_НаименованиеПриИзменении(Элемент)
    Сообщить ("ПриИзменении расширения РК2" );
    УстановитьВыполнениеОбработчиковСобытия(Ложь) ;
КонецПроцедуры
```

В этом случае события будут вызваны в следующем порядке:

[Копировать в буфер обмена](#)

```
ПриИзменении расширения РК2
```

Видно, что обработчики из **Расширения конфигурации 1** и расширяемой конфигурации вызваны не были.

Если порядок регистрации расширений будет другим (**Расширение конфигурации 2**, **Расширение конфигурации 1**), то последовательность вызова обработчиков станет такой:

[Копировать в буфер обмена](#)

```
ПриИзменении расширения РК1
ПриИзменении расширения РК2
```

Из-за изменения последовательности регистрации расширений, оказался не вызванным только обработчик расширяемой конфигурации. Из данного примера видно, что изменение порядка регистрации расширений существенно влияет на количество «отмененных» вызовов обработчиков.

Рассмотрим более подробно поведение системы при регистрации методов обратного вызова. Для этого установим прежний порядок регистрации расширений:

- **Расширение конфигурации 1**;
- **Расширение конфигурации 2**.

Добавим в обработчик события `ПриИзменении` каждого расширения вызов метода `УстановитьВыполнениеПослеОбработчиковСобытия()`. «Итоговый» модуль формы будет выглядеть следующим образом:

[Копировать в буфер обмена](#)

```
// Расширяемая форма
&НаКлиенте
Процедура НаименованиеПриИзменении(Элемент)
    Сообщить ("ПриИзменении расширяемого объекта");
КонецПроцедуры
// Расширение конфигурации 1
&НаКлиенте
Процедура РК1_НаименованиеПриИзменении(Элемент)
    Сообщить ("ПриИзменении расширения РК1");
    УстановитьВыполнениеПослеОбработчиковСобытия ("РК1_НаименованиеПриИзмененииПосле", Неопределено);
КонецПроцедуры
&НаКлиенте
Процедура РК1_НаименованиеПриИзмененииПосле(Элемент, ДополнительныйПараметр)
    Сообщить ("ПриИзменении после расширения РК1");
КонецПроцедуры
// Расширение конфигурации 2
&НаКлиенте
Процедура РК2_НаименованиеПриИзменении(Элемент)
    Сообщить ("ПриИзменении расширения РК2");
    УстановитьВыполнениеПослеОбработчиковСобытия ("РК2_НаименованиеПриИзмененииПосле", Неопределено);
КонецПроцедуры
&НаКлиенте
Процедура РК2_НаименованиеПриИзмененииПосле(Элемент, ДополнительныйПараметр)
    Сообщить ("ПриИзменении после расширения РК2");
КонецПроцедуры
```

В этом случае порядок вызова событий и методов будет следующим:

[Копировать в буфер обмена](#)

```
ПриИзменении расширения РК2
ПриИзменении расширения РК1
ПриИзменении расширяемого объекта
ПриИзменении после расширения РК1
ПриИзменении после расширения РК2
```

Видно, что обработчики вызываются в порядке, обратном порядку регистрации расширений, а методы обратного вызова – в прямом порядке.

Добавим в **Расширение конфигурации 2**, в обработчик `РК2_НаименованиеПриИзменении()`, запрет дальнейшего исполнения обработчиков:

[Копировать в буфер обмена](#)

```
&НаКлиенте
Процедура РК2_НаименованиеПриИзменении(Элемент)
    Сообщить ("ПриИзменении расширения РК2");
    УстановитьВыполнениеПослеОбработчиковСобытия ("РК2_НаименованиеПриИзмененииПосле", Неопределено);
    УстановитьВыполнениеОбработчиковСобытия (Ложь);
КонецПроцедуры
```

В результате будет получен следующий порядок вызова обработчиков и методов:

[Копировать в буфер обмена](#)

```
ПриИзменении расширения РК2
ПриИзменении после расширения РК2
```

Таким образом, видно, что прикладной разработчик может ограниченно управлять выполнением обработчиков и методов обратного вызова, но пользоваться этим механизмом следует достаточно аккуратно.

32.5.1.2. Командный интерфейс формы

Командный интерфейс формы дополняется командными интерфейсами расширений. Команды, добавляемые в командный интерфейс формы вручную, будут отображаться после команд расширяемой конфигурации (в режиме «1С:Предприятие»). Следствием этого является то, что в результирующей форме могут одновременно существовать несколько команд с одинаковым представлением.

Удалить команду расширяемой конфигурации невозможно.

Необходимо учитывать, что в редакторе формы в конфигураторе, разработчик не будет видеть фрагменты командного интерфейса, которые предоставляются расширяемой конфигурацией.

32.5.1.3. Подсистемы

Подсистема может быть как заимствованным, так и собственным объектом расширения.

Из состава заимствованной подсистемы нельзя удалить объекты конфигурации, которые добавлены туда в расширяемой конфигурации. Но в состав заимствованной подсистемы можно добавить другие объекты: как собственные, так и заимствованные (относительно расширения).

Собственная подсистема расширения может включать как собственные объекты расширения, так и заимствованные объекты.

32.5.1.4. Права доступа

Доступ к собственным объектам расширения зависит от того, есть в расширении заимствованные роли или нет:

- В расширении нет заимствованных ролей – в этом случае все собственные объекты расширения доступны для всех пользователей прикладного решения с установленным расширением. Права доступа расширяемой конфигурации не влияют на доступ к собственным объектам расширения.
- В расширении есть заимствованные роли – в этом случае доступ к собственным объектам расширения устанавливается для заимствованных ролей. Для остальных ролей доступ к собственным объектам отсутствует. Таким образом, в расширение имеет смысл добавлять те роли, для которых необходимо установить возможность использования собственных объектов расширения.

Также следует помнить о следующих особенностях работы с правами доступа в расширении:

- В расширении не поддерживается изменение общих прав доступа (доступность клиентских приложений, право на доступ к [Все функции](#) и т. д.).
- В заимствованных ролях не поддерживается изменение доступа к собственным объектам расширяемой конфигурации.
- В заимствованных ролях не поддерживается изменение ограничений доступа к данным, а также создание новых ограничений.
- В расширении не поддерживается создание новых ролей.

32.5.1.5. Отчеты и обработки

Создание нового отчета или обработки не отличается от таковой в конфигурации, не содержащей расширения (см. [здесь](#)). В тоже время, имеется возможность расширить существующие формы отчетов или обработок расширяемой конфигурации, выполнив заимствование необходимого объекта (см. [здесь](#)).

32.5.1.6. Профили безопасности

При подключении расширения к прикладному решению имеется возможность указать, будет расширение выполняться в безопасном режиме или нет. Это можно сделать:

1. Явным указанием флажка [Безопасный режим](#). В этом случае расширение будет выполняться в безопасном режиме (см. [здесь](#)) или небезопасном режиме (в зависимости от состояния флажка).
2. Указанием имени профиля безопасности, в рамках которого будет выполняться расширение. Данная возможность доступна только в том случае, если прикладное решение используется в клиент-серверном варианте. Подробнее о профилях безопасности можно прочитать в книге «1С:Предприятие 8.3. Клиент-серверный вариант. Руководство администратора».

Указание профиля безопасности при подключении расширения указывает, что режим работы расширения определяется настройками указанного профиля безопасности. В профиле безопасности доступ к расширению указывается в настройках элемента [Внешние модули](#).

Если в профиле безопасности запрещено небезопасное использование внешних модулей, а расширение требует такое использование, то его (расширение) можно внести в список исключений. Чтобы получить значение контрольной суммы расширения для добавления в список исключений, следует воспользоваться возможностями диалогов работы с расширениями или программного интерфейса работы с расширениями. Полученное значение контрольной суммы следует ввести в соответствующее поле в консоли администрирования кластера.

При подключении расширения проверяется возможность включения небезопасного режима (если требуется) и если текущее расширение не соответствует требованиям указанного профиля безопасности – подключение расширения не происходит и выдается ошибка.

Для изменения настроек безопасного режима при подключении расширения необходимо, чтобы пользователь, выполняющий эти настройки, обладал правом [Администрирование](#) (в дополнение к праву [АдминистрированиеРасширенийКонфигурации](#), которое позволяет выполнять остальные действия с расширениями).

32.6. Подключение расширений и время их работы

Подключение расширения к сеансу пользователя выполняется перед вызовом обработчика события [УстановкаПараметровСеанса](#) и остается подключенным до завершения сеанса. Обновление расширения (с помощью инструментов администрирования) не приводит к обновлению расширения в существующих сеансах. Существующие сеансы продолжают работу с той «версией» расширения, которое было доступно в момент старта сеанса. Для того чтобы пользовательский сеанс начал работать с новой «версией» расширения – необходимо выполнить перезапуск сеанса.

Доступный пользователю набор расширений зависит от набора используемых разделителей в момент старта сеанса. При сохранении расширения в информационную базу, оно будет записано со значениями разделителей, которые установлены в текущем сеансе. В дальнейшем расширение используется только в тех сеансах, которые имеют точно такой же набор разделителей. В связи с этим, к сеансу одного и тот же пользователя, подключающегося к одной и той же разделенной информационной базе с разным набором разделителей, может быть подключен разный набор расширений.

При попытке подключения расширений, каждое расширение последовательно проверяется на соответствие контролируемых реквизитов между расширением и конфигурацией информационной базы. Если хотя-бы один контролируемый реквизит не проходит проверку, пользователю выдается сообщение об ошибке и это расширение не подключается. При проверке свойств с несколькими значениями (составной тип, состав общего реквизита) значение из расширения контролируется на вхождение – все значения, которые есть в расширении, должны быть в расширяемой конфигурации, однако часть состава расширяемой конфигурации может не входить в контролируемое свойство. Как следствие, все объекты расширения перестают участвовать в формировании результирующей конфигурации. Ошибки подключения выдаются только в сеансах клиентских приложений.

32.7. Программное управление расширениями

В случае необходимости реализации собственных инструментов работы с расширениями следует использовать специальный программный интерфейс. Его методы доступны в том случае, если текущий пользователь обладает правом [АдминистрированиеРасширенийКонфигурации](#). Текущий сеанс должен работать в небезопасном режим в том случае, если будет использоваться метод [Записать \(\)](#) объекта [РасширениеКонфигурации](#).

Программный интерфейс предоставляется посредством специального менеджера расширений конфигурации, который представлен свойством глобального контекста [РасширенияКонфигурации](#). Подробное описание методов приведено в синтакс-помощнике.

Глава 33. Сервисные возможности

Система «1С:Предприятие» включает в себя множество различных сервисных возможностей и вспомогательных режимов. Они описаны в этой главе.

33.1. Управление окнами

33.1.1. Состояние (режим размещения) окна

33.1.1.1. Установка режимов размещения

Окна могут размещаться на экране в разных режимах:

- обычное расположение, когда окно располагается только внутри рабочей области программы;
- свободное расположение, когда окно может свободно перемещаться по всему экрану (не только внутри рабочей области программы);
- прикрепляемое, когда окно может «прикрепиться» к границам рабочей области программы;
- прячущееся.

Для изменения способа размещения в контекстном меню заголовка окна нужно выбрать требуемый режим размещения.

Окна в режиме [Свободное](#) всегда располагаются поверх окон, находящихся в других режимах. Если несколько окон находятся в режиме [Свободное](#), то активное окно всегда будет поверх других окон.

Для окон в состоянии [Обычное](#) и [Свободное](#) доступна установка способности окна соединяться. Если для пары окон, находящихся в одинаковом состоянии, установлено свойство [Соединяемое](#) (в контекстном меню заголовка окна), то данные окна могут соединяться. Чтобы соединить окна, находящиеся в состоянии [Обычное](#), требуется при их перемещении нажать клавишу [Shift](#). Для соединенных окон также можно устанавливать способность соединяться с другими окнами.

Если свойство [Соединяемое](#) не установлено, то выбор режимов [Прикрепленное](#) и [Прячущееся](#) недоступен.

Для режимов [Прикрепленное](#) и [Прячущееся](#) свойство [Соединяемое](#) всегда установлено и не может быть изменено.

Особого описания требует режим состояния окна [Прикрепленное](#). В этом режиме окно может быть:

- прикреплено к одной из сторон окна configurатора;
- прикреплено к любой стороне другого окна, для которого установлен режим состояния [Прикрепленное](#);
- расположено поверх другого прикрепляемого окна (совмещенные окна).

Если для окна выбран режим [Прикрепленное](#), то окно прикрепляется к одной из границ рабочей области или к другому окну, находящемуся в режиме [Прикрепленное](#).

Для изменения размера прикрепленного окна нужно подвести указатель мыши к свободной (обращенной в сторону рабочей области) границе и потянуть границу мышью.

В режиме [Свободное](#) при перетаскивании окна и приближении к границе рабочей области программы или другому окну, находящемуся в режиме [Прикрепленное](#), контур окна может измениться скачкообразно. Если в этот момент отпустить кнопку мыши, то окно прикрепится и его состояние станет [Прикрепленное](#).

Для прикрепленных окон изменить состояние можно также с помощью перетаскивания окна мышью. Чтобы перетащить окно, следует в области заголовка окна нажать левую кнопку мыши и, не отпуская ее, перетащить окно в другое место. Состояние окна меняется на [Свободное](#).

Для прикрепляемых окон важной особенностью являются различные способы прикрепления нескольких окон к одной стороне окна configurатора или другого окна. Окна можно расположить слоями, когда каждое окно будет занимать всю сторону рабочей области окна configurатора или другого окна. Можно расположить окна последовательно, когда каждое окно будет расположено в одном слое вдоль границы другого окна. Если прикрепляемых окон более двух, то можно часть из них расположить слоями, а часть – последовательно.

Для расположения одного прикрепляемого окна (совмещение) поверх другого нужно выполнить следующие действия:

- захватить мышью заголовок окна;
- переместить окно так, чтобы его заголовок оказался над заголовком другого прикрепляемого окна, при этом контур перетаскиваемого окна должен иметь снизу контур закладки;
- отпустить кнопку мыши.

Результатом этих действий будет прикрепляемое окно, внизу которого будут закладки. Текст закладок совпадает с текстом заголовков окон. На закладках располагаются совмещенные окна.

Для показа нужного окна следует щелкнуть нужную закладку.

Команда **Закреть** в этом случае закрывает только активную закладку.

Чтобы разделить совмещенные окна, достаточно захватить мышью закладку нужного окна и перетащить ее в сторону (следует контролировать перенос контуром перемещаемого окна).

Если при переносе указатель мыши остается в области закладок, то таким способом можно изменить порядок следования закладок.

Если на экране есть совмещенные окна, то при совмещении с ними другого окна можно перетаскивать его не на заголовок, а в область закладок. При этом можно сразу выбрать порядок следования закладок.

Окна можно размещать на экране в режиме **Прячущееся**. Если выбрать режим **Прячущееся**, то сбоку рабочей области программы появляется дополнительная строка для ярлыков окон, в которую добавляется ярлык текущего окна, а само окно прикрепляется к этой стороне рабочей области. Расположение этой строки определяется расположением окна в режиме **Прикрепленное**. В этом режиме окно остается на экране, пока активно. Как только активизировать любое другое окно, предыдущее окно прячется. Чтобы показать это окно, необходимо подвести указатель мыши к ярлыку данного окна (щелкать кнопкой мыши не нужно). При выводе указателя мыши вне окна прячущееся окно автоматически убирается с экрана.

Для соединенных окон также можно устанавливать режим состояния, которое отличается от состояния совмещенных окон. Например, удобно установить режим **Прячущееся**. В этом случае в дополнительной строке показываются только пиктограммы ярлыков окон. При подведении к любой пиктограмме указателя мыши показывается окно, а в дополнительной строке кроме пиктограммы показывается и его наименование.

33.1.1.2. Восстановление положения окна

Как описывалось выше, положение, размер и состояние окон можно изменять. При этом при закрытии последние параметры показа окна сохраняются, и при повторном открытии окно открывается с последними значениями параметров показа.

Для восстановления первоначального положения, размера и состояния окна следует в контекстном меню заголовка окна (или заголовка панели окон) выбрать пункт **Восстановить положение окна**. При этом запомненные значения параметров показа сбрасываются и восстанавливаются первоначальные, какие были при первом открытии. Окно закрывается и вновь открывается с новыми параметрами показа.

33.1.1.3. Выбор режима окон для форм объектов

Режим состояния окон для форм прикладных объектов при их первом открытии в режиме 1С:Предприятие можно настроить. Настройка производится в категории свойств **Окно** для каждой формы объекта.

Состояние окна – указывается, в каком состоянии будет окно при первом открытии. Если выбрано **Обычное**, то для данного окна недоступно изменение состояния окна.

Соединяемое окно – указывает на способность окна соединяться с другими окнами.

Положение окна – если указано **Авто**, то положение окна определяется в зависимости от состояния и размеров рабочей области. Если указано **Центрировать**, то положение окна при первом открытии будет выбираться по центру рабочей области. Если указано **Не перекрывать владельца**, то при открытии подчиненного окна при наличии свободного пространства рабочей области экрана положение окна будет выбираться таким образом, чтобы не перекрывать окно владельца или элемент управления. Если не найдено место для его размещения, подчиненное окно «пытается» не закрывать левую часть окна владельца при условии, что удастся освободить 50 % ширины окна владельца.

Положение прикрепленного окна – определяется положение окна формы в прикрепленном состоянии относительно рабочей области.

Изменять размер – разрешает или запрещает изменять размер. Выбор значения **НеИзменять** не влияет на способность изменения размера в состоянии **Прикрепленное** и **Прячущееся**.

33.1.2. Диалог настройки окон

Диалог предназначен для работы со списком всех открытых окон. С его помощью производится переход к выбранному открытому окну, сохранение сделанных изменений, управление расположением окон, закрытие одного, группы или всех открытых окон.

Для вызова диалога окон нужно выбрать пункт **Окна – Окна...** В диалоге выводится список окон, открытых к настоящему моменту. В список включаются только те окна, которые имеют статус состояния **Обычное**.

Если содержимое открытого окна модифицировано, то такое окно в строке списка помечается символом «*», располагающимся слева от наименования окна.

Все действия выполняются с одним или несколькими окнами, указанными в списке. Выбор или отмена выбора нескольких строк осуществляется стандартным образом.

Действия, которые выполняются с одним или группой открытых окон, описаны в следующей таблице.

Кнопка	Действие	Число выбранных окон (условие)
Активизировать	Активизирует указанное окно	Одно
Сохранить	Производится сохранение произведенных изменений для некоторых видов документов (например, текстовых и табличных документов), для которых это предусмотрено	Любое
Закрыть окна	Закрываются все выбранные окна	Любое
Расположить	Производится управление расположением окон на экране. Горизонтально – располагает выбранные окна слева направо. Вертикально – располагает выбранные окна сверху вниз. Подряд – располагает выбранные окна подряд	Больше одного
Соединить	Производится соединение двух окон	Установлено разрешение соединения
Разъединить	Производится разъединение соединенных окон	
ОК	Закрывает диалог «Окна»	Любое

33.1.3. Использование режима «Новое окно»

Пункт **Окна – Новое окно** позволяет сделать копию любого окна редактора форм, текстового или табличного редактора. При этом на рабочей области программы открывается новое окно-копия. Его содержимое полностью совпадает с исходным. В заголовке окна программа выводит порядковый номер окна.

В этом режиме удобно производить просмотр, сравнение и редактирование информации, размещенной в окне. Изменения, внесенные в любом месте любого окна, сразу отображаются во всех других окнах-копиях.

Рекомендуется использовать данный режим, если окно представляет форму объекта конфигурации, элементы которой распределены по страницам, и требуется одновременно просматривать эти страницы; или требуется одновременно просматривать модуль и диалог формы. Таким же образом использование данного режима позволяет редактировать тексты или макеты. При этом в одном окне можно просматривать одну часть текста или макета, а в другом – другую часть.

33.1.4. Служебные окна

В процессе работы с конфигуратором для выдачи пользователю различной вспомогательной информации используются служебные окна. К таким окнам относятся:

- окно сообщений,
- окно результата поиска во всех текстах,
- шаблоны текста,
- синтакс-помощник.

Первые два вида окон автоматически вызываются на экран в случае необходимости (шаблоны текста и синтаксис-помощника вызываются пользователем).

Назначение каждого из перечисленных окон описано в соответствующих разделах данного Руководства, а в настоящем разделе будут изложены общие приемы работы с такими окнами.

Для управления окнами используются контекстные меню, вызываемые в заголовке окна и внутри окна. С помощью первых производится управление поведением самого окна (подробнее см. [здесь](#)), а с помощью вторых – действия с содержимым окна.

Содержимое окна сообщений автоматически не очищается. Для его очистки следует выполнить команду [Очистить](#) контекстного меню.

33.1.5. Заккрытие окон

Любое окно можно закрыть с помощью мыши (нажатия кнопки закрытия окна), а также с помощью клавиатуры, используя комбинацию клавиш [Shift + Esc](#) или [Ctrl + F4](#).

33.1.6. История переходов между окнами

История перемещения активности между окнами запоминается в течение сеанса. Данный сервис можно использовать, например, для возврата к исходному окну при переходе к месту определения переменных, процедур и функций.

Для перемещения по истории переходов назад нужно выбрать пункт [Окна – Переместиться назад](#). Для перемещения по истории переходов вперед – выбрать пункт [Окна – Переместиться вперед](#).

33.2. Настройка параметров конфигуратора

Для установки различных параметров работы конфигуратора используется функция [Сервис – Параметры](#).

Окно [Параметры](#) организовано в виде картотеки. Все параметры, которые можно установить, объединены в группы. Для доступа к управляющим элементам конкретной группы нужно щелкнуть мышью на соответствующей закладке окна.

33.2.1. Общие

Данная закладка содержит единственный параметр, который определяет режим работы и интерфейс конфигуратора – [Редактирование конфигурации для режимов запуска](#).

Параметр может принимать два значения:

- [Управляемое приложение](#) – система скрывает элементы интерфейса, позволяющие редактировать свойства и объекты, имеющие смысл только для обычного режима запуска;
- [Управляемое приложение и обычное приложение](#) – не выполняется скрывание элементов интерфейса и имеется возможность одновременно редактировать конфигурацию и в обычном, и в управляемом режиме.

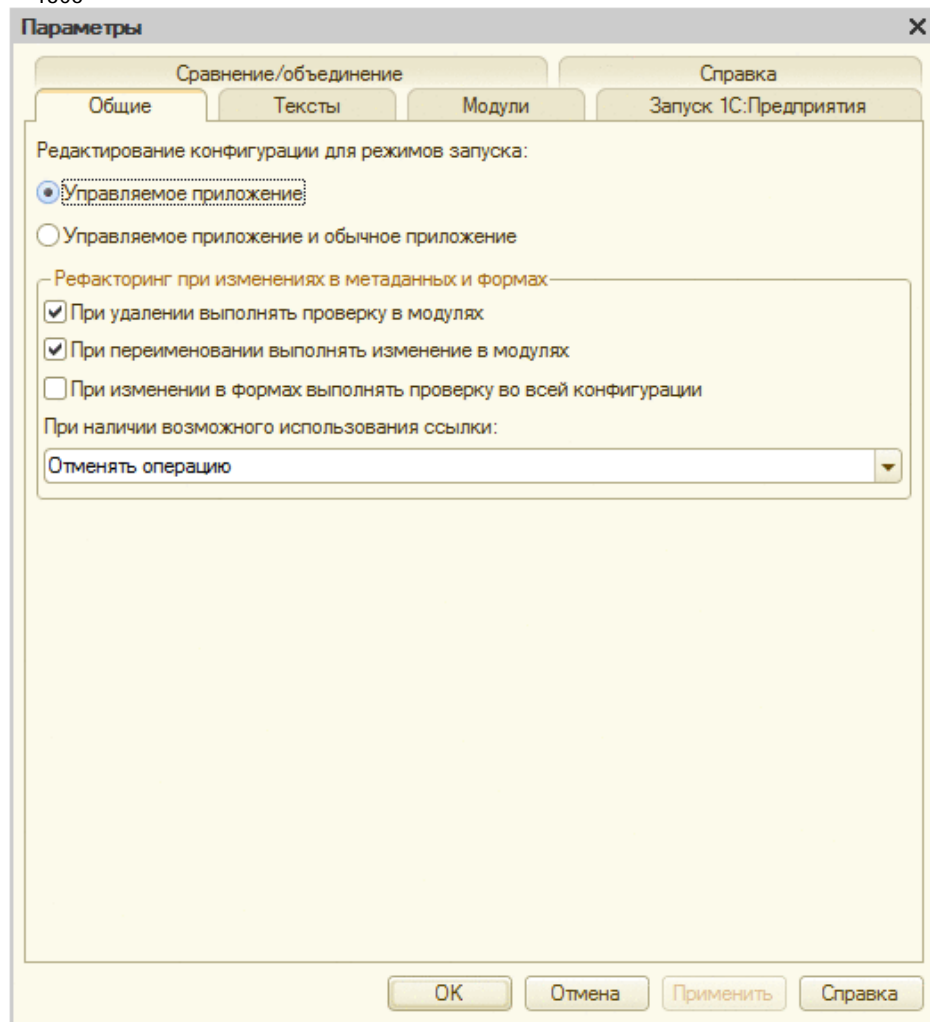


Рис. 537. Общие параметры

Данный параметр хранится в разрезе информационной базы, что позволяет настраивать его независимо для каждой информационной базы. По умолчанию параметр выставляется в режим редактирования для управляемого режима, однако при загрузке конфигурации, у которой свойство **Основной режим запуска** установлено в значение **Обычный** (или это свойство меняется интерактивно), параметр принимает значение редактирования управляемого и обычного режима запуска.

Ниже перечислены элементы интерфейса конфигуратора, которые скрываются при переключении параметра на управляемый режим запуска:

- ветка дерева метаданных **Стили**;
- ветка дерева метаданных **Интерфейсы**;
- свойство конфигурации **Основной интерфейс**;
- выбор основного интерфейса в диалоге редактирования пользователя;
- закладка **Форма** в параметрах конфигуратора;
- выпадающий список **Основной режим запуска** на закладке **Модули – Проверка**;
- свойство конфигурации **Использовать управляемые формы в обычном приложении**;
- свойство конфигурации **Использовать обычные формы в управляемом приложении**;
- закладка **Интерфейсы** в окнах редактирования метаданных;
- пункт контекстного меню конфигурации **Открыть модуль обычного приложения**;
- признак создания управляемой формы в конструкторе форм;
- команды **Конструктор выходных форм**;

- свойство общего модуля [Клиент \(обычное приложение\)](#);
- закладка [Дополнительные](#) на закладке [Формы](#) окна редактирования объекта конфигурации;
- обработчики событий [Обработка интерактивной активации](#) и [Перед интерактивным выполнением](#) у точки маршрута бизнес-процесса типа [Действие](#).

Группа параметров [Рефакторинг при изменениях в метаданных и формах](#) позволяет настроить, как система будет реагировать при выполнении таких изменений:

- [При удалении выполнять проверку в модулях](#) – при установленном флажке система будет выполнять поиск случаев использования удаляемого объекта. При обнаружении случаев использования (или подозрений на такое использование) система запросит дополнительное подтверждение удаления и выведет на экран результаты поиска.
- [При переименовании выполнять изменение в модулях](#) – при установленном флажке система будет выполнять поиск случаев использования старого имени объекта до переименования объекта конфигурации, и выполнять изменения в программном коде после такового.
- [При изменении в формах выполнять проверку по всей конфигурации](#) – установленный флажок предписывает системе выполнять проверку использования старого имени реквизита, элемента или параметры управляемой формы при переименовании или удалении. При этом проверка в модуле формы выполняется в любом случае, безотносительно состояния данного флажка.

Параметр [При наличии возможного использования ссылок](#) позволяет настроить поведение рефакторинга при невозможности однозначно определить использование удаляемого или переименоваемого объекта:

- [Задавать вопрос](#) – Для каждого случая возможного использования ссылки будет задаваться вопрос.
- [Продолжать операцию](#) – будет выполнено требуемое действие и замена для случаев фактического использования. Случаи возможного использования будут отображены на экране.
- [Отменять операцию](#) – в этом случае действие выполнено не будет, а результаты поиска использования ссылки на объект будут выведены на экран.

33.2.2. Тексты

Управляющие элементы закладки [Тексты](#) позволяют настроить поведение редактора текстовых документов.

[Перетаскивание текста](#). В редакторе текстов для перемещения и копирования блоков текста можно использовать режим [drag&drop](#) (перенеси и оставь).

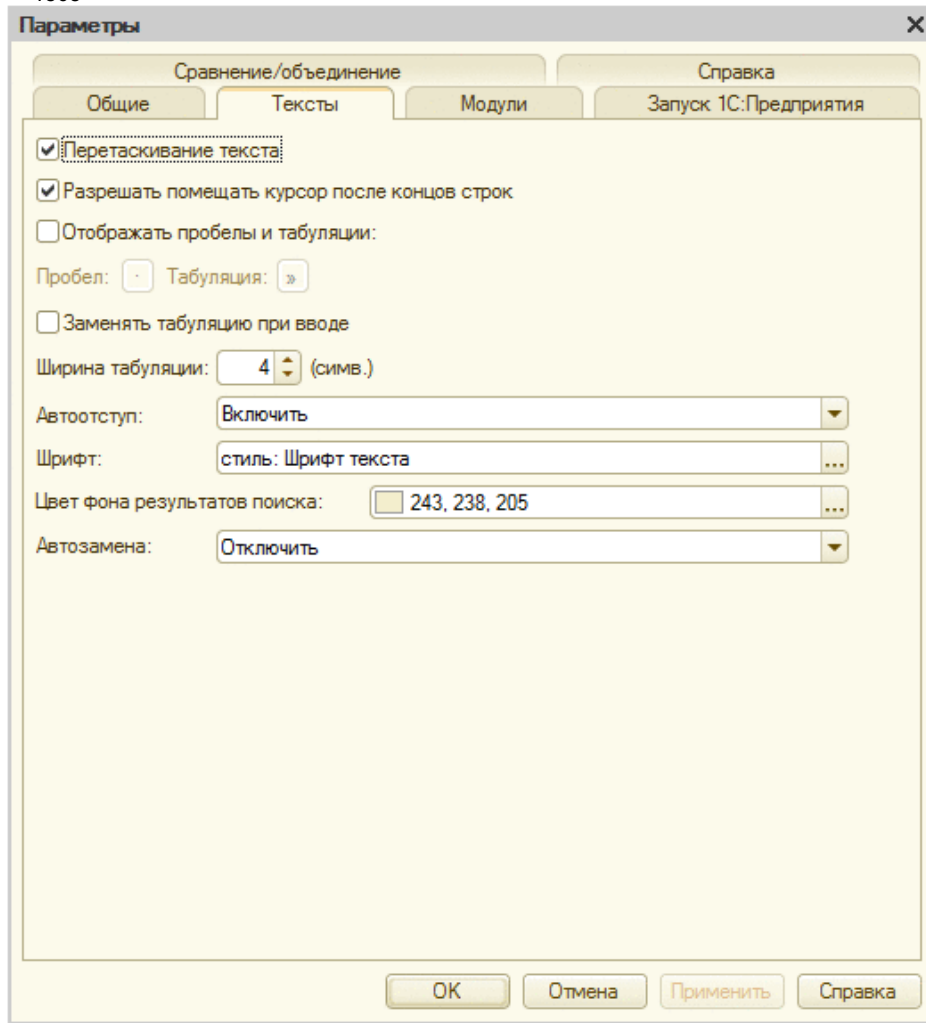


Рис. 538. Параметры редактирования текстов

Разрешить помещать курсор после концов строк. Если установлено, то курсор можно помещать после концов строк. В противном случае положение курсора ограничивается положением символа **Перевод строки**.

Отображать пробелы и табуляцию. Если установлено, то символы пробелов и табуляции будут показываться в тексте. Этот режим устанавливается для проверки форматирования текста. Если установлено свойство **Отображать пробелы и табуляцию**, то становятся доступными свойства **Пробел** и **Табуляция** для ввода символов, которыми будет осуществляться показ пробелов и табуляции:

- **Пробел** – указывается символ показа пробелов;
- **Табуляция** – указывается символ показа табуляции.

Заменять табуляцию при вводе. Если установлено, то при вводе текста символ табуляции заменяется на установленное в свойстве **Ширина табуляции** число пробелов.

Ширина табуляции. Устанавливает соответствие символа табуляции числу символов при вводе текста.

Автоотступ. Устанавливает или отменяет автоматический отступ при вводе символа **Перевод строки** (нажатие клавиши **Enter**).

Шрифт. Выбор шрифта, используемого при вводе текста. Для ввода текста модулей рекомендуется выбирать моноширинные шрифты (например, **Courier New**).

Цвет фона результатов поиска позволяет указать, каким цветом будет выделяться фон результатов поиска в редакторе текстов.

Автозамена. Если свойство включено, то вводимый текст, если он совпадает с текстом, указанным в шаблоне, в реквизите **Автоматически заменять строку**, будет заменяться текстом шаблона. Если выбрано значение **Включать с подсказкой**, то при наборе заменяемого текста после небольшой паузы будет выведен текст шаблона. Если выбрано значение **Только подсказка**, то после небольшой паузы будет выводиться текст шаблона, но замена текста не будет производиться.

Для того чтобы вводимый текст заменялся текстом шаблона, нужно после набора строки нажать клавишу **Пробел** или **Enter**.

33.2.3. Модули

Управляющие элементы закладки [Тексты модулей](#) позволяют настроить выделение цветом синтаксических конструкций в модулях, автоотступ, задать шаг табуляции для модулей, настроить группировку. Настройки объединены в два раздела, и каждый размещен на отдельной закладке.

33.2.3.1. Настройка параметров редактирования

На закладке [Редактирование](#) определяется выделение цветом синтаксических конструкций в модулях, автоотступ и задается шаг табуляции для модулей.

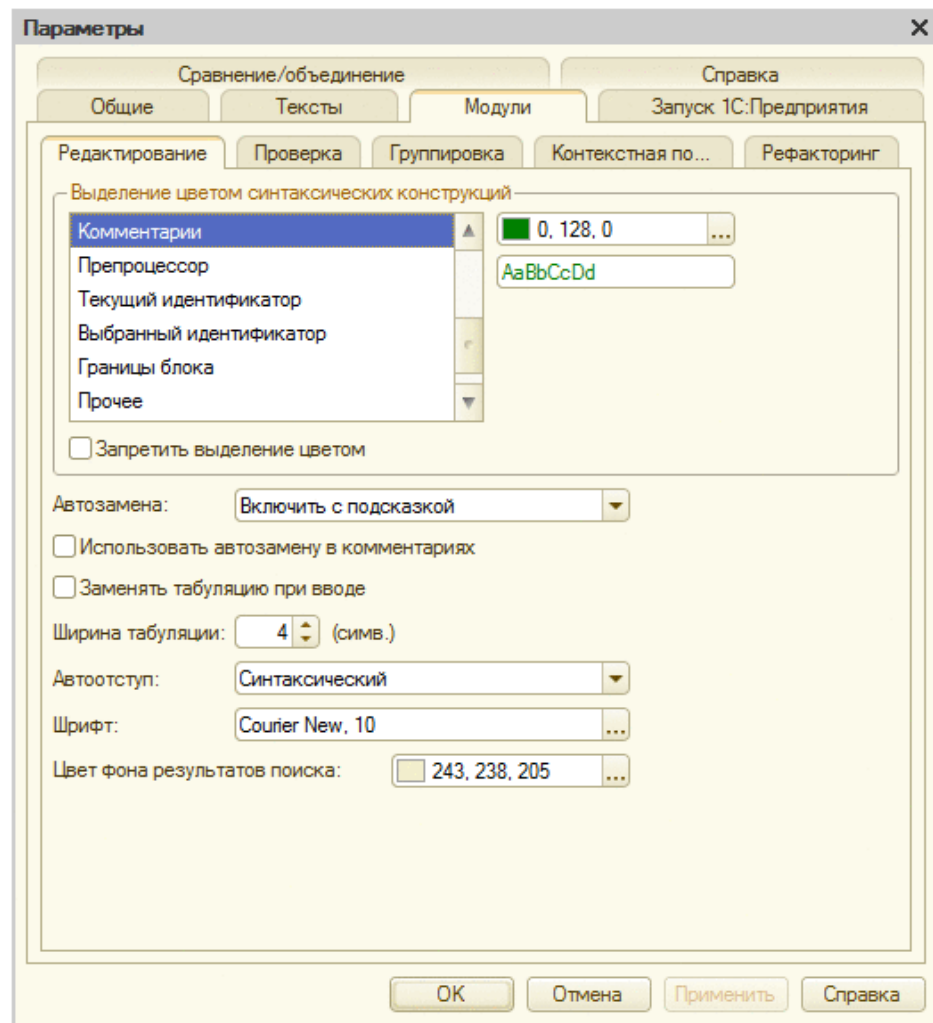


Рис. 539. Настройка редактирования текстов модулей

Выделение цветом синтаксических конструкций. Воспринимать текст модуля легче, если определенные синтаксические конструкции (константы, имена, операторы, комментарии и т. д.) будут выделены цветом. Цвет также выполняет дополнительную функцию проверки правильности при вводе текста модуля. В случае неправильного написания наименования оператора он не будет выделен нужным цветом. Для каждой конструкции (константы, имена, операторы, комментарии и т. д.) можно определить свой цвет.

Среди синтаксических конструкций, которые могут выделяться цветом, отдельно рассмотрим [Текущий идентификатор](#), [Выбранный идентификатор](#) и [Границы блока](#).

Цвет, указанный для конструкции [Текущий идентификатор](#), служит для того, чтобы выделять все использования любого идентификатора, «внутри» которого находится курсор редактора текстов. При этом для выделения не надо предпринимать никаких дополнительных действий. Достаточно просто указать на идентификатор курсором.

```

// Заголовок
Область = Макет.ПолучитьОбласть ("Заголовок");
ТабличныйДокумент.Вывести (Область);

// Шапка
Шапка = Макет.ПолучитьОбласть ("Шапка");
Шапка.Параметры.Заполнить (ЭтотОбъект);
ТабличныйДокумент.Вывести (Шапка);

// Товары
Область = Макет.ПолучитьОбласть ("ТоварыШапка");
ТабличныйДокумент.Вывести (Область);
ОбластьТовары = Макет.ПолучитьОбласть ("Товары");

```

Рис. 540. Текущий идентификатор

Цвет, указанный для конструкции [Выбранный идентификатор](#), служит для выделения всех использований идентификатора, который полностью выделен (двойным щелчком мыши или другим способом).

```

// Шапка
Шапка = Макет.ПолучитьОбласть ("Шапка");
Шапка.Параметры.Заполнить (ЭтотОбъект);
ТабличныйДокумент.Вывести (Шапка);

// Товары
Область = Макет.ПолучитьОбласть ("ТоварыШапка");
ТабличныйДокумент.Вывести (Область);
ОбластьТовары = Макет.ПолучитьОбласть ("Товары");

```

Рис. 541. Выделенный идентификатор

СОВЕТ. Если не требуется автоматически выделять все идентификаторы, «внутри» которых попадает курсор, то можно установить у конструкции [Текущий идентификатор](#) цвет, соответствующей конструкции [Идентификаторы](#). Тогда увидеть использование какого-либо идентификатора будет возможно, выделив его.

При выделении идентификаторов также будут выделяться строковые константы, которые удовлетворяют правилам формирования идентификаторов.

ПРИМЕЧАНИЕ. По умолчанию цвета, указанные для конструкций [Текущий идентификатор](#) и [Выделенный идентификатор](#), совпадают с цветом фона и требуют дополнительной настройки.

Цвет, указанный для конструкции [Границы блока](#), служит для выделения парных скобок (), (,], [] и ключевых слов, которые используются совместно, например [Процедура ... КонецПроцедуры](#), [Если ... Иначе ... КонецЕсли](#) и т. д. Выделение осуществляется в том случае, если курсор находится «внутри» ключевого слова или скобки.

[Запретить выделение цветом](#). Если установлено, то текст модуля будет показан как обычный текст, а настройка выделения цветом будет недоступна.

[Автозамена](#). Если свойство включено, то вводимый текст, если он совпадает с текстом, указанным в шаблоне, в реквизите [Автоматически заменять строку](#), будет заменяться на текст шаблона. Если значение свойства выбрано [Включить с подсказкой](#), то при наборе заменяемого текста после небольшой паузы будет выведен текст шаблона. Если значение свойства выбрано [Только подсказка](#), то при наборе заменяемого текста после небольшой паузы будет выведен текст подсказки.

Для того чтобы вводимый текст заменялся текстом шаблона, нужно после набора строки нажать клавишу [Пробел](#) или [Enter](#). Если значение свойства [Автозамена](#) выбрано [Отключить](#) или [Только подсказка](#), то данные действия не выполняются.

[Использовать автозамену в комментариях](#). Если установлено, то при вводе в комментариях текста, совпадающего с текстом, указанным в шаблоне, будет выведен текст шаблона.

[Заменять табуляцию при вводе](#). Если установлено, то при вводе текста символ табуляции заменяется на установленное в свойстве [Ширина табуляции](#) число пробелов.

[Ширина табуляции](#). Указывается эквивалентное число символов [Пробел](#) для одного символа табуляции.

[Автоотступ](#). Устанавливает или отменяет автоматический отступ при вводе символа [Перевод строки](#) (нажатие клавиши [Enter](#)). При выборе значения [Синтаксический](#) ввод текста модуля будет производиться с учетом текущей синтаксической конструкции.

Шрифт. Выбор шрифта, используемого при вводе текста. Для ввода текста модулей рекомендуется выбирать моноширинные шрифты (например, [Courier New](#)).

Цвет фона результатов поиска позволяет указать, каким цветом будет выделяться фон результатов поиска в редакторе текстов на встроенном языке.

33.2.3.2. Настройки параметров проверки модулей

На закладке [Проверка](#) производится установка режимов проверки модулей.

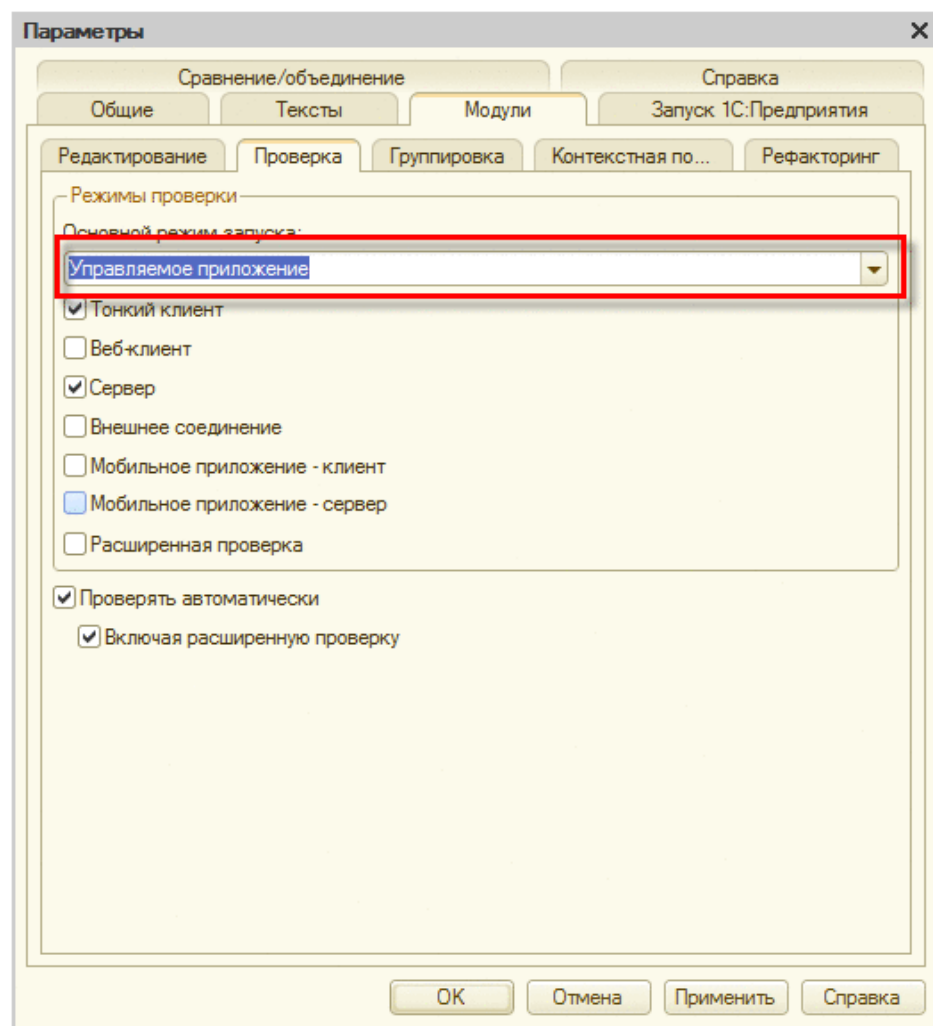


Рис. 542. Настройка синтаксической проверки модулей

С помощью настроек проверки модулей осуществляется подбор подходящей среды, в которой может быть осуществлена компиляция. Среда исполнения включает в себя общие модули с соответствующим признаком и специализированные модули, присутствующие в том или ином режиме исполнения (модуль управляемого приложения, модуль сеанса и т. д.). Если компиляция среды исполнения прошла успешно, происходит проверка модуля, для которого вызвана синтаксическая проверка. При этом выполняется столько проходов проверки (включая компиляцию среды исполнения), сколько флажков установлено в диалоге. Проверка не выполняется для тех модулей, которые не существуют в выбранном режиме проверки, например, модуль приложения не проверяется в случае выбора режима проверки [Тонкий клиент](#).

Наличие выпадающего списка [Основной режим запуска](#) в данной форме зависит от параметра [Редактирование конфигурации для режимов запуска](#) (см. [здесь](#)). Если значение параметра равно [Управляемое приложение](#), то выпадающий список отсутствует на форме, кроме того, в этом случае из диалога скрывается параметр [Толстый клиент \(обычное приложение\)](#). При открытии диалога, значение этого параметра всегда соответствует значению свойства конфигурации [Основной режим запуска](#).

Если выпадающий список [Основной режим запуска](#) присутствует на форме, то имеется возможность задать режимы проверки для каждого режима запуска. По умолчанию при выборе режима [Обычный](#) установлен только флажок [Толстый клиент \(обычное приложение\)](#), а при выборе режима [Управляемое приложение](#) – [Тонкий клиент](#) и [Сервер](#). При этом можно задать новый набор флажков для каждого режима запуска и нажать кнопку [Применить](#) или [ОК](#). При этом после изменения значения параметра [Редактирование конфигурации для режимов запуска](#) (см. [здесь](#)) конфигуратор восстановит необходимый набор режимов проверки.

Определены следующие инструкции препроцессора в зависимости от выбранного режима проверки:

Режим проверки	Инструкция препроцессора
Толстый клиент (обычное приложение)	Клиент, НаКлиенте, ТолстыйКлиентОбычноеПриложение
Тонкий клиент	ТонкийКлиент, Клиент, НаКлиенте
Веб-клиент	ВебКлиент, Клиент, НаКлиенте
Сервер	Сервер, НаСервере
Внешнее соединение	ВнешнееСоединение

Если установлен флажок [Проверять автоматически](#), то текст модуля будет автоматически проверяться на наличие синтаксических ошибок при записи и закрытии. Установка флажка [Включая расширенную проверку](#) приводит к тому, что для ограниченного набора типов выполняется проверка обращений к методам и свойствам объектов «через точку». Также выполняется проверка корректности указания параметров, заданных строкой, для ограниченного набора методов. Подробнее о видах проверки см. [здесь](#).

33.2.3.3. Настройка параметров группировки

На закладке [Группировка](#) производится установка режимов группировки и сворачивания различных синтаксических конструкций. В таблице по строкам представлены виды конструкций.

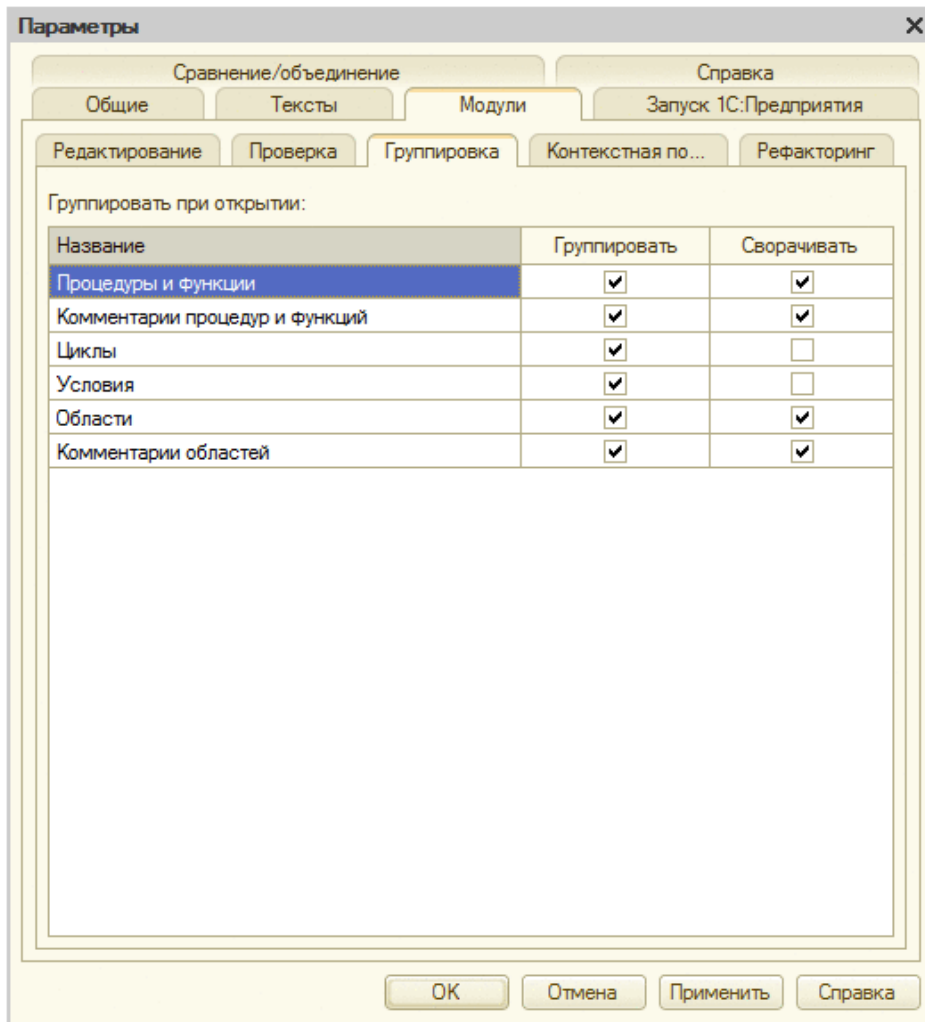


Рис. 543. Настройка группировки

В колонке [Группировать](#) установленный флажок означает, что данная синтаксическая конструкция будет автоматически сгруппирована. Следует учитывать, что группируются только синтаксические конструкции второго уровня вложенности.

В колонке [Сворачивать](#) установленный флажок означает, что данная синтаксическая конструкция будет автоматически свернута при открытии модуля.

33.2.3.4. Настройки контекстной подсказки

На закладке [Контекстная подсказка](#) производится настройка запуска контекстной подсказки при вводе текстов

модулей.

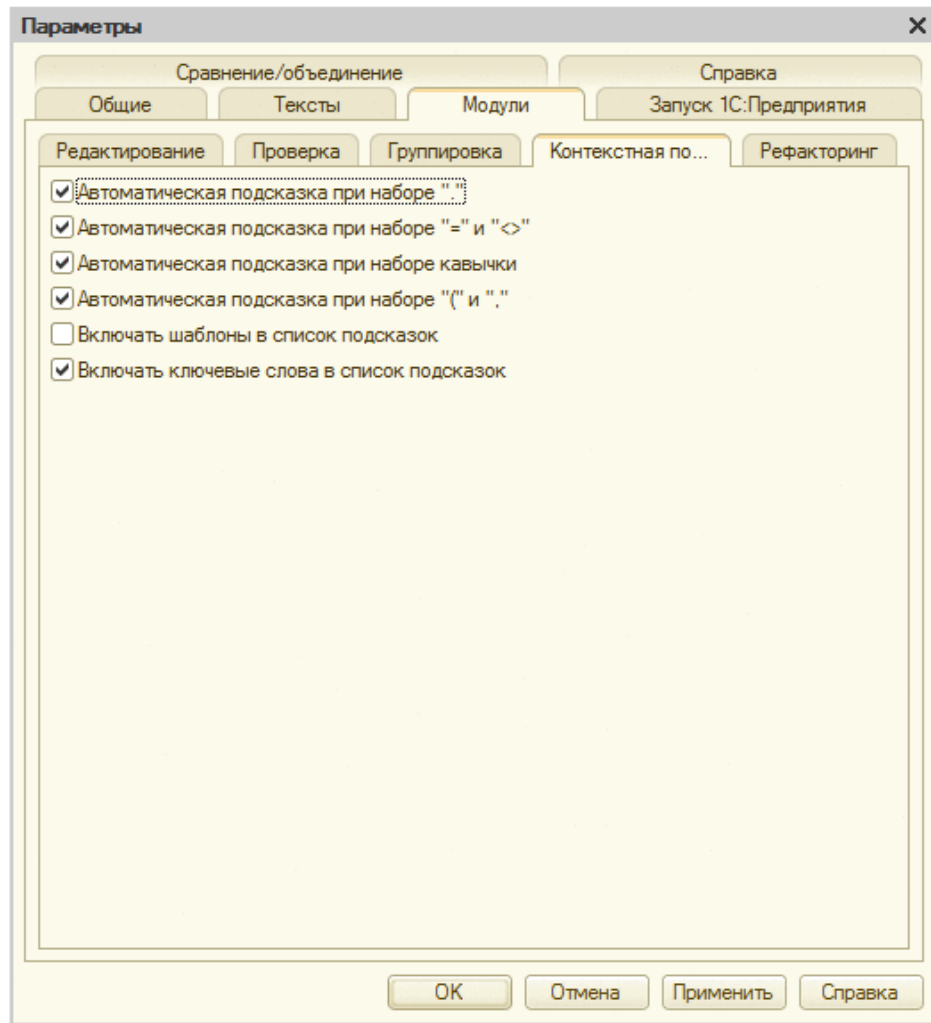


Рис. 544. Настройка контекстной подсказки

Если флажок [Автоматическая подсказка при наборе "."](#) установлен, то контекстная подсказка будет вызываться автоматически при наборе символа ".".

Если флажок [Автоматическая подсказка при наборе "=" и "<>"](#) установлен, то контекстная подсказка для выбора значений системных перечислений будет автоматически вызываться при наборе символа "=" и оператора "<>".

Флажок [Автоматическая подсказка при наборе кавычки](#) включает контекстную подсказку для параметров некоторых методов, если параметр задается строкой.

Флажок [Автоматическая подсказка при наборе "\(" или ","](#) включает контекстную подсказку параметров процедур и функций системы, включая процедуры и функции прикладного решения, комментарии к которым оформлены особым образом (см. [здесь](#)).

Если флажок [Включать шаблоны в список подсказок](#) установлен, то в список будут включаться те шаблоны, у которых определена строка автозамены.

Если флажок [Включать ключевые слова в список подсказок](#) установлен, то в список будут включены все ключевые слова (например, [Если](#), [Процедура](#), [Цикл](#), [Возврат](#) и т. д.).

33.2.3.5. Настройки рефакторинга

На закладке [Рефакторинг](#) производится настройка поведения системы при вызове поиска нерекондуемых синхронных вызовов (см. [здесь](#)).



Рис. 545. Настройка рефакторинга

Флажок **Преобразовывать модальные вызовы** определяет, будет или нет механизм рефакторинга обрабатывать вызовы модальных методов работы с диалогами.

Флажок **Преобразовывать синхронные вызовы расширений и внешних компонент** определяет, будет или нет механизм рефакторинга обрабатывать синхронные методы работы с файлами, расширениями и внешними компонентами.

33.2.4. Запуск системы «1С:Предприятие»

33.2.4.1. Основные параметры

На закладке выбирается запускаемое приложение, а также способ указания пользователя при запуске системы «1С:Предприятие».

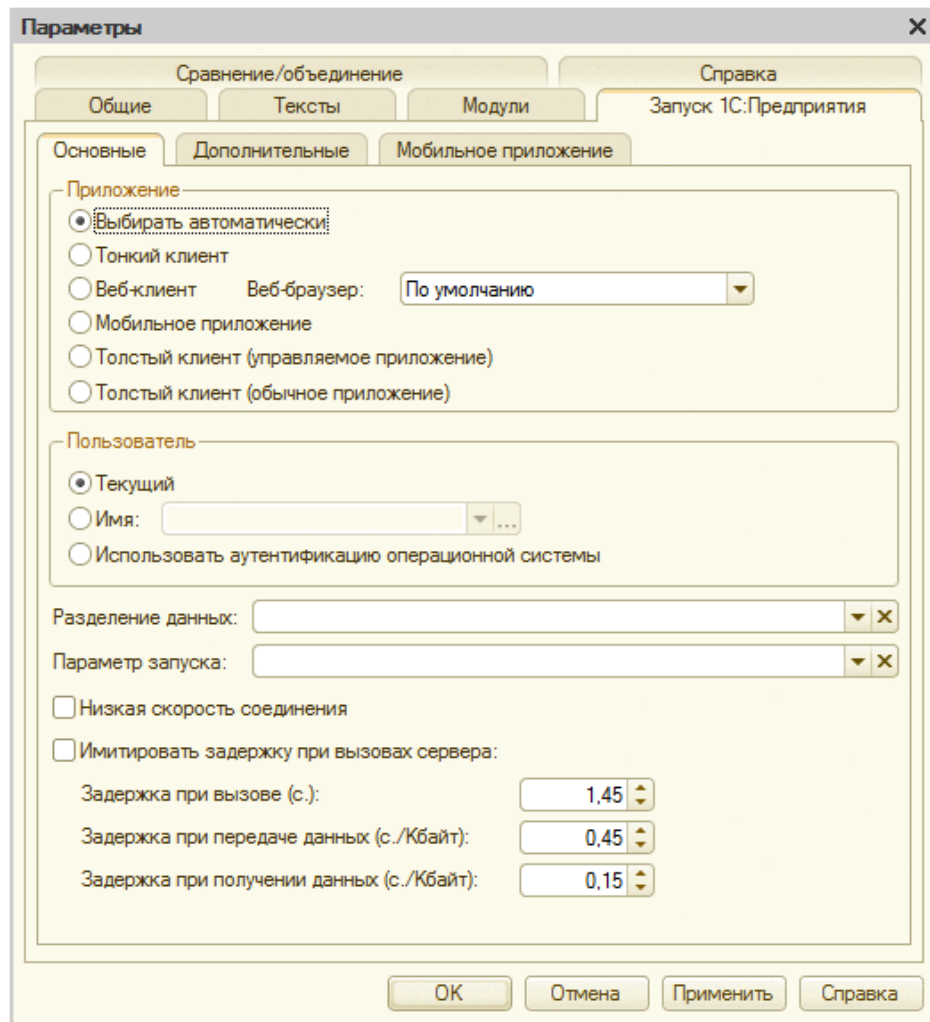


Рис. 546. Параметры запуска «1С:Предприятия»

В группе **Приложение** определяется, какой клиент будет запускаться при запуске режима 1С:Предприятие из конфигуратора.

Если режим установлен в **Выбирать автоматически**, то система будет выбирать вид клиента в зависимости от значения свойства конфигурации **Основной режим запуска** и свойства **Режим запуска** того пользователя, от имени которого будет выполняться запуск «1С:Предприятия». Если в свойствах конфигурации и в свойствах пользователя ничего специально не указано, то:

- если свойство **Основной режим запуска** установлено в значение **Обычный**, будет запущен толстый клиент в обычном режиме;
- если свойство **Основной режим запуска** установлено в значение **Управляемое приложение**, будет запущен тонкий клиент.

Если при запуске «1С:Предприятия» из конфигуратора нужно выбирать запуск конкретного вида клиента, то необходимо это делать вручную (пункты ниже **Выбирать автоматически**). В случае указания запуска веб-клиента имеется возможность указать, какой веб-браузер будет запущен:

- веб-браузер по умолчанию,
- Microsoft Internet Explorer,
- Mozilla Firefox,
- Google Chrome,
- Safari.

При запуске веб-клиента из конфигуратора происходит проверка версии платформы, для которой ранее была опубликована информационная база. Если версии отличаются, пользователю выдается соответствующее сообщение с предложением выполнить повторную публикацию. В случае утвердительного ответа инициируется открытие диалога публикации информационной базы (подробнее см. книгу «1С:Предприятие 8.3. Руководство администратора»).

В группе **Пользователь** определяется, от лица какого пользователя будет запущено «1С:Предприятие» при запуске из

- **Текущий** – из списка пользователей.
- **Имя** – указывается конкретный пользователь (аналог параметра командной строки `/N`). Имеется возможность также выбрать требуемого пользователя из списка. Для этого предназначена кнопка "..." в правой части поля.
- **Использовать аутентификацию операционной системы** – если установлен, то пользователь будет выбран с помощью аутентификации операционной системы (аналог параметра командной строки `/WA`).

Разделение данных – позволяет указать значения разделителей так, как они указываются для параметра `/Z` (подробнее см. [здесь](#)). Строка, указанная в этом поле, должна соответствовать правилу указания значений разделителей, но не должна содержать кавычек, обрамляющих значения при указании этих значений в параметре `/Z`. Например, при указании в командной строке следующего параметра: `/Z "-1,+4"` в поле **Разделение данных** следует указать `-1,+4`. Система запоминает различные варианты введенных значений разделителей и позволяет переключаться между ними.

Параметр запуска – параметр запуска (аналог параметра командной строки `/C`), который доступен для обработки через свойство глобального контекста `ПараметрЗапуска`.

Низкая скорость соединения – позволяет при запуске тонкого клиента устанавливать или отменять режим соединения с низкой скоростью. Значение флажка запоминается для конкретной информационной базы. Для информационной базы, для которой в списке баз не установлен режим запуска с низкой скоростью соединения, флажок отображается снятым. Если флажок установлен, то «1С:Предприятие» будет запущено с использованием низкой скорости.

Для информационной базы, для которой в списке баз установлен режим запуска с низкой скоростью соединения, флажок отображается серым. Если флажок снят, то «1С:Предприятие» будет запущено с использованием обычной скорости. Флажок является аналогом параметра командной строки `/O`.

Имитировать задержку при вызовах сервера – флажок включает имитацию работы тонкого клиента в условиях существенных временных задержек, возникающих при взаимодействии с сервером. Данный механизм по умолчанию выключен.

Задержка при вызове, с. – определяет величину задержки на каждый вызов сервера клиентским приложением.

Задержка при передаче данных на сервер (с./Кбайт) – определяет величину задержки при передаче данных на сервер. Время указывается в расчете на 100 байт передаваемых данных.

Задержка при получении данных с сервера (с./Кбайт) – определяет величину задержки при получении данных на сервер. Время указывается в расчете на 100 байт получаемых данных.

ПРИМЕЧАНИЕ. Значения, установленные по умолчанию, позволяют эмулировать GPRS-соединение.

Режим имитации задержек при вызове сервера можно включить также через командную строку: `/EmulateServerCallDelay [-CallXXXX] [-SendYYYY] [-ReceiveZZZZ]`, где:

- **EmulateServerCallDelay** – ключ командной строки, включающий механизм имитации задержек при вызове сервера;
- **Call** – параметр, указывающий величину задержки при вызове сервера в миллисекундах (если не указано, то задержка равна 100 мс);
- **Send** – параметр, указывающий величину задержки при передаче данных (если не указано, то задержка равна 100 мс);
- **Receive** – параметр, указывающий величину задержки при получении данных с сервера (если не указано, то задержка равна 100 мс).

33.2.4.2. Дополнительные параметры

На данной закладке собраны дополнительные параметры, которые могут помочь при разработке прикладных решений (в частности, при их отладке).

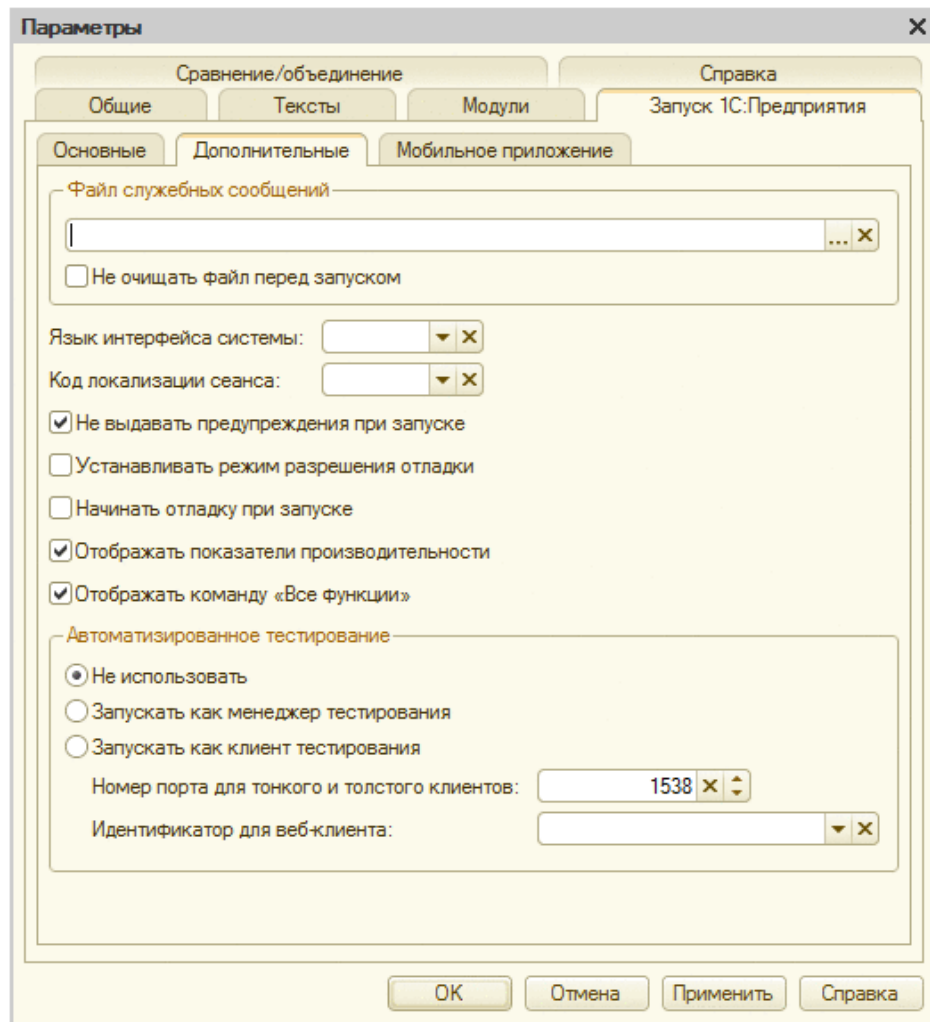


Рис. 547. Дополнительные параметры запуска «1С:Предприятия»

Файл служебных сообщений – если требуется автоматически сохранять служебные сообщения в файл, то в данном параметре указывается его имя (аналог параметра командной строки `/OUT`).

Не очищать файл перед запуском – если флажок установлен, то при следующем запуске сообщения будут добавляться в указанный файл (аналог параметра командной строки `/OUT` с ключом `-NoTruncate`).

Язык интерфейса – указывается национальный интерфейс (аналог параметра командной строки `/L`).

Код локализации сеанса – указывается код локализации сеанса (аналог параметра командной строки `/VL`).

Не выдавать предупреждения при запуске – если флажок установлен, то при запуске подавляются стартовые сообщения:

- Конфигурация базы данных не соответствует сохраненной конфигурации. Продолжить?
- Возможностей Вашего компьютера недостаточно для редактирования справки по конфигурации. Для редактирования справки необходимо установить Microsoft Internet Explorer версии 6.0 или выше.
- Возможностей Вашего компьютера недостаточно для редактирования html-документов, в том числе разделов справки. Для редактирования html-документов необходимо установить Microsoft Internet Explorer версии 6.0 или выше. В данном запуске редактирование html-документов будет недоступно (аналог параметру, передаваемому через ключ командной строки `/DisableStartupMessages`).

Установить режим разрешения отладки – если флажок установлен, то при запуске в режиме 1С:Предприятие возможно выполнение отладки (аналог параметра командной строки `/DEBUG`).

Начинать отладку при запуске – если флажок установлен, то при запуске системы «1С:Предприятие» подключение будет выполнено автоматически.

ПРИМЕЧАНИЕ. Если «1С:Предприятие» запускается из конфигуратора по нажатию кнопки **F5** (или вызова пункта меню **Отладка – Начать отладку**), то режим отладки включается автоматически (вне зависимости от состояния флажков **Установить режим разрешения отладки** и **Начинать отладку при запуске**).

Отображать показатели производительности – если флажок установлен, то при запуске «1С:Предприятия» на панели избранного и истории появляется специальная кнопка. При ее нажатии отображается окно, которое содержит информацию о текущей статистике взаимодействия с сервером «1С:Предприятия» (аналог параметра командной строки `/DisplayPerformance`).

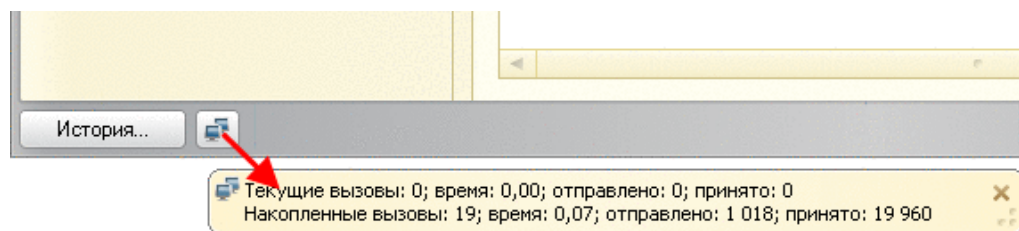


Рис. 548. Показывать вызовы сервера

Подробное описание механизма отображения вызовов сервера см. [здесь](#).

Отображать команду "Все функции" – управляет появлением пункта меню **Главное меню – Все функции** (см. [рис. 549](#)) и является аналогом параметра командной строки `/DisplayAllFunctions`. Вызов данной команды отображает форму, которая содержит перечень всех доступных пользователю прикладных объектов (используется в управляемом режиме толстого клиента, а также тонкого и веб-клиентах). По умолчанию команда меню отключена.

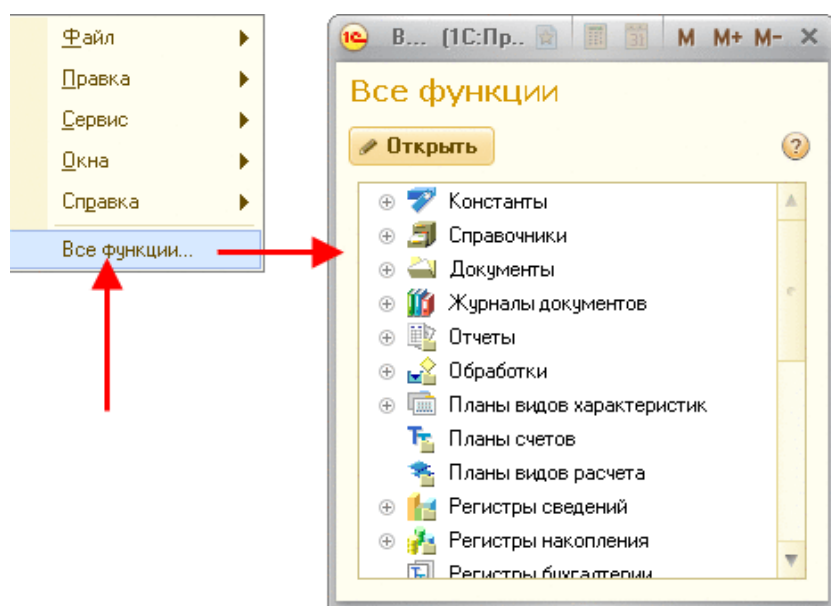


Рис. 549. Меню «Сервис – Все функции»

ПРИМЕЧАНИЕ. Формы, открываемые из окна **Все функции**, всегда открываются во вспомогательном окне приложения.

Также в окне **Все функции** доступны **Стандартные функции**, которые предназначены для выполнения некоторых стандартных процедур: просмотр списка активных пользователей, удаление помеченных объектов и т. д. Подробнее с описанием стандартных функций можно ознакомиться в книге «1С:Предприятие 8.3. Руководство администратора».

Группа **Автоматизированное тестирование** определяет, в какой роли (с точки зрения автоматизированного тестирования, подробнее см. [здесь](#)) будет запускать клиентское приложение при запуске из конфигуратора:

- **Не использовать** – запущенное клиентское приложение «не участвует» в автоматизированном тестировании;
- **Запускать как менеджер тестирования** – запускаемое клиентское приложение будет выступать в роли менеджера тестирования (будет использовать ключ командной строки `TestManager`);
- **Запускать как клиент тестирования** – запускаемое клиентское приложение будет выступать в роли клиента тестирования (будет использоваться ключ командной строки `TestClient`). Для взаимодействия с менеджером тестирования будет использоваться порт, заданный параметром **Номер порта для тонкого и толстого клиентов** (параметр `TPort` для ключа командной строки `TestClient`). Для веб-клиента будет использоваться идентификатор, указанным в параметре **Идентификатор для веб-клиента** (ключ командной строки `TestClientID`).

33.2.4.3. Мобильное приложение

На данной закладке указываются параметры, необходимые системе для работы с Android SDK.

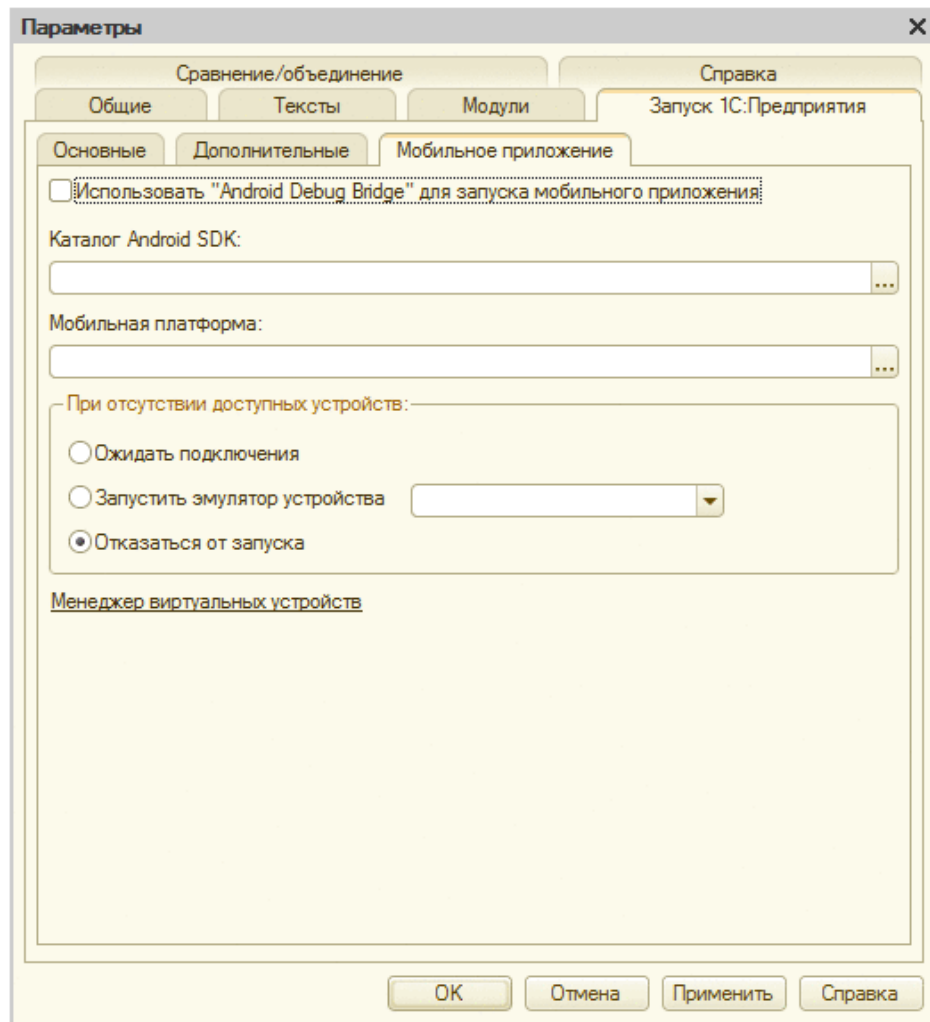


Рис. 550. Параметры запуска мобильного приложения

Флажок [Использовать "Android debug bridge"](#) для запуска мобильного приложения определяет способ передачи конфигурации на мобильное устройство:

- Флажок установлен: для передачи конфигурации на мобильное устройство используется [Android debug bridge \(adb\)](#).
- Флажок сброшен: используется схема с передачей конфигурации с помощью веб-сервера.

[Каталог Android SDK](#) – определяет путь к установленному [Android SDK](#). Данный параметр следует указать **до** выполнения любой операции с использованием [Android debug bridge](#).

[Мобильная платформа](#) – содержит ссылку на архив мобильной платформы, которая будет использоваться на мобильном устройстве (или эмуляторе). В архиве должен быть каталог [Android](#), в котором должны располагаться файлы [1cm-arm.apk](#) (мобильная платформа разработчика для архитектуры ARM) и [1cm-x86.apk](#) (мобильная платформа разработчика для архитектуры ARM).

Группа параметров [При отсутствии доступных устройств](#) определяет поведение конфигуратора в том случае, если к компьютеру, на котором работает конфигуратор, не подключено мобильное устройство или на этом компьютере не запущен эмулятор:

- [Ожидать подключения](#) – в этом случае конфигуратор будет ожидать подключения физического устройства или запуск эмулятора. Ожидание выполняется без тайм-аута.
- [Запустить эмулятор устройства](#) – в этом случае будет выполнена попытка запустить указанный, в соседнем поле, эмулятор устройства. Эмулятор создается с помощью менеджера виртуальных устройств [Android](#) (входит в состав [Android SDK](#)).
- [Отказаться от запуска](#) – в этом случае выполнение действия будет прервано.

Гиперссылка [Менеджер виртуальных устройств](#) предназначен для запуска приложения [AVD Manager](#) из каталога установки [Android SDK](#).

33.2.5. Сравнение/объединение

На закладке настраиваются параметры внешних программ, которые могут использоваться для операций сравнения/объединения модулей.

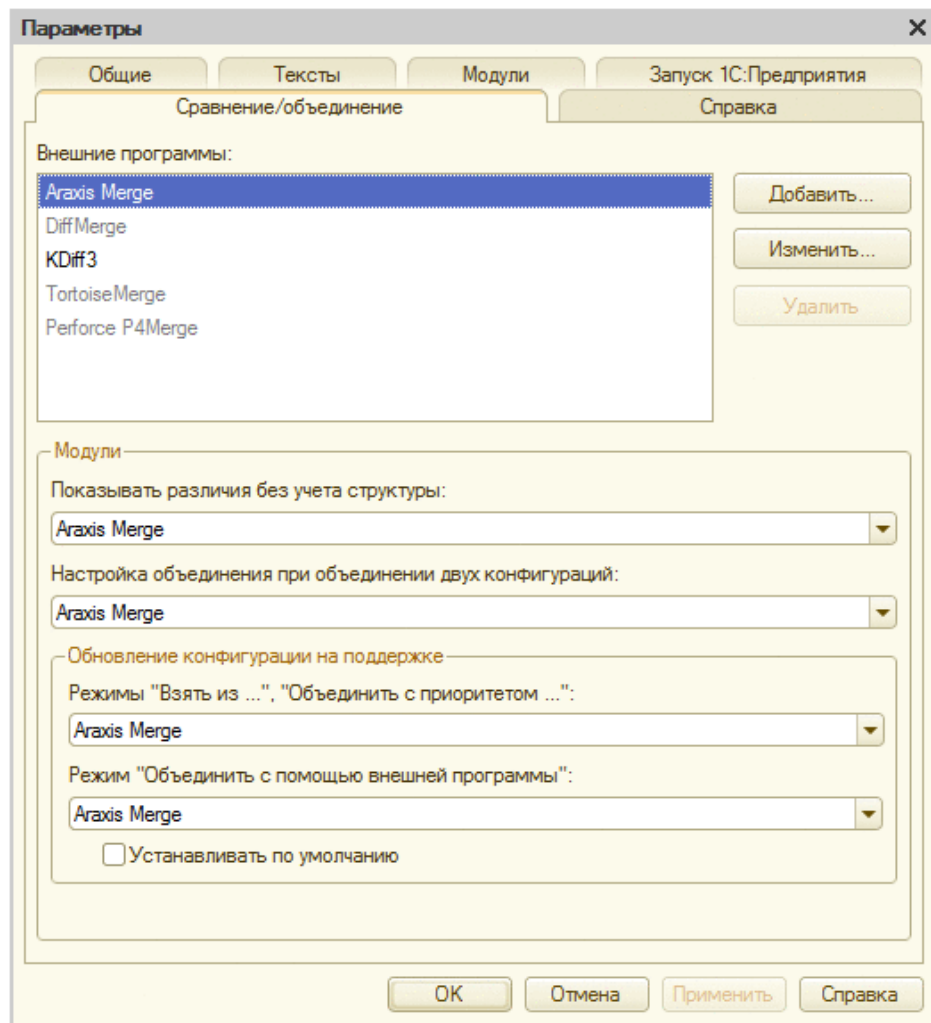


Рис. 551. Настройки внешних программ для операций сравнения/объединения

В списке **Внешние программы** перечислены те программы сравнения содержимого файлов, которые могут использоваться для операций сравнения/объединения (см. [здесь](#)).

По умолчанию в конфигураторе настроены параметры для следующих внешних программ:

- Araxis Merge
 - Используемая ОС: Windows;
 - Сайт: <http://www.araxis.com/merge/>;
 - Поддерживаемые режимы:
 - Сравнение двух файлов;
 - Объединение двух файлов;
 - Трехстороннее объединение.
- KDiff3:
 - Используемая ОС: Windows, Linux;
 - Сайт: <http://kdiff3.sourceforge.net/>;
 - Поддерживаемые режимы:
 - Сравнение двух файлов;
 - Объединение двух файлов;

- Трехстороннее объединение;
- Автоматическое трехстороннее объединение;
- TortoiseMerge:
 - Используемая ОС: Windows;
 - Сайт: <http://tortoisesvn.net/TortoiseMerge.html>;
 - Поддерживаемые режимы:
 - Сравнение двух файлов;
 - Трехстороннее объединение.
- Perforce P4Merge:
 - Используемая ОС: Windows, Linux;
 - Сайт: <http://www.perforce.com/product/components/perforce-visual-merge-and-diff-tools>;
 - Поддерживаемые режимы:
 - Сравнение двух файлов;
 - Трехстороннее объединение.
- DiffMerge:
 - Используемая ОС: Windows;
 - Сайт: <https://sourcegear.com/diffmerge/>;
 - Поддерживаемые режимы:
 - Сравнение двух файлов;
 - Трехстороннее объединение.

Однако для использования внешней программы сравнения требуется дополнительная настройка – необходимо указать путь к исполняемому файлу.

В группе **Модули** указывается, какой механизм будет использоваться для выполнения той или иной операции:

- **Показать различия без учета структуры** – выбирается механизм, который будет использоваться при выборе соответствующей команды в окне сравнения/объединения конфигураций. Выбор пункта **Стандартный способ** означает, что будет использоваться встроенный в конфигуратор механизм сравнения содержимого двух файлов. Также для выбора доступны внешние программы, для которых указаны свойства **Путь** и **Сравнение двух файлов**.
- **Настройка объединения при объединении двух конфигураций** – выбирается механизм, который будет использоваться при объединении двух конфигураций. Выбор пункта **Стандартный способ** означает, что будет использоваться встроенный в конфигуратор механизм объединения содержимого двух файлов. Также для выбора доступны внешние программы, для которых указаны свойства **Путь** и **Объединение двух файлов**.

Группа **Обновление конфигурации на поддержке** описывает поведение системы в случае, когда выполняется обновление конфигурации, которая находится на поддержке:

- **Режимы "Взять из ...", "Объединить с приоритетом..."** – указывает, что будет использоваться для интерактивного выполнения объединения. Выбор пункта **Стандартный способ** означает, что будет использоваться встроенный в конфигуратор механизм объединения (двустороннего). Также для выбора доступны внешние программы, у которых заполнены свойства **Путь** и **Трехстороннее объединение**.
- **Режим "Объединить с помощью внешней программы"** – указывает, с помощью какой внешней программы будет выполняться трехстороннее объединение и позволяет управлять доступностью режима **Объединить с помощью внешней программы** в настройках обновления конфигурации. Для выбора доступны внешние программы, у которых заполнены свойства **Путь** и **Трехстороннее объединение**.

Если в данном свойстве указано значение **Не используется**, то из настроек режима объединения пропадает возможность указания режима **Объединить с помощью внешней программы**. А при сохранении настроек

конфигуратора все режимы объединения, где было указано использование внешней программы, заменяются на режим [Взять из новой конфигурации поставщика](#) (если открыто окно обновления основной конфигурации).

Если в свойстве указана какая-либо внешняя программа и установлен флажок [Устанавливать по умолчанию](#), то при сохранении настроек устанавливается режим сравнения [Объединить с помощью внешней программы](#).

Существует возможность добавить собственную утилиту сравнения содержимого файлов. Для этого следует нажать кнопку [Добавить ...](#) и указать все параметры в открывшемся диалоге.

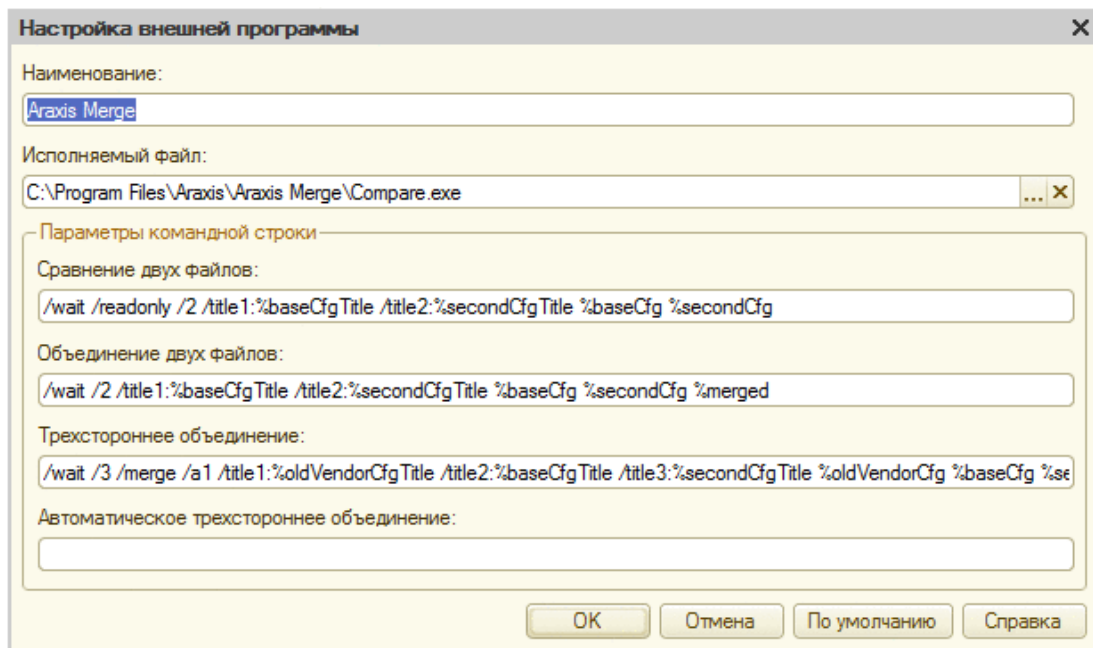


Рис. 552. Настройка внешней программы

Поле [Наименование](#) служит для указания представления программы сравнения.

Путь к утилите следует указать в поле [Исполняемый файл](#) (см. [рис. 551](#)). Если путь к исполняемому файлу не указан, то такая внешняя программа отображается в списке [Внешние программы](#) (см. [рис. 551](#)) серым цветом и эти программы нельзя выбирать в других настройках сравнения и объединения. На [рис. 551](#) серым цветом отмечены DiifMerge, TortoiseMerge и Perforce P4Merge. Для программ Araxis Merge и KDiff3 указан путь к исполняемому файлу.

В следующих полях (группа [Командная строка](#)) задаются параметры командной строки внешней программы, которые используются при выполнении той или иной операции из конфигуратора:

- [Сравнение двух файлов](#) – выполняется операция интерактивного сравнения содержимого двух файлов;
- [Объединение двух файлов](#) – выполняется операция интерактивного объединения содержимого двух файлов (с возможностью модификации получаемого результата);
- [Трехстороннее объединение](#) – выполняется операция интерактивного объединения содержимого трех файлов (с возможностью модификации получаемого результата);
- [Автоматическое трехстороннее объединение](#) – автоматическое объединение содержимого трех файлов. Выполняется не интерактивно при условии отсутствия каких-либо конфликтов при выполнении объединения.

В каждое одноименно поле вводятся параметры командной строки, которые будут передаваться внешней программе для выполнения операции. Если операция не поддерживается утилитой – соответствующее поле настройки остается незаполненным.

При формировании командной строки можно использоваться следующие predefined параметры:

- `%baseCfg` – путь к файлу из основной конфигурации;
- `%secondCfg` – путь к файлу из второй конфигурации. Второй конфигурацией может быть:
 - Конфигурация из файла;
 - Конфигурация поставщика (новая конфигурация поставщика при обновлении);
 - Конфигурация из хранилища конфигурации;

- Конфигурация базы данных.
- `%oldVersionCfg` – путь к файлу из предыдущей конфигурации поставщика;
- `%merged` – путь к результирующему файлу (результату объединения);
- `%baseCfgTitle` – заголовок окна для файла из основной конфигурации;
- `%secondCfgTitle` – заголовок окна для файла из второй конфигурации;
- `%oldVendorCfgTitle` – заголовок окна для файла из предыдущей конфигурации поставщика.

33.2.6. Справка

На закладке производится настройка вида показа различной справочной информации.

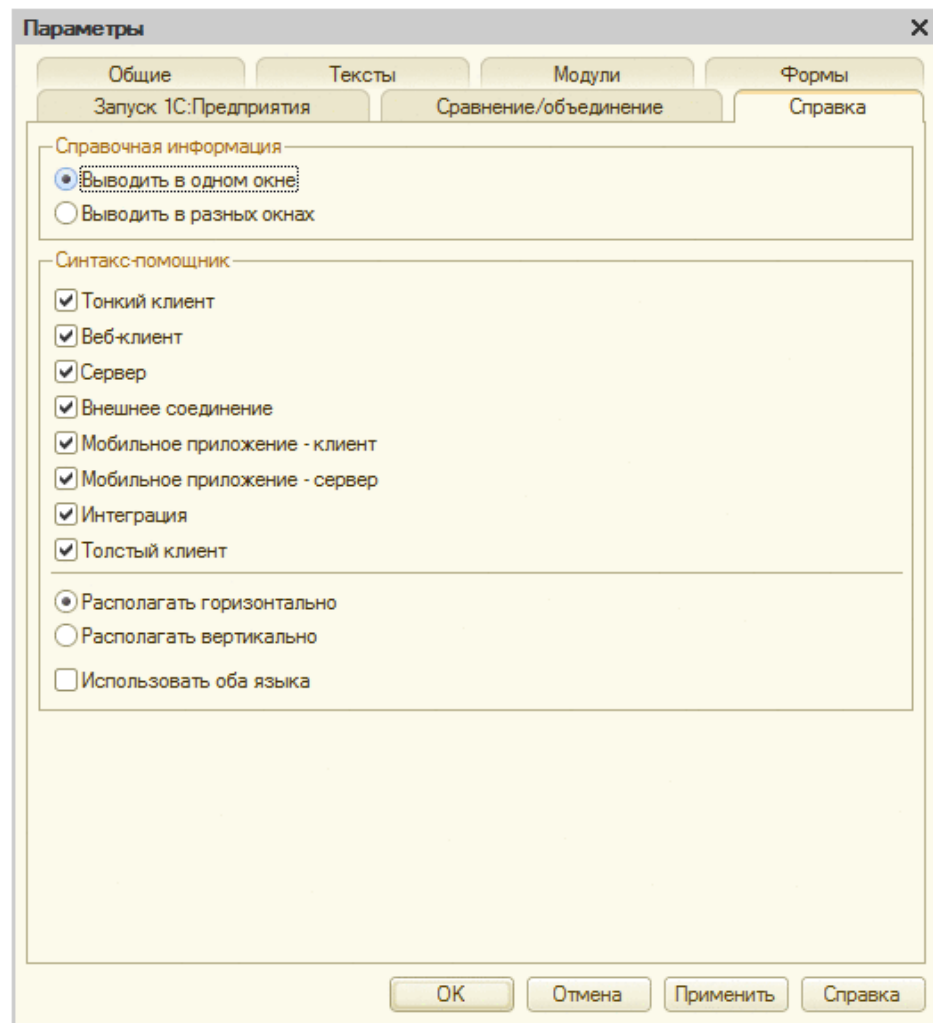


Рис. 553. Настройка отображения справочной информации

Справочная информация может быть показана в одном окне или же разделена по нескольким независимым окнам (параметры [Выводить в одном окне](#)/[Выводить в разных окнах](#)).

Синтакс-помощник может располагаться горизонтально или вертикально (параметры [Располагаться горизонтально](#)/[Располагаться вертикально](#)) и использовать оба языка для терминов (параметр [Использовать оба языка](#)).

Имеется возможность настроить фильтр отображения дерева содержания и индекса синтакс-помощника по доступности объектов для того или иного режима запуска «1С:Предприятия». Можно, например, отобразить разделы синтакс-помощника только по тем объектам, которые доступны в тонком клиенте.

На данную закладку также можно попасть с помощью кнопки командной панели синтакс-помощника.

33.3. Калькулятор

Описание работы с калькулятором см. в разделе «Калькулятор» документации «1С:Предприятие 8.3. Руководство

33.4. Календарь

Описание работы с календарем см. в разделе «Календарь» документации «1С:Предприятие 8.3. Руководство пользователя».

33.5. Шаблоны текста

Конфигуратор имеет возможность сохранения часто используемых фрагментов текста и быстрой вставки запомненных фрагментов в редактируемый текстовый документ или модуль.

Фрагменту текста ставится в соответствие условная комбинация символов – краткое имя фрагмента, называемое шаблоном. Если в параметрах конфигулятора включен режим автоподстановки, то нужный фрагмент текста будет автоматически вставляться в редактор при вводе шаблона.

ВНИМАНИЕ! Режим автоподстановки включается отдельно для модулей и текста на разных закладках окна настройки параметров конфигулятора.

33.5.1. Ведение списка шаблонов

Шаблоны хранятся в файлах. Можно выделить **стандартные шаблоны**, поставляемые вместе с «1С:Предприятием», и **пользовательские** шаблоны, формируемые пользователем. Стандартные шаблоны располагаются в каталоге **bin** каталога установки конкретной версии «1С:Предприятия». Пользовательские шаблоны могут располагаться в любом доступном месте жесткого диска. Состав подключенных шаблонов сохраняется в разрезе варианта встроенного языка – можно сформировать разный набор подключенных шаблонов для различных вариантов встроенного языка. Стандартные шаблоны поставляются для обоих вариантов встроенного языка.

Настройка шаблонов выполняется в окне [Шаблоны текста](#), в режиме, который можно вызвать из пункта [Сервис – Шаблоны текста](#).

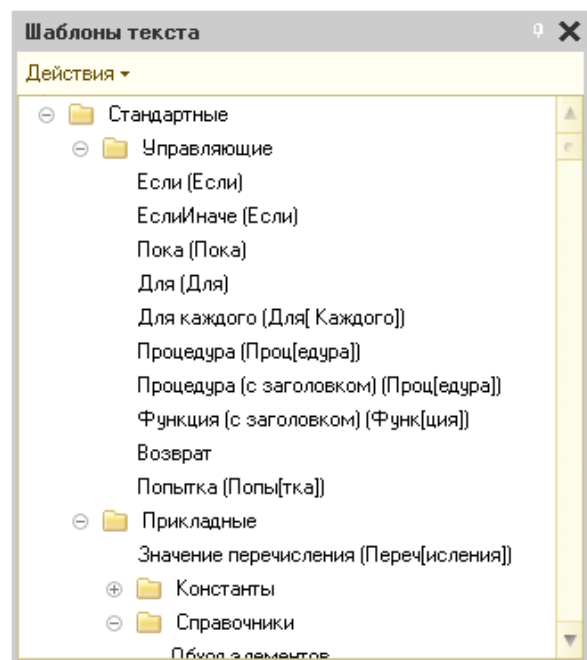


Рис. 554. Шаблоны текста

Для управления шаблонами нужно выбрать пункт [Действия – Настройка шаблонов](#) окна [Шаблоны текста](#). На экран выводится окно [Настройка шаблонов](#).

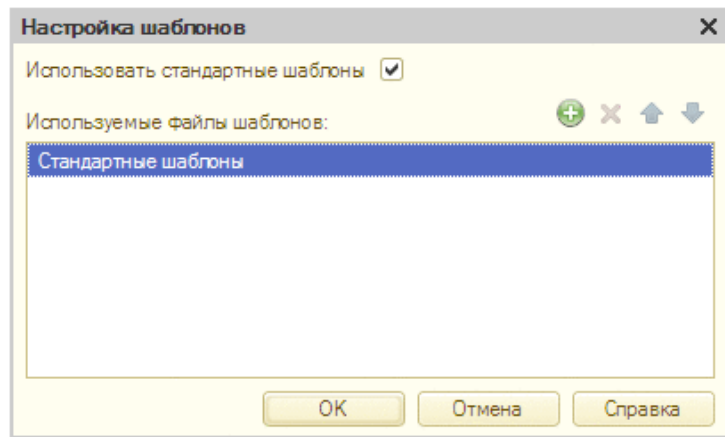


Рис. 555. Настройка шаблонов

Стандартные шаблоны добавляются с помощью флажка [Использовать стандартные шаблоны](#). Снятие этого флажка приводит к тому, что использование стандартных шаблонов становится невозможным. Каждая версия «1С:Предприятия» будет использовать свой набор стандартных шаблонов. Помимо стандартных файлов можно подключать пользовательские шаблоны. Для этого нужно нажать кнопку [Добавить](#) панели инструментов окна настройки шаблонов и выбрать необходимый файл шаблонов.

СОВЕТ. Рекомендуется хранить пользовательские шаблоны в каталоге, отличном от каталога установки «1С:Предприятия». Например, шаблоны можно хранить в специальном подкаталоге каталога [Мои документы](#).

Для упорядочивания файлов в списке следует использовать кнопки [Переместить вверх](#) и [Переместить вниз](#).

Чтобы удалить ненужный шаблон, следует выбрать его в списке файлов и нажать кнопку [Удалить](#).

33.5.2. Редактирование шаблона

Для редактирования существующего файла нужно выбрать пункт [Действия – Изменить](#). На экран выводится окно редактирования шаблонов, расположенных в файле.

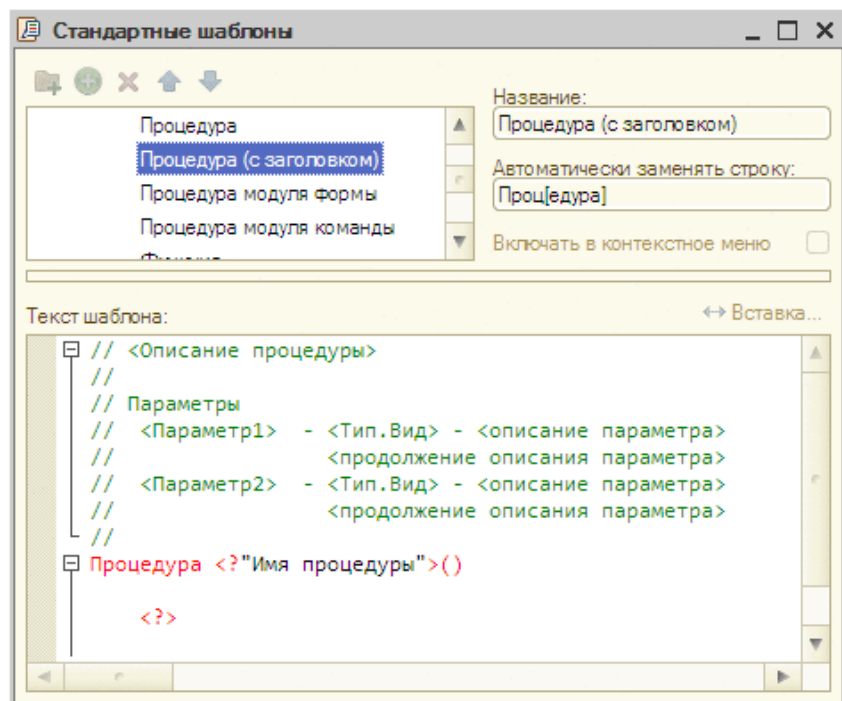


Рис. 556. Редактор шаблонов

ВНИМАНИЕ! Редактирование стандартных шаблонов невозможно. В редакторе шаблонов они доступны только для просмотра.

В реквизите [Название](#) указывается название шаблона, как он показывается в списке шаблонов. В реквизите [Автоматически заменять строку](#) указывается строковая последовательность, после набора которой конфигуратор позволяет заменить ее на шаблон. Замена производится при нажатии клавиши [Пробел](#) или [Enter](#).

Кроме того, после набора строки на экран выдается подсказка, содержащая текст шаблона, как показано на рисунке

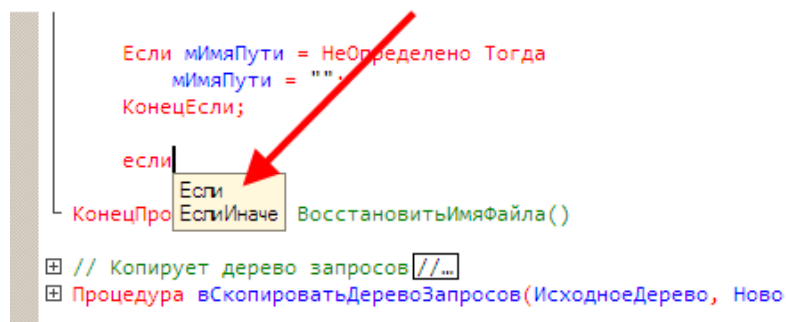


Рис. 557. Использование шаблонов

Окно редактирования шаблонов содержит панель инструментов, с помощью которой ведется создание групп шаблонов и отдельных шаблонов, удаление ненужных шаблонов и упорядочивание списка. В левой части окна расположено дерево шаблонов. В правой части производится редактирование наименования группы шаблонов и редактирование шаблона.

Для удобства использования шаблоны можно группировать по папкам. Чтобы создать группу, следует воспользоваться командой контекстного меню **Добавить папку** или нажать соответствующую кнопку панели инструментов. Ввести наименование группы в реквизите **Название**.

Для создания шаблона выбрать группу, в которой будет размещен шаблон, и воспользоваться командой контекстного меню **Добавить** или нажать соответствующую кнопку панели инструментов. В реквизите **Название** указать наименование шаблона (по этому наименованию производится выбор нужного шаблона при вставке в редактируемый текст). В поле **Автоматически заменять строку** нужно указать начальную часть строки, при написании которой в окне текстового документа программа произведет подстановку написанного текста текстом из шаблона (если включен режим автоподстановки). Автоподстановка производится после набора последовательности и нажатия клавиши **Enter** или клавиши **Пробел**. В многострочном поле **Текст шаблона** нужно ввести текст, которым должен быть заменен текст, указанный в поле **Автоматически заменять строку**, или который записывается в текст документа при выборе шаблона.

Чтобы шаблон был доступен при интерактивном выборе, нужно установить флажок **Включать в контекстное меню**.

Выбор шаблона для вставки в текст (модуль) осуществляется из контекстного меню, из списка шаблонов. Шаблоны размещаются в списке, если установлен флажок **Включать в контекстное меню**.

33.5.2.1. Управляющие конструкции шаблона

Произвольный запрос

Описание:

Используется для ввода произвольного текста. При использовании такого шаблона на экран выводится диалог на ввод контекстно зависимой части шаблона.

Синтаксис:

<? "Подсказка" >

Параметры:

<Подсказка>

Текст поясняющей надписи.

Установка курсора

Описание:

Конструкция используется для установки курсора в указанное место после вставки текста шаблона.

Синтаксис:

<?>

Специальный запрос

Описание:

Конструкция используется для ввода объектов конфигурации, предопределенных элементов и других данных.

Синтаксис:

<?"Подсказка", <Ключевое слово>[, <Параметр1>[, <Параметр2>...]]

Параметры:

<Подсказка>

Поясняющий текст в запросе на ввод.

<Ключевое слово>

Вид запроса.

<Параметр1>...

Параметры запроса.

Описания специальных запросов приведены в следующей таблице:

| Русский вариант | Английский вариант | Описание |
|--|--|---|
| БизнесПроцесс | BusinessProcess | Выбор вида бизнес-процесса |
| ВариантВыбора <Подсказка 1>, <Строка для вставки 1>, ... , <Подсказка N>, <Строка для вставки N> | VariantChoice <Подсказка 1>, <Строка для вставки 1>, ... , <Подсказка N>, <Строка для вставки N> | <p>При использовании данного шаблона на экран выводится список строк для выбора.</p> <p>Каждая строка списка состоит из подсказки и текста вставки.</p> <p>Например:</p> <p>«Подсказка 1» – параметр, поясняющий текст первой строки списка;</p> <p>«Строка для вставки 1» – параметр, текст, который будет вставлен при выборе первой строки</p> |
| ВыборТипа | TypeChoice | Выбор типа |
| Документ | Document | Выбор вида документа |
| ЖурналДокументов | DocumentJournal | Выбор вида журнала документов |
| Задача | Task | Выбор вида задачи |
| ЗначениеПеречисления | EnumValue | Выбор значения перечисления |
| ИмяПользователя | UserName | Ввод имени пользователя |
| КонструкторОписанияТипов | TypeDescriptionConstructor | Конструктор описания типов |
| | | |

| | | |
|--|--|---|
| ПолноеИмяПользователя | UserFullName | Ввод полного имени пользователя |
| ИмяПользователяХранилищаКонфигурации | ConfigurationStorageUserName | Ввод полного имени пользователя хранилища конфигурации |
| Константа | Constant | Выбор константы |
| КритерийОтбора | FilterCriterion | Выбор критерия отбора |
| Обработка | DataProcessor | Выбор обработки |
| Отчет | Report | Выбор отчета |
| Перерасчет | Recalculation | Выбор перерасчета |
| Перечисление | Enum | Выбор перечисления |
| ПланВидовРасчета | ChartOfCalculationTypes | Выбор плана видов расчета |
| ПланВидовХарактеристик | ChartOfCharacteristicTypes | Выбор плана видов характеристик |
| ПланОбмена | ExchangePlan | Выбор плана обмена |
| ПланСчетов | ChartOfAccounts | Выбор плана счетов |
| ПланВидовРасчетаПредопределенныеДанные | ChartOfCalculationTypesPredefinedData | Выбор предопределенных данных плана видов расчета. Сначала выбирается план видов расчета, а затем значение предопределенных данных |
| ПланВидовХарактеристикПредопределенныеДанные | ChartOfCharacteristicTypesPredefinedData | Выбор предопределенных данных плана видов характеристик. Сначала выбирается план видов характеристик, а затем значение предопределенных данных |
| ПланСчетовПредопределенныеДанные | ChartOfAccountsPredefinedData | Выбор предопределенных данных плана счетов. Сначала выбирается план счетов, а затем значение предопределенных данных |
| ОбъектМетаданных | MDOObjectsSubset | Выбор объектов метаданных. При формировании шаблона выводится диалог, в котором следует отметить виды объектов метаданных. При выборе шаблона на экран выводится окно выбора, в котором содержатся все объекты метаданных указанных видов |
| | | |

| | | |
|----------------------------------|------------------------------|--|
| Последовательность | Seq | Выбор последовательности |
| РегистрБухгалтерии | AccountingRegister | Выбор регистра бухгалтерии |
| РегистрНакопления | AccumulationRegister | Выбор регистра накопления |
| РегистрРасчета | CalculationRegister | Выбор регистра расчета |
| РегистрСведений | InformationRegister | Выбор регистра сведений |
| Справочник | Catalog | Выбор справочника |
| СправочникПредопределенныеДанные | CatalogPredefinedData | Выбор предопределенных данных справочника. Сначала выбирается справочник, а затем значение предопределенных данных |
| ТекстЗапроса | QueryText | Осуществляется ввод текста запроса с использованием конструктора запросов |
| ДатаВремя, <Форматная строка> | DateTime, <Форматная строка> | Ввод текущей даты в формате, указанном в параметре <Форматная строка> |
| ФорматнаяСтрока | FormatString | Осуществляется ввод текста форматной строки с использованием конструктора форматной строки |

33.5.2.2. Редактор текстов шаблонов

Структурно шаблон может состоять из статической (неизменяемой) и динамической части, содержание которой контекстно зависимо и не может быть задано наперед. Например, оператор **Если** встроенного языка имеет следующую конструкцию:

[Копировать в буфер обмена](#)

```
Если <?"Условие 1"> Тогда
ИначеЕсли <?"Условие 2"> Тогда
КонецЕсли;
<?"Условие...">
```

Это контекстно зависимые части шаблона.

Для универсальности использования шаблона применяется механизм вставки управляющей конструкции. В тексте шаблона размещают управляющие конструкции – последовательности символов, которые при вставке заменяющего текста шаблона вызывают выполнение каких-либо действий. Например, управляющие конструкции позволяют запрашивать у пользователя какую-либо информацию и помещать эту информацию во вставляемый текст.

Управляющие конструкции можно вставлять в шаблон вручную или использовать специальный запрос для интерактивной вставки управляющих конструкций.

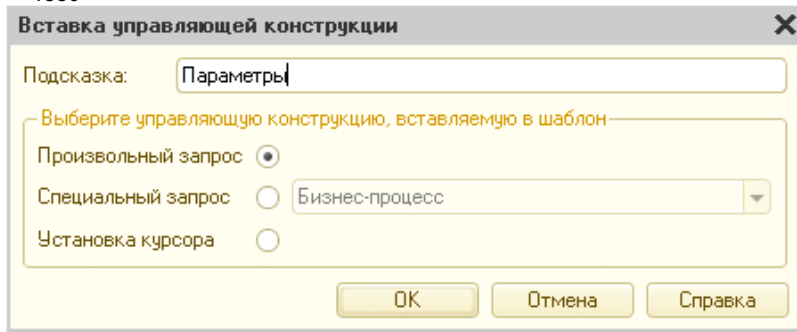


Рис. 558. Вставка управляющей конструкции

Нужно установить курсор в то место шаблона, которое содержит переменную часть, и нажать кнопку [<->Вставка...](#) На экран выводится диалог [Вставка управляющей конструкции](#).

В реквизите [Подсказка](#) указывается строка текста, которая выводится в виде пояснения к выполняемому действию при запросе на ввод контекстно зависимой части шаблона. Так как шаблон может содержать несколько управляющих конструкций, ввод которых осуществляется последовательно, этот текст поясняет суть выполняемых действий и место размещения переменной части.

Группа переключателей позволяет выбрать вид управляющей конструкции, вставляемой в шаблон.

Произвольный запрос используется для ввода произвольного текста. В тексте шаблона появится управляющая конструкция вида `<?"Текст подсказки">`.

Управляющая конструкция типа [Произвольный запрос](#) используется следующим образом. Если в заменяющем тексте встречается конструкция `<?"Текст подсказки">`, то перед вставкой заменяющего текста на экран будет выдан запрос с текстом подсказки и полем для ввода текста. В поле ввода можно указать текст, который будет вставлен в заменяющий текст вместо конструкции `<?"Текст подсказки">`. Конструкций типа [Произвольный запрос](#) в заменяющем тексте может быть несколько, причем несколько конструкций могут иметь одинаковый текст подсказки. При вставке заменяющего текста на каждую из различающихся конструкций будет выдан свой запрос для ввода текста. На каждый набор одинаковых конструкций будет выдан один общий запрос, а введенный в запрос текст заменит все одинаковые конструкции.

[Специальный запрос](#) используется для выбора объектов конфигурации. Для того чтобы поместить в шаблон управляющей конструкции выбор типа объекта конфигурации с помощью окна [Вставка управляющей конструкции](#), нужно в поле [Подсказка](#) поместить имя запроса и в списке выбрать конструкцию (выбор объекта конфигурации, предопределенного элемента объекта конфигурации или специальной конструкции), например [Справочник](#).

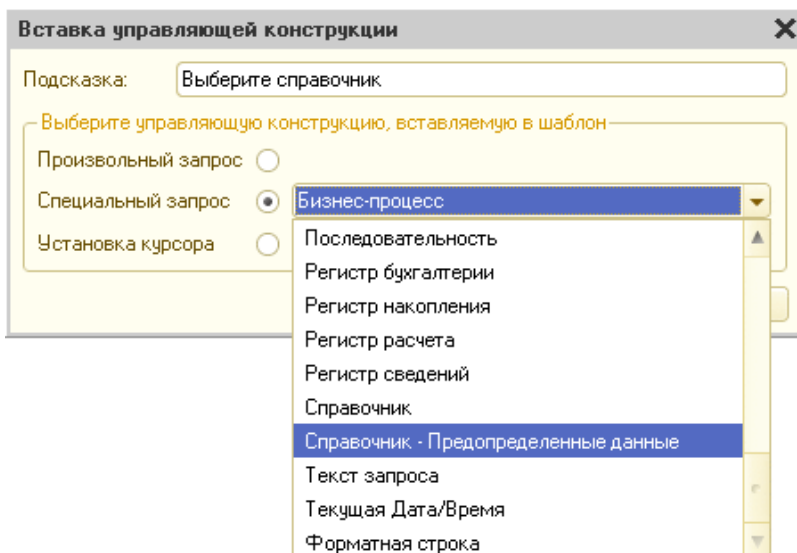


Рис. 559. Специальный запрос

При использовании такого шаблона на экран выводится список объектов конфигурации для выбора. Наименование выбранного объекта подставляется вместо набранного текста. Позиция выбранной строки списка запоминается, и при следующем выборе данного шаблона список будет позиционирован на эту строку.

В числе возможных значений специального запроса дополнительно опишем выбор варианта. При формировании шаблона с выбором данного типа запроса после нажатия [ОК](#) на экран выводится окно формирования списка возможных значений.

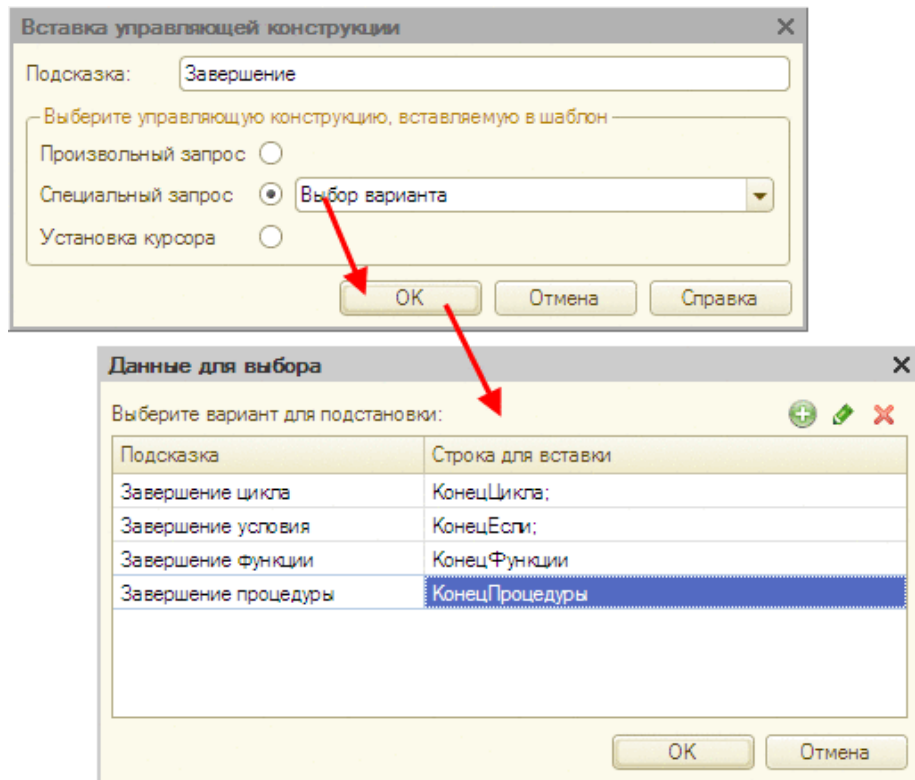


Рис. 560. Данные для выбора

В колонке **Подсказка** вводится текст, который появляется в списке выбора при использовании данного шаблона. В колонке **Строка вставки** вводится текст, который размещается в текстовом документе.

Редактировать список можно только в самом тексте шаблона. При повторном вызове вставки конструкции будет добавлена новая конструкция.

После нажатия **OK** в текст шаблона будет помещена управляющая конструкция.

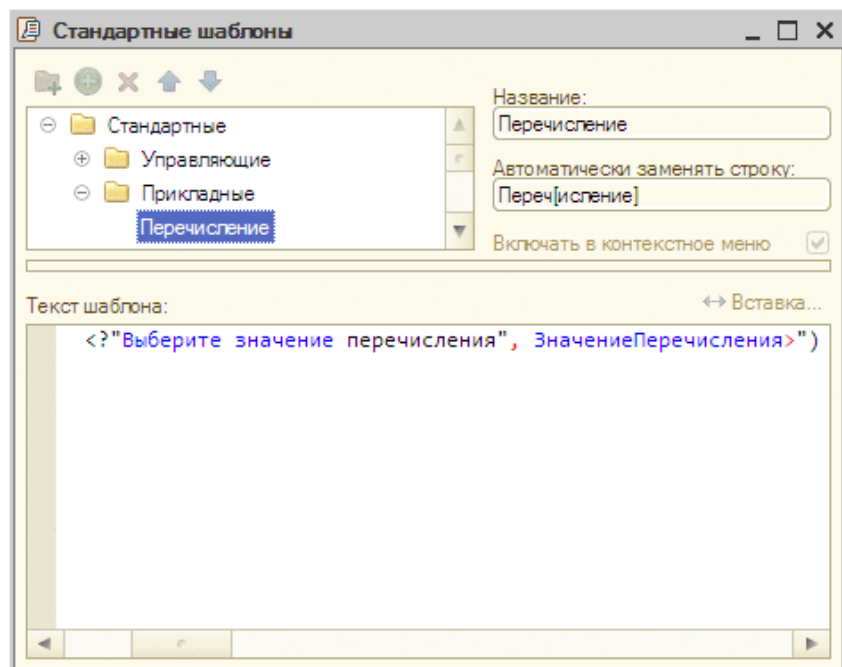


Рис. 561. Шаблон с управляющей конструкцией

В дереве шаблонов (или при автоподстановке) выберем данный шаблон.

```

////////////////////////////////////
// ПЕРЕМЕННЫЕ МОДУЛЯ
//
Перем ДрайверСканераШтрихкодов Экспорт; // Сканер штрихкодов

```

Процедура ПередНачаломРаботыСистемы(Отказ)

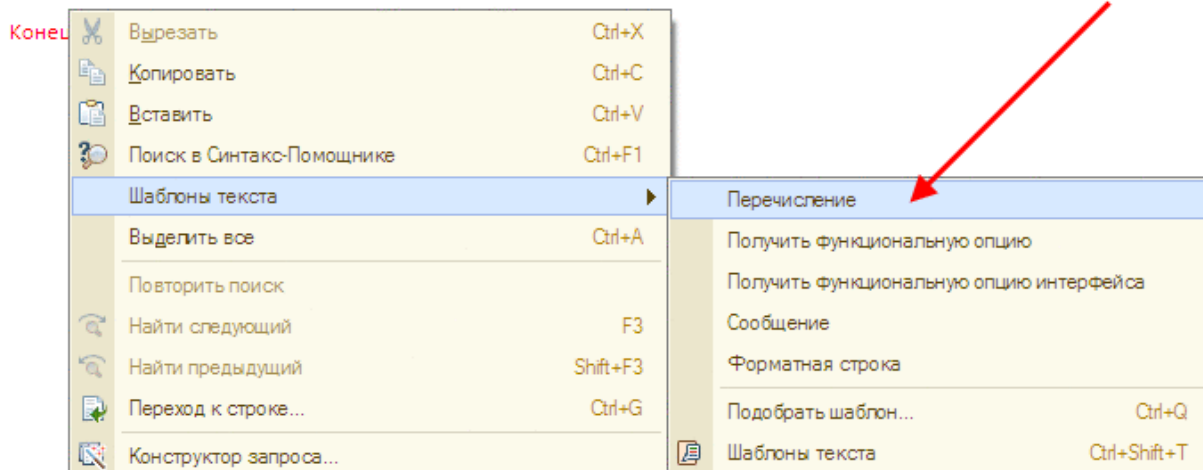


Рис. 562. Контекстное меню

В этом случае на экран выводится запрос выбора вида справочника.

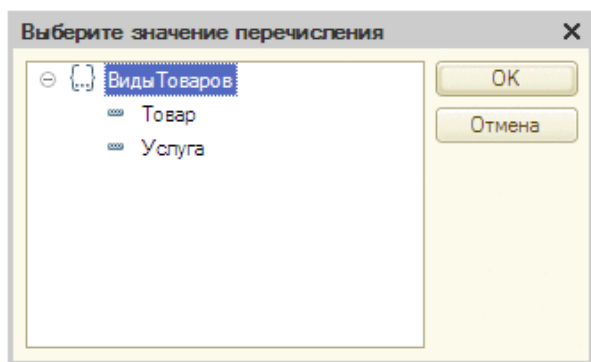


Рис. 563. Выбор перечисления

В результате выбора будет сформирован следующий текст:

[Копировать в буфер обмена](#)
 ПредопределенноеЗначение ("Перечисление . ВидыТоваров . Товар")

Установка курсора – используется для установки курсора в указанное место после вставки текста шаблона. В заменяющий текст шаблона будет вставлена конструкция вида `<?>`. Такая конструкция указывает, куда следует поместить курсор после ввода заменяющего текста. Если таких конструкций в заменяющем тексте несколько, то курсор будет установлен на место первой по порядку конструкции `<?>`.

При написании текста шаблона допускается многократное использование управляющих конструкций, а также использование уже готовых шаблонов.

33.6. Синтакс-помощник

Синтакс-помощник – средство, облегчающее разработку модулей. Основная задача синтакс-помощника – предоставить специалисту, выполняющему конфигурирование системы «1С:Предприятие», оперативную подсказку по встроенному языку.

Для вызова синтакс-помощника служит пункт **Сервис – Синтакс-Помощник**.

Окно синтакс-помощника разделено на две части. Разделение может быть выполнено по вертикали или по горизонтали.

В нижнюю (или правую) часть выводится описание выбранного раздела встроенного языка. Содержимое верхней (или левой) части определяется выбранной закладкой.

Окно содержит три закладки: [Содержание](#), [Индекс](#) и [Поиск](#). На первой размещается иерархический список элементов встроенного языка системы «1С:Предприятие»: операторов, управляющих конструкций, процедур и функций, системных констант и др., а вторая и третья предназначены для поиска по наименованию элемента встроенного языка и произвольного поиска по тексту описания соответственно.

Если выбрано горизонтальное расположение синтакс-помощника, то вместо закладок на командной панели отображаются одноименные кнопки.

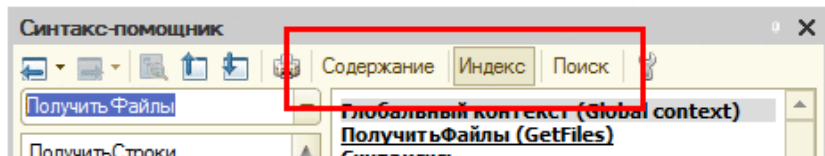


Рис. 564. Горизонтальное расположение

Имеется возможность настроить фильтр отображения дерева содержания и индекса синтакс-помощника по доступности объектов для того или иного режима запуска «1С:Предприятия».

33.6.1. Настройка синтакс-помощника

Описание кнопок командной панели синтакс-помощника приведено на [рис. 565](#).

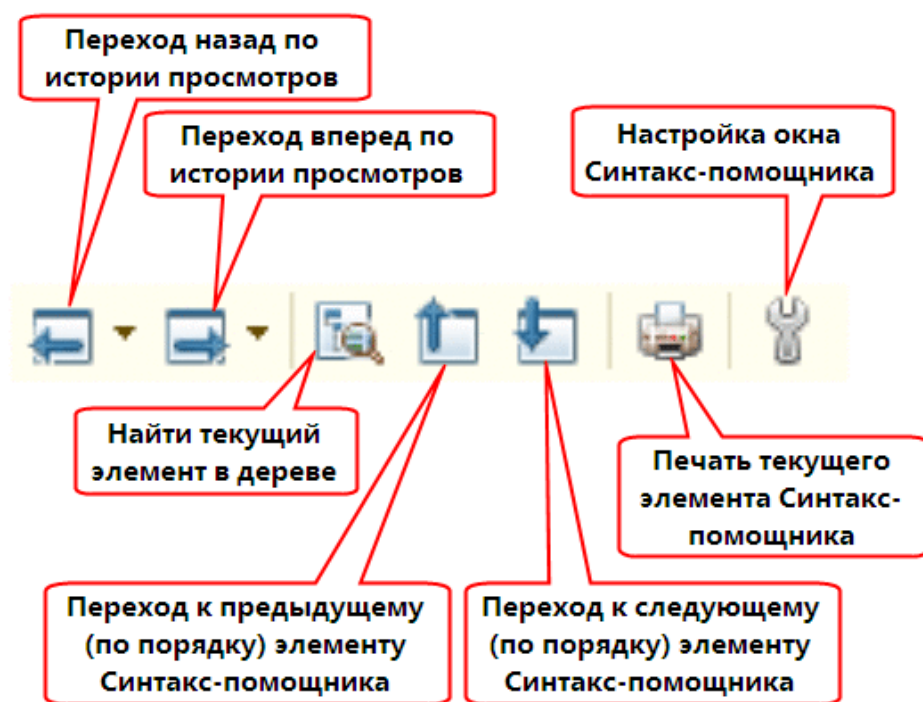


Рис. 565. Описание кнопок командной панели

33.6.2. Закладка «Содержание» синтакс-помощника

Для удобства все элементы встроенного языка объединены в тематические разделы, представленные в виде ветвей дерева, которое показывается в верхнем поле окна синтакс-помощника.

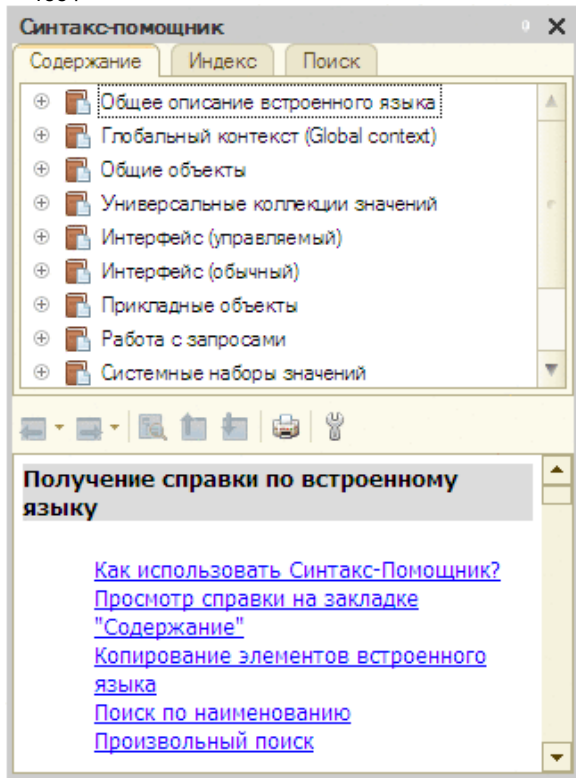


Рис. 566. Окно синтакс-помощника

Приемы работы в окне синтакс-помощника стандартны для иерархических структур данных, представленных в виде дерева. Для быстрого раскрытия ветви рекомендуем использовать клавиши «+», «-» и «*» цифровой клавиатуры: «+» раскрывает ветвь на один уровень, «-» – сворачивает, «*» – раскрывает все подветви данной ветви.

Для получения описания следует в верхней части окна синтакс-помощника выбрать наименование элемента языка. Описание элемента показывается в нижней части окна. Описание может содержать ссылки на описания упоминаемых элементов встроенного языка.

В окне расположена панель инструментов, с помощью которой можно просмотреть историю выбранных ранее описаний.

Для показа местоположения в структуре элементов встроенного языка элемента, текущее описание которого в данный момент выведено в нижнюю часть, в панели инструментов нужно нажать кнопку [Найти текущий элемент в дереве](#).

33.6.3. Копирование элементов встроенного языка

Синтакс-помощник имеет функцию копирования выбранного элемента встроенного языка в редактор текстов.

Для копирования необходимо выделить наименование нужного элемента языка в древовидном списке и использовать буфер обмена. В окно редактора текстов, в место расположения курсора, будет перенесена «заготовка» элемента встроенного языка.

Кроме использования буфера обмена для переноса «заготовок» элементов встроенного языка можно использовать режим [drag & drop](#) (перенеси и оставь): достаточно перетащить мышью выбранный элемент встроенного языка из окна синтакс-помощника в окно редактора текстов.

33.6.4. Поиск в синтакс-помощнике

Поиск в синтакс-помощнике может выполняться по наименованию элемента встроенного языка и по произвольному тексту описания.

На закладке [Индекс](#) окна синтакс-помощника выполняется поиск по наименованию элемента встроенного языка. В верхней части содержится поле для ввода наименования нужного элемента языка и полный список наименований элементов.

В процессе ввода наименования программа производит специальный режим контекстного поиска. При поиске с каждым набранным символом осуществляется переход на первое наименование списка, которое начинается вводимым текстом.

Для вызова режима контекстного поиска необходимо выбрать пункт [Поиск](#) в синтакс-помощнике, в контекстном меню текстового редактора. На экран выводится окно синтакс-помощника, открытого на закладке [Индекс](#).

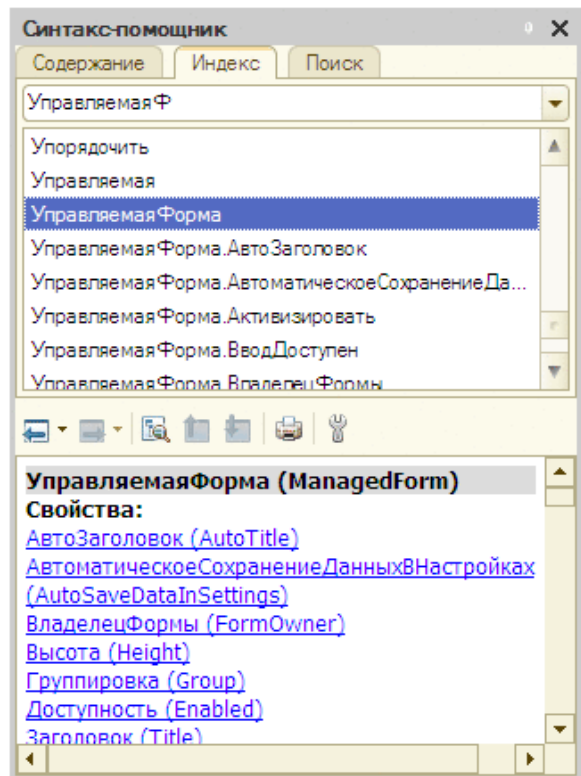


Рис. 567. Индекс синтакс-помощника

Если найденный (указанный) элемент используется в нескольких объектах, то на экран выводится окно [Выбор главы](#), содержащее список вхождений, относящихся к выбранному элементу.

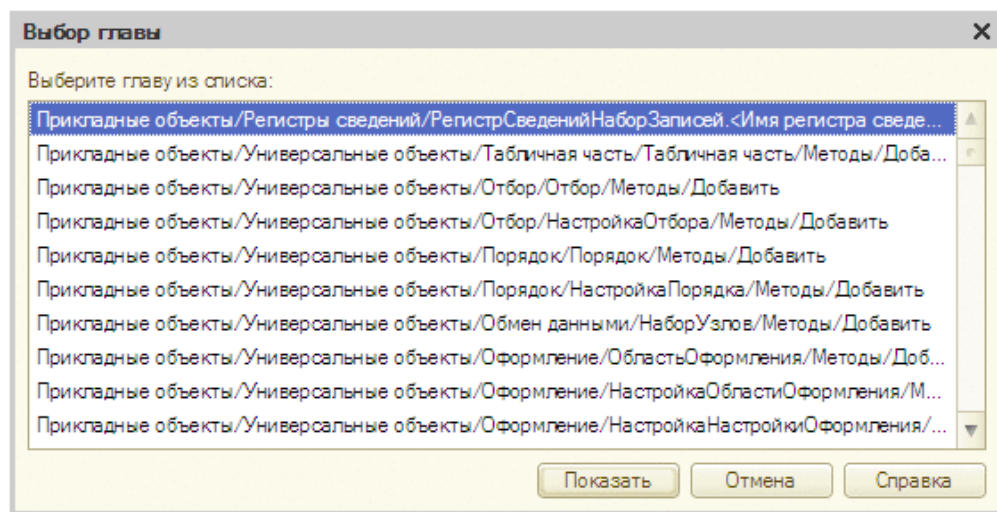


Рис. 568. Выбор главы

Теперь, если нажать кнопку [Показать](#), в окне синтакс-помощника будет выдано описание найденного элемента языка, выбранное в окне вхождений.

СОВЕТ. Если в окне [Выбор главы](#) находится много строк (как на [рис. 568](#)), то можно нажать клавиши **Ctrl + F** и в окне поиска ввести имя искомого объекта. Это ускорит поиск нужной главы.

Описание элемента встроенного языка можно искать в синтакс-помощнике непосредственно при редактировании модуля. Для выполнения поиска следует поместить курсор на элемент встроенного языка, по которому требуется получить описание, или выделить блок текста и нажать клавиши **Ctrl + F1**.

Если слово, на котором находится текстовый курсор, или выделенный блок является элементом встроенного языка, описанным в синтакс-помощнике, сразу откроется окно синтакс-помощника с описанием этого элемента языка.

Если элемент встроенного языка описан в нескольких тематических разделах синтакс-помощника, то сначала откроется окно [Выбор главы](#) для выбора нужного тематического раздела.

Если выбранное слово или блок текста не является элементом встроенного языка, при нажатии клавиш **Ctrl + F1** также будет открыто окно контекстного поиска в синтакс-помощнике.

Произведенный поиск запоминается в списке, который можно вызвать, в поле ввода наименования.

Для просмотра главы следует выбрать ее в списке и нажать клавишу **Enter**. Описание выбранной главы показывается в нижнем поле.

На закладке **Поиск** окна синтакс-помощника выполняется поиск по произвольному тексту описания. В верхней части содержится поле для ввода строки поиска и поле со списком найденных описаний элементов встроенного языка.

Для начала поиска нужно начать вводить текст. В процессе ввода программа выполняет поиск глав, в которых встречается введенный текст. Регистр ввода может быть любым, слова текста учитываются целиком (если не использован оператор «*») и с учетом морфологии. Допускается использование поисковых операторов (см. [здесь](#)).

В процессе ввода программа оперативно выводит список этих глав. Если введенный текст нигде не встречается, то ниже поля ввода программа выводит об этом сообщение.

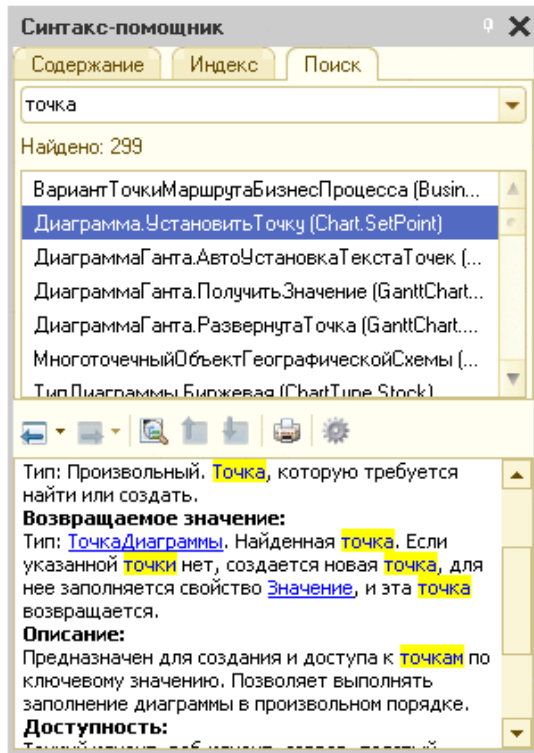


Рис. 569. Поиск в синтакс-помощнике

При открытии главы программа покажет описание таким образом, чтобы было видно первое вхождение указанного текста.

Для просмотра главы нужно выбрать ее в списке и нажать клавишу **Enter**. Описание выбранной главы показывается в нижнем поле.

После того как глава описания элемента встроенного языка найдена и окно синтакс-помощника открыто, для поиска положения главы в дереве описаний нужно использовать кнопку командной панели **Найти текущий элемент в дереве**.

Если в процессе просмотра выбирались несколько страниц, то с помощью команд **Вперед** и **Назад** можно вернуться к просмотренным страницам.

Описание синтаксиса поисковых выражений см. [здесь](#).

33.6.5. Использование методических материалов

При просмотре статьи синтакс-помощника имеется возможность перейти к различным методическим материалам, доступным на веб-ресурсах фирмы «1С». Для этого в нижней части статьи синтакс-помощника расположена гиперссылка **Методические материалы**.

Справочник Менеджер (CatalogsManager)

<Имя справочника> (<Catalog name>)

Использование:

Только чтение.

Описание:Тип: [Справочник Менеджер](#).

Набор свойств содержит менеджеры отдельных справочников. Доступ к менеджеру осуществляется по имени, как оно задано в Конфигураторе.

Доступность:

Сервер, толстый клиент, внешнее соединение, мобильное приложение(сервер).

[Методическая информация](#)

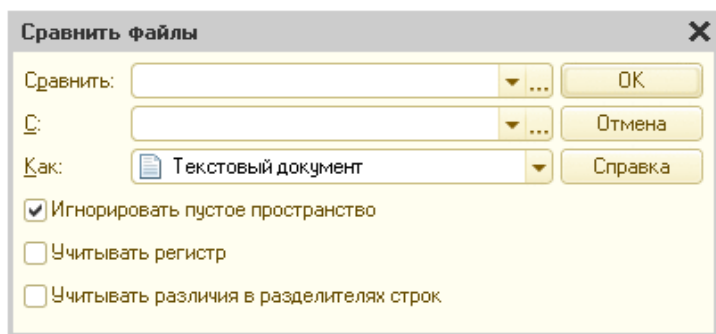
Рис. 570. Методическая информация

При нажатии на эту ссылку откроется веб-браузер, в котором будет показана специальная страница со ссылками на дополнительные материалы по изучаемой статье синтаксис-помощника. Для перехода по представленным ссылкам может потребоваться авторизация на соответствующем сайте.

В ОС Windows ссылка будет открыта в Microsoft Internet Explorer, а в ОС Linux используемый веб-браузер будет зависеть от настроек системы.

33.7. Сравнение файлов

Режим сравнения файлов дает возможность пользователю сравнить два любых файла. Для этого необходимо выбрать пункт [Файл – Сравнить файлы](#). На экран будет выдан диалог для выбора сравниваемых файлов.

**Рис. 571. Сравнение файлов**

В полях [Сравнить](#) и [С](#) необходимо указать имена сравниваемых файлов. Имена файлов можно ввести вручную, выбрать из выпадающего списка или нажатием кнопки выбора (...) вызвать стандартный диалог открытия файла и выбрать файл в этом диалоге. В диалоге, в поле [Тип файлов](#), можно указать любой тип исходного файла.

В поле [Как](#) указывается вид сравнения файлов. Сравниваются текстовый и табличный документы, внешняя обработка. Также допускается двоичное сравнение, при котором сравнение производится на двоичном уровне. В этом случае результатом сравнения будет сообщение о совпадении или различии файлов.

Если в полях [Сравнить](#) и [С](#) указаны файлы различных типов, то выбранные файлы будут интерпретироваться при сравнении к виду, указанному в поле [Как](#). Затем выполняется сравнение, и на экран выводится информация о его результатах.

Для начала сравнения нужно нажать кнопку [OK](#).

Если выбран вид сравнения [Текстовый документ](#), то выбранные файлы будут интерпретироваться при сравнении как текстовые, несмотря на то, что в полях [Сравнить](#) и [С](#) могут быть выбраны нетекстовые (по расширению) файлы. Если указанные файлы или один из них не может быть интерпретирован как текстовый, то производится двоичное сравнение.

33.7.1. Сравнение внешних обработок

Если сравниваются внешние обработки, то выполняется стандартная процедура сравнения внешних обработок (подробнее см. [здесь](#)).

Выбор флажка [Устанавливать соответствие по именам объектов](#) означает, что установка соответствия объектов сначала будет производиться по именам, а затем по внутренним идентификаторам. Если флажок не установлен, то

установка соответствия производится по внутренним идентификаторам.

33.8. Встроенная система получения справочной информации

Информацию по работе с системой «1С:Предприятие» и по конкретным объектам и режимам можно получить, используя систему справочной информации. Она вызывается в любой момент с помощью пункта [Справка](#) или нажатием клавиши **F1**. При этом вызывается раздел справочной информации, соответствующий тому режиму, в котором сейчас работает пользователь.

Пункт меню [Содержание](#) и клавиша **Shift + F1** позволяют обратиться к общему содержанию справочной информации, из которого можно получать информацию по работе в режиме Конфигуратор.

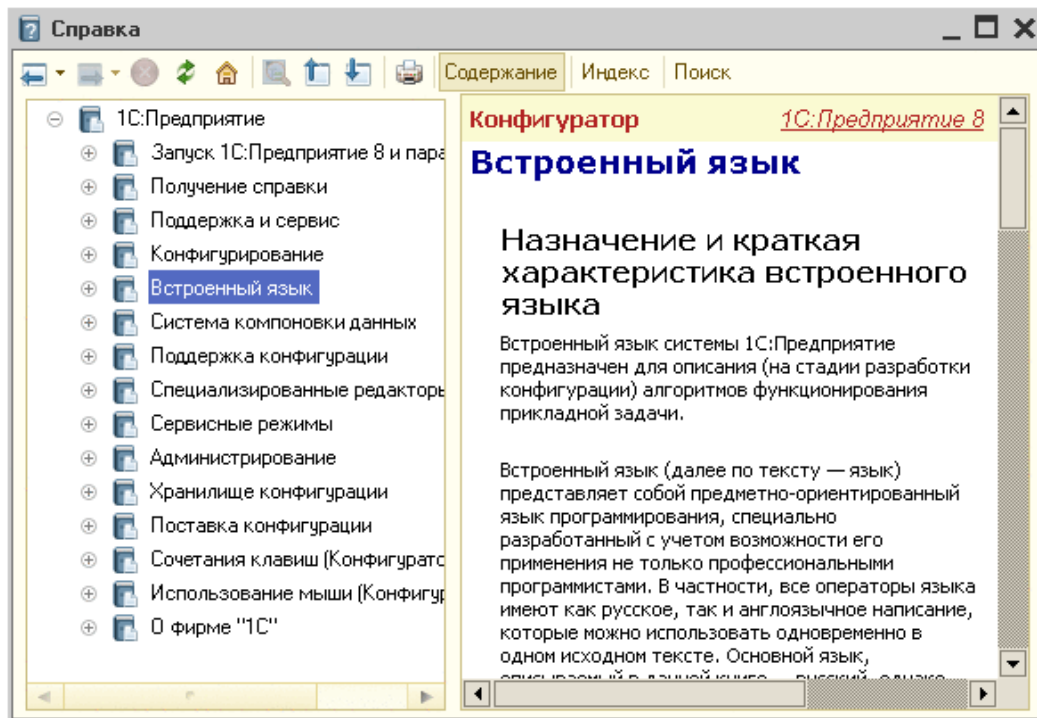


Рис. 572. Содержание справки

Справочная информация содержит описания разделов конфигурирования, описания встроенного языка и дополнительной справочной информации.

В верхней части окна справочной информации расположена панель инструментов, с помощью которых осуществляется навигация и поиск информации. Ниже панели инструментов отображается содержание справки.

Глава 34. Внешние компоненты

«1С:Предприятие» является расширяемой системой. Для расширения функциональных возможностей используются внешние компоненты.

В системе используются две технологии создания внешних компонент:

- с использованием Native API,
- с использованием технологии COM.

Это позволяет создавать внешние компоненты для ОС семейства Windows и ОС семейства Linux, а также для веб-клиента, работающего в веб-браузерах Microsoft Internet Explorer, а также Mozilla Firefox, Google Chrome и Safari.

Внешние компоненты, предназначенные для работы в среде веб-клиента, должны быть упакованы в ZIP-архив особой структуры. Такой файл можно использовать также на сервере «1С:Предприятия» и в других клиентах.

При работе с компонентами, написанными с использованием Native API, следует придерживаться следующей схемы работы:

- В интерфейсе реализуется команда, которая вызывает установку внешней компоненты на компьютер пользователя (метод глобального контекста `УстановитьВнешнююКомпоненту()`). Данная операция обязательная для работы в тонком и веб-клиентах.
- Компонента подключается с помощью метода `ПодключитьВнешнююКомпоненту()`. При этом следует помнить, что попытка подключить неустановленную компоненту приведет к ошибке.
- Подключенная компонента используется согласно предоставленному программному интерфейсу.

СОВЕТ. Не следует объединять в один программный код установку и подключение внешней компоненты. Установка считается однократным событием, и повторная установка вызовет интерактивное исключение.

Если используется компонента, упакованная в ZIP-архив, то для указания параметров внешней компоненты (имя устанавливаемого файла, тип, архитектура, используемый веб-браузера) служит специальный файл-манифест, формируемый разработчиком компоненты.

Получаемые из конфигурации или информационной базы внешние компоненты сохраняются после получения и используются при последующем подключении без повторного получения (компонента, используемая на сервере, сохраняется только на время работы рабочего процесса сервера).

ПРИМЕЧАНИЕ. Подключение внешних компонент, выполненных по технологии COM, с помощью методов `ЗагрузитьВнешнююКомпоненту()`, `ПодключитьВнешнююКомпоненту()`, а также регистрация объектов компоненты выполняются «для пользователя». Если регистрация «для пользователя» завершилась неудачно, то предпринимается попытка выполнить регистрацию «для компьютера».

Если внешняя компонента подключается с помощью метода `ПодключитьВнешнююКомпоненту()`, то формирование параметра в операторе `Новый()` (имени создаваемого типа) выполняется из следующих частей:

- Текст `Addin.`;
- Имя, указанное в качестве параметра `Имя` метода `ПодключитьВнешнююКомпоненту()`;
- Имя объекта, реализованного во внешней компоненте.

Если во внешней компоненте реализован объект Таймер и подключение внешней компоненты выполняется следующим образом: `ПодключитьВнешнююКомпоненту(НавСсылка, "MyName")`; то для создания объекта из внешней компоненты, следует указать следующий идентификатор: `Новый("Addin.MyName.Таймер")`.

34.1. Особенности работы в тонком и толстом клиентах

При работе в толстом и тонком клиентах можно использовать как внешние компоненты, разработанные по технологии COM, так и внешние компоненты, разработанные по технологии Native API (как отдельными файлами на диске, так и в ZIP-архивах особого вида).

Для обновления компоненты достаточно повторно выполнить операцию установки внешней компоненты с помощью метода `УстановитьВнешнююКомпоненту()`.

В момент установки внешние компоненты устанавливаются в каталог `%APPDATA%\1C\1Cv8\ExtCompT`. Каталог установки внешних компонент не считается кешем и не очищается при вызове «1С:Предприятия» с ключом командной строки `ClearCache`. Использование метода `УстановитьВнешнююКомпоненту()` для тонкого клиента является обязательным.

34.2. Особенности работы в веб-клиенте

При работе в веб-клиенте можно использовать компоненты, разработанные по технологии Native API или COM и упакованные в ZIP-архив особого вида. Компоненты, разработанные по технологии COM, можно использовать только в том случае, если веб-клиент работает в ОС семейства Windows. При установке файла внешней компоненты будет запрошено разрешение пользователя на установку расширения.

Особенности настройки веб-браузеров для работы с внешними компонентами описаны в книге «1С:Предприятие 8.3. Руководство администратора».

Каждая внешняя компонента устанавливается в виде отдельного расширения веб-браузера. Такое расширение представляет собой

установочный пакет, подготовленный специальным образом для конкретного вида браузеров. Внешние компоненты устанавливаются:

- для веб-браузера Microsoft Internet Explorer – в каталог `%ALLUSERSPROFILE%\Application Data\1C\1CEWebExt (%ALLUSERSPROFILE%\1C\1CEWebExt` для ОС Windows Vista и выше);
- для веб-браузера Mozilla Firefox – в каталог профиля текущего пользователя веб-браузера (создается веб-браузером);
- для веб-браузера Google Chrome – в каталог профиля текущего пользователя веб-браузера (создается веб-браузером);
- для веб-браузера Safari – определяется разработчиком внешней компоненты.

Возможно удаление внешних компонент:

- для веб-браузера Microsoft Internet Explorer – через механизм установки/удаления программ панели управления;
- для веб-браузера Mozilla Firefox – через механизм работы с дополнениями веб-браузера;
- для веб-браузера Google Chrome – через механизм работы с плагинами веб-браузера (переход по ссылке <chrome://plugins/>);
- для веб-браузера Safari – стандартным способом.

ВНИМАНИЕ! Подключение внешних компонент из файла на диске для веб-клиента в целях безопасности не поддерживается.

Также следует помнить, что работа с COM-объектами доступна только в веб-браузере Microsoft Internet Explorer и ОС Windows. Также отсутствует поддержка типа `COMSafeArray` и возможность совершать вызовы методов COM-объектов, которые работают со значениями данного типа.

Для обновления компоненты необходимо выполнить одну из следующих операций:

- удалить компоненту, как указано выше, и заново выполнить установку с помощью метода `УстановитьВнешнююКомпоненту()`;
- изменить имя объекта компоненты в исходном тексте компоненты и в файле манифеста (при этом имя файла с компонентой может не меняться) и заново выполнить установку с помощью метода `УстановитьВнешнююКомпоненту()`.

При использовании внешних компонент в конфигурации следует придерживаться определенных правил и рекомендаций. Выполнение этих правил необходимо для того, чтобы внешние компоненты устанавливались без проблем на всех поддерживаемых веб-браузерах.

- Исключить использование конструкций вида:

```
Если Не ПодключитьВнешнююКомпоненту() Тогда
    УстановитьВнешнююКомпоненту()
КонецЕсли;
```

[Копировать в буфер обмена](#)

- Установка внешней компоненты или расширений платформы (расширение работы с файлами, расширение работы с криптографией) должна быть интерактивной. Пользователь прикладного решения должен самостоятельно принять решение об установке. В диалоге установки должно быть сказано, для чего нужна компонента и что не будет работать, если ее не устанавливать.
- Рекомендуется интегрировать установку внешней компоненты в процесс выполнения прикладного действия. Например:
 1. Пользователь воспользовался командой [Отправить отчет](#).
 2. Для этого прикладному решению необходимо, чтобы была установлена какая-либо внешняя компонента.
 3. Прикладное решение проверяет, установлена или нет внешняя компонента.
 4. Если внешняя компонента не установлена, пользователю отображается диалоговое окно, содержащую информацию о том, что для отправки отчета нужно установить компоненту и две кнопки: для установки внешней компоненты и для продолжения отправки отчета.
 5. Затем пользователь последовательно выполняет два интерактивных действия:
 1. Установку внешней компоненты;
 2. Продолжение отправки отчета.
- В прикладном решении должны быть реализованы инструменты для установки пользователем внешних компонент и расширений платформы в любой момент работы. Действия по установке внешних компонент и расширений платформы не обязательно должны выполняться в процессе выполнения какого-либо действия.

34.3. Особенности работы на сервере

При работе на сервере «1С:Предприятия» допустимо использовать только компоненты, разработанные по технологии Native API, которые могут быть как отдельными файлами, так и упакованными в специальные ZIP-архивы.

Для подключения внешней компоненты используется метод `ПодключитьВнешнююКомпоненту()` (без использования метода `УстановитьВнешнююКомпоненту()`, который недоступен на сервере).

При работе на сервере желательно выполнять подключение внешней компоненты непосредственно перед ее использованием. Кроме того, компоненты, используемые на сервере, должны поддерживать все архитектуры и платформы. Это связано с тем, что в общем

случае серверный код может исполняться на разных рабочих процессах (в разные моменты времени), которые могут исполняться на разных компьютерах. При этом компьютеры могут быть как разной архитектуры, так и функционировать под управлением различных операционных систем.

Файл внешней компоненты сохраняется рабочим процессом до своего перезапуска, поэтому повторное подключение внешней компоненты (до перезапуска рабочего процесса или переключения исполнения на другой рабочий процесс) происходит быстрее, чем первое подключение.

34.4. Асинхронная работа с внешней компонентой

Использование внешних компонент в асинхронном режиме имеет ряд особенностей:

1. Подключение внешней компоненты выполняется с помощью асинхронного метода `НачатьПодключениеВнешнейКомпоненты()`.
2. Для обращения к свойствам внешней компоненты следует использовать асинхронные методы `НачатьУстановку<ИмяСвойства>()` и `НачатьПолучение<ИмяСвойства>()`.
3. Вместо обращения к методу компоненты следует использовать асинхронные методы `НачатьВызов<ИмяМетода>()`.

Методы `Начать...()` автоматически добавляются платформой к существующим свойствам и методам внешней компоненты. Саму внешнюю компоненту не требуется отдельно перерабатывать для поддержки работы в асинхронном режиме, однако необходимо выполнить перекомпиляцию существующей внешней компоненты для поддержки асинхронной работы.

Для метода `НачатьВызов...()` первым параметром следует указывать описание оповещения, а остальные параметры соответствуют набору параметров оригинального метода внешней компоненты.

При использовании веб-браузера Google Chrome поддерживается только асинхронное подключение и использование внешних компонент, в том время как для других веб-браузеров допустимо использование как синхронного, так и асинхронного способа работы. Это определяется свойством конфигурации [Режим использования синхронных вызовов расширений и внешних компонент](#) (см. [здесь](#)).

34.5. Примеры работы с внешней компонентой

34.5.1. Технология Native API

Внешние компоненты, выполненные по технологии Native API, можно подключать не только в клиентских приложениях, но и на сервере приложений «1С:Предприятия».

При работе компоненты на сервере вызов `ПодключитьВнешнююКомпоненту()` необходимо выполнять каждый раз перед созданием экземпляра внешней компоненты, т. к. в общем случае неизвестно, на каком сервере будет исполняться вызов (это может быть Windows, Linux, 32-разрядная или 64-разрядная ОС).

34.5.1.1. Установка внешней компоненты на клиента

[Копировать в буфер обмена](#)

```
УстановитьВнешнююКомпоненту("ОбщийМакет.ДрайверСканераШтрихкодов");
```

34.5.1.2. Загрузка из файла на диске

Файл внешней компоненты должен находиться на компьютере пользователя и быть доступен в пути поиска (переменная окружения `PATH`) операционной системы.

[Копировать в буфер обмена](#)

```
СисИнфо = Новый СистемнаяИнформация;
Если СисИнфо.ТипПлатформы = ТипПлатформы.Windows_x86 Тогда
    ПодключитьВнешнююКомпоненту("AddInCPP.dll", "MyName", ТипВнешнейКомпоненты.Native);
ИначеЕсли СисИнфо.ТипПлатформы = ТипПлатформы.Windows_x86_64 Тогда
    ПодключитьВнешнююКомпоненту("AddInCPP64.dll", "MyName", ТипВнешнейКомпоненты.Native);
ИначеЕсли СисИнфо.ТипПлатформы = ТипПлатформы.Linux_x86 Тогда
    ПодключитьВнешнююКомпоненту("libAddInCPP.so", "MyName", ТипВнешнейКомпоненты.Native);
Иначе
    ПодключитьВнешнююКомпоненту("libAddInCPP64.so", "MyName", ТипВнешнейКомпоненты.Native);
КонецЕсли;
ОбъектКомпоненты = Новый("AddIn.MyName.ComponentExtension");
```

ПРИМЕЧАНИЕ. Следует обратить внимание на формирование параметра в команде `Новый`.

34.5.1.3. Загрузка компоненты из макета

[Копировать в буфер обмена](#)

```
СисИнфо = Новый СистемнаяИнформация;
Если СисИнфо.ТипПлатформы = ТипПлатформы.Windows_x86 Тогда
    ПодключитьВнешнююКомпоненту("Обработка.Компонента.Макет.AddInWindows32", "MyName", ТипВнешнейКомпоненты.Native);
ИначеЕсли СисИнфо.ТипПлатформы = ТипПлатформы.Windows_x86_64 Тогда
    ПодключитьВнешнююКомпоненту("Обработка.Компонента.Макет.AddInWindows64", "MyName", ТипВнешнейКомпоненты.Native);
ИначеЕсли СисИнфо.ТипПлатформы = ТипПлатформы.Linux_x86 Тогда
    ПодключитьВнешнююКомпоненту("Обработка.Компонента.Макет.AddInLinux32", "MyName", ТипВнешнейКомпоненты.Native);
Иначе
    ПодключитьВнешнююКомпоненту("Обработка.Компонента.Макет.AddInLinux64", "MyName", ТипВнешнейКомпоненты.Native);
КонецЕсли;
ОбъектКомпоненты = Новый("AddIn.MyName.ComponentExtension");
```

ПРИМЕЧАНИЕ. Следует обратить внимание на формирование параметра в команде `Новый`.

34.5.1.4. Загрузка из информационной базы

В информационной базе должен быть справочник `ВнешниеКомпоненты` и реквизиты для указанных названий с типом `ХранилищеЗначения` с файлами внешних компонент.

[Копировать в буфер обмена](#)

```
Перем Ссылка;
СисИнфо = Новый СистемнаяИнформация;
Если СисИнфо.ТипПлатформы = ТипПлатформы.Windows_x86 Тогда
    Ссылка = ПолучитьНавигационнуюСсылку(Справочники.ВнешниеКомпоненты.НайтиПоКоду("000000001"), "КомпонентаWindows32");
ИначеЕсли СисИнфо.ТипПлатформы = ТипПлатформы.Windows_x86_64 Тогда
    Ссылка = ПолучитьНавигационнуюСсылку(Справочники.ВнешниеКомпоненты.НайтиПоКоду("000000001"), "КомпонентаWindows64");
ИначеЕсли СисИнфо.ТипПлатформы = ТипПлатформы.Linux_x86 Тогда
    Ссылка = ПолучитьНавигационнуюСсылку(Справочники.ВнешниеКомпоненты.НайтиПоКоду("000000001"), "КомпонентаLinux32");
Иначе
    Ссылка = ПолучитьНавигационнуюСсылку(Справочники.ВнешниеКомпоненты.НайтиПоКоду("000000001"), "КомпонентаLinux64");
КонецЕсли;
ПодключитьВнешнююКомпоненту(Ссылка, "MyName", ТипВнешнейКомпоненты.Native);
ОбъектКомпоненты = Новый("AddIn.MyName.ComponentExtension");
```

ПРИМЕЧАНИЕ. Следует обратить внимание на формирование параметра в команде `Новый`.

В рассмотренных примерах работы с внешними компонентами по технологии `Native API` также можно подключать внешние компоненты, реализованные с использованием технологии `COM` только для ОС `Windows`.

34.5.1.5. Подключить компоненту из ZIP-архива

В конфигурации должен быть общий макет с именем `ДрайверСканераШтрихКодов` с типом `ДвоичныеДанные`. В макете находится ZIP-архив особой структуры, в котором содержатся внешние компоненты для всех поддерживаемых операционных систем, браузеров и архитектур процессоров.

[Копировать в буфер обмена](#)

```
ПодключитьВнешнююКомпоненту("ОбщийМакет.ДрайверСканераШтрихКодов", "Сканер");
Компонента = Новый("AddIn.Сканер.BarcodeReader");
```

ПРИМЕЧАНИЕ. Следует обратить внимание на формирование параметра в команде `Новый`.

34.5.2. COM-технология

34.5.2.1. Загрузить внешнюю компоненту

[Копировать в буфер обмена](#)

```
ЗагрузитьВнешнююКомпоненту("MyComponent.dll");
Компонента = Новый("AddIn.ComponentExtension");
```

34.5.2.2. Подключить внешнюю компоненту

[Копировать в буфер обмена](#)

```
ПодключитьВнешнююКомпоненту("AddIn.MyObject");
Компонента = Новый("AddIn.ComponentExtension");
```

ПРИМЕЧАНИЕ. Команда `ЗагрузитьВнешнююКомпоненту()` используется для совместимости с предыдущими версиями «1С:Предприятия».

Глава 35. Особенности разработки кроссплатформенных прикладных решений

Данная глава содержит рекомендации по разработке прикладных решений, работающих под управлением и ОС Windows и ОС Linux.

35.1. Особенности файловой системы

В ОС Linux при работе с файлами следует учитывать, что имена файлов и каталогов являются регистрочувствительными. Например, файлы с именами `file.txt` и `File.txt` являются разными файлами и могут располагаться в одном каталоге.

В ОС Linux отсутствует понятие дисков, которое активно используется в ОС Windows. В ОС Linux существует корневая файловая система ("/") в любое место которой (в общем случае) монтируются другие файловые системы, расположенные на других носителях (диски и разделы, флэш-накопители, ISO-файлы и т. д.). Наиболее распространенными (но не обязательными!) местами монтирования внешних носителей являются точки монтирования (каталоги) `/mnt` или `/media`. При монтировании внешних носителей, они могут монтироваться как `/media/<Метка раздела>` или как `/media/<Идентификатор устройства>`. При этом автоматическое монтирование внешнего носителя не гарантировано.

Отсутствие буквенных обозначений дисков в ОС Linux может привести к следующим проблемам:

- В прикладном решении используются явно заданные пути, которые ориентированы только на использование ОС Windows;
- В прикладном решении используются механизмы проверки введенных путей, где проверяется, в том числе, и указание буквы диска;
- При указании пути, который начинается с корневого каталога, следует помнить, что для ОС Linux такой путь однозначно определяет положение каталога или файла, а для ОС Windows – нет. Например, путь `\temp\file.txt` (или его linux-аналог `/temp/file.txt`) в ОС Linux точно укажет расположение и имя файла: это файл `file.txt`, расположенный в каталоге `temp` корневой файловой системы. Для ОС Windows все не так однозначно. Такой путь означает, что файл `file.txt` расположен в каталоге `temp`, который расположен в корневом каталоге текущего диска. При этом текущий диск может быть разным при двух последующих запусках одного и того же клиентского приложения с одной и той-же информационной базой.

При разработке прикладных решений следует обратить внимание на следующий момент: в ОС Windows и ОС Linux используются разные разделители пути. В ОС Windows используется "\" («обратный слэш»), а в ОС Linux – "/" («прямой слэш»). При работе с методами встроенного языка такую разницу учитывать нет необходимости, т. к. они корректно работают с любым разделителем. Но если требуется передавать путь во внешнее приложение, следует указывать те разделители пути, которые используются в той ОС, под управлением которой работает «1С:Предприятие». Для получения информации о разделителях пути предназначены функции `ПолучитьРазделительПути()`, `ПолучитьРазделительПутиКлиента()`, `ПолучитьРазделительПутиСервера()`.

Наряду с различными разделителями пути, ОС Windows и ОС Linux по разному обрабатывают файловые маски. Например, маска `*.*` означает указание всех файлов в ОС Windows и указание всех файлов, содержащих в имени "." – в ОС Linux. Такое различие может приводить к неожиданному поведению методов встроенного языка `НайтиФайлы()`, `УдалитьФайлы()` и объекта `ДиалогВыбораФайла`. Особенности указания файловых масок в различных операционных системах отражены в синтакс-помощнике для функций `НайтиФайлы()`, `УдалитьФайлы()`. Для получения файловой маски, описывающей все файлы, предназначены функции `ПолучитьМаскуВсеФайлы()`, `ПолучитьМаскуВсеФайлыКлиента()`, `ПолучитьМаскуВсеФайлыСервера()`.

Еще одной особенностью ОС Linux является то, что в ней нет возможности обращаться к сетевым ресурсам, используя UNC-нотацию (в этом случае адрес имеет следующий вид: `\\server\folder\folder\file.ext`). Поэтому, при необходимости использовать какой-либо внешний сетевой ресурс, его необходимо предварительно смонтировать на корневую файловую систему. При этом следует помнить, что каждое монтирование может выполняться в разную точку монтирования (разный каталог). Также можно использовать доступ к внешним сетевым ресурсам с помощью специальных приложений

(например, `smbclient`), однако следует учитывать, что такие приложения не обязательно будут установлены в используемой копии ОС Linux.

Более подробную информацию об устройстве файловой системы ОС Linux и принципах работы с ней можно получить в документации к операционной системе или в сети Интернет.

35.2. Работа с внешними устройствами

Имена внешних устройств (и доступ к ним) по-разному выполняется в различных операционных системах. Например, последовательные порты в ОС Windows будут называть `COM1`, `COM2` и т. д. В ОС Linux (в зависимости от дистрибутива и устройства) порты могут именоваться так: `/dev/ttyS0`, `/dev/ttyS1` и т. д. или так: `/dev/ttyUSB0`, `/dev/ttyUSB1` и т. д. Эту особенность следует учитывать как при разработке внешних компонент, работающих с оборудованием, так и при разработке диалогов настройки внешних компонент. Например, перечень портов, которые можно указать в обработке, следует получать непосредственно из настраиваемой внешней компоненты с помощью специального метода.

35.3. Ограниченность в правах

В ОС Linux не принято работать от имени учетной записи супер-пользователя (`root`). Для обычного пользователя для записи всегда доступен только домашний каталог и, как правило, каталог временных файлов. Остальная часть файловой системы может быть доступна или только на чтение или быть недоступной совсем (вся или некоторые каталоги). В связи с этим, при разработке следует учитывать, какой доступ будет иметь пользователь, от лица которого выполняется «1С:Предприятие», в каталоги, которые возвращают методы глобального контекста:

- `КаталогДокументов()` – доступ на чтение и запись;
- `КаталогПрограммы()` – доступ на чтение;
- `КаталогВременныхФайлов()` – доступ на чтение и запись;
- `РабочийКаталогДанныхПользователя()` – доступ на чтение и запись.

Для доступа к внешним устройствам может потребоваться, чтобы пользователь входил в какую-либо специализированную группу (или группы). Подробнее об этом следует смотреть в документации к используемой операционной системе.

35.4. Шрифты

При разработке прикладных решений рекомендуется всегда использовать стилевые шрифты, без явного указания гарнитуры шрифта. При необходимости можно использовать шрифты, входящие в состав Microsoft Core Fonts (<http://sourceforge.net/projects/corefonts/>). Не рекомендуется использовать в прикладных решениях другие шрифты, т. к. они могут отсутствовать той или иной используемой операционной системой, при этом внешний вид прикладного решения будет искажен.

35.5. ПолеHTMLДокумента

В «1С:Предприятии», работающем под управлением ОС Linux, для работы с HTML используется WebKit. «1С:Предприятие», работающее под управлением ОС Windows, для работы с HTML используется текущая версия Internet Explorer. В связи с этим, при работе с HTML следует учитывать следующие особенности:

- Отображение HTML-документа в различных операционных системах может различаться.
- При доступе к DOM-модели HTML-документа с помощью свойства `ПолеHTMLДокумента.Документ` следует использовать только те свойства и методы, которые доступны во всех веб-браузерах, поддерживаемых системой «1С:Предприятие». Это обусловлено как использованием разных средств для отображения HTML-документа в клиентском приложении, так и тот факт, что для работы с прикладным решением может использоваться веб-доступ, где увеличивается количество используемых веб-браузеров и их версий.

35.6. Механизмы COM, OLE, ActiveDocument

Данные механизмы недоступны в ОС Linux. Для интеграции необходимо использовать другие средства, например файловый обмен в формате XML или web-сервисы.

Для внешних компонент, реализованных по COM-технологии, рекомендуется переработать их с использованием технологии NativeAPI.

Глава 4. Встроенный язык

«1С:Предприятие» является гибкой настраиваемой системой, с помощью которой можно решать широкий круг задач в сфере автоматизации деятельности предприятий. Специфические алгоритмы конфигурации описываются в конфигураторе системы «1С:Предприятие», в программных модулях, содержащих тексты на встроенном языке системы «1С:Предприятие».

4.1. Назначение и краткая характеристика встроенного языка

Встроенный язык системы «1С:Предприятие» предназначен для описания (на стадии разработки конфигурации) алгоритмов функционирования прикладной задачи.

Встроенный язык (далее по тексту – язык) представляет собой предметно-ориентированный язык программирования, специально разработанный с учетом возможности его применения не только профессиональными программистами. В частности, все операторы языка имеют как русское, так и англоязычное написание, которое можно использовать одновременно в одном исходном тексте. Основной язык, описываемый в данной книге, – русский, однако для каждого оператора языка приводится его англоязычный синоним.

При своей относительной простоте язык обладает некоторыми объектно-ориентированными возможностями, например, правила доступа к свойствам и методам специализированных типов данных (документам, справочникам и т. п.) подобны свойствам и методам объектов, используемых в других объектно-ориентированных языках. Однако специализированные типы данных не могут определяться средствами самого языка, а задаются в визуальном режиме.

Типизация переменных в языке не жесткая, т. е. тип переменной определяется ее значением. Переменные не обязательно объявлять в явном виде. Неявным определением переменной является ее первое упоминание в левой части оператора присваивания. Возможно также явное объявление переменных при помощи соответствующего оператора. Допускается применение массивов, структур, соответствий и других универсальных коллекций значений.

4.2. Формат исходных текстов программных модулей

4.2.1. Что такое программный модуль

Программные модули в конфигурации системы «1С:Предприятие» не являются самостоятельными программами в общепринятом понимании этого слова, поскольку они являются только частью всей конфигурации. Программный модуль – это текст на встроенном языке, в котором размещены тексты процедур и функций с необходимыми алгоритмами, вызываемые системой в определенные моменты работы. Поэтому программный модуль не имеет формальных границ своего описания типа: «Начало модуля» – «Конец модуля».

Место размещения конкретного программного модуля предоставляется конфигуратором в тех точках конфигурации, которые требуют описания специфических алгоритмов функционирования. Эти алгоритмы следует оформлять в виде процедур или функций, которые будут вызваны самой системой в заранее предусмотренных ситуациях (например, при нажатии кнопки в диалоговом окне).

Каждый отдельный программный модуль воспринимается системой как единое целое, поэтому все процедуры и функции программного модуля выполняются в едином контексте.

4.2.2. Контекст выполнения программного модуля

Каждый программный модуль связан с остальной частью конфигурации. Эта связь называется **контекстом** выполнения модуля.

Следует различать два вида контекста:

- глобальный контекст,
- локальный контекст выполнения конкретного модуля.

4.2.2.1. Глобальный контекст

Глобальный контекст образуется:

- значениями свойств и методов глобального контекста;
- системными перечислениями и системными наборами значений (например, `КодВозвратаДиалога` и `Символь`).

Глобальный контекст виден всем программным модулям и определяет общую языковую среду конфигурации.

4.2.2.2. Локальный контекст

Локальный контекст модуля образуется тем конкретным местом конфигурации задачи, для которого использован программный модуль. Локальный контекст виден только конкретному программному модулю и определяет для модуля набор непосредственно доступных модулю объектов, их свойств и методов.

4.2.3. Виды программных модулей

В системе «1С:Предприятие» существуют несколько видов программных модулей. Они различаются по месту размещения и доступному контексту.

4.2.3.1. Модуль управляемого приложения

Модулем управляемого приложения называется модуль, который автоматически выполняется в момент загрузки конфигурации, при старте системы «1С:Предприятие» в следующих режимах:

- тонкого клиента,
- веб-клиента,
- толстого клиента в режиме управляемого приложения.

Модуль управляемого приложения предназначен для отработки действий, связанных с сеансом работы конечного пользователя (прежде всего обработки начала и окончания сеанса работы). Модуль управляемого приложения недоступен для процедур, работающих на сервере. В нем рекомендуется реализовывать только обработчики соответствующих событий. Следует помнить, что необработанное исключение в каком-либо обработчике события модуля управляемого приложения (`ПередНачаломРаботыСистемы()`, `ПриНачалеРаботыСистемы()`, `ПередЗавершениемРаботыСистемы()`, `ПриЗавершенииРаботыСистемы()`) приведет к аварийному завершению работы всей системы.

Например, в следующем примере аварийного завершения не произойдет:

```
Процедура ПриНачалеРаботыСистемы()
    Попытка
        а=1/0;
    Исключение
        Сообщить("Деление на 0");
    КонецПопытки;
КонецПроцедуры
```

[Копировать в буфер обмена](#)

В тоже время следующий пример приведет к аварийному завершению:

```
Процедура ПриНачалеРаботыСистемы()
    а=1/0;
КонецПроцедуры
```

[Копировать в буфер обмена](#)

Процедуры и функции модуля управляемого приложения, а также переменные, для которых в заголовке указано ключевое слово **Экспорт**, являются доступными:

- в неглобальных клиентских общих модулях,
- клиентских процедурах и функциях модуля команды,
- клиентских процедурах и функциях модуля управляемой формы.

В контексте модуля управляемого приложения доступны:

- часть глобального контекста, которая может исполняться в управляемом приложении;
- экспортируемые процедуры и функции любых клиентских общих модулей;
- экспортируемые процедуры и функции серверных неглобальных общих модулей, у которых установлено свойство **Вызов сервера**.

4.2.3.2. Модуль внешнего соединения

Модуль внешнего соединения расположен, как и модуль приложения, в корневом разделе конфигурации. В нем располагаются процедуры-обработчики событий, которые инициализируются при старте и окончании работы системы в режиме внешнего соединения (COM-соединения).

В модуле внешнего соединения возможно объявление переменных, а также объявление и описание процедур и функций, которые будут доступны для внешнего приложения.

Объекты «1С:Предприятия», доступные извне через COM-соединение:

- экспортируемые процедуры/функции модуля внешнего соединения;
- экспортируемые процедуры/функции общих модулей:
 - включение и исключение модулей целиком выполняются с помощью установки свойств общих модулей;
 - включение и исключение фрагментов общих модулей выполняются с помощью инструкций препроцессора;
- глобальный контекст «1С:Предприятия».

Модуль присутствует только в сессии внешнего соединения. В данном режиме характерно полное отсутствие пользовательского интерфейса.

4.2.3.3. Модуль сеанса

Модулем сеанса называется модуль, который автоматически выполняется при старте системы «1С:Предприятие» в момент загрузки конфигурации.

Модуль сеанса предназначен для инициализации параметров сеанса и обработки действий, связанных с сеансом работы. Этот общий модуль всегда исполняется в привилегированном режиме сервера «1С:Предприятия». Установка параметров сеанса выполняется в обработчике события `УстановкаПараметровСеанса()`.

Модуль сеанса может содержать только определения процедур и функций, может использовать процедуры из общих модулей конфигурации и не содержит экспортируемых процедур и функций.

Вызов обработчика `УстановкаПараметровСеанса()` производится до вызова обработчика события `ПередНачаломРаботыСистемы()` (`ПриНачалеРаботыСистемы()` в случае модуля внешнего соединения).

4.2.3.4. Общие модули

Общие модули располагаются в отдельной ветке дерева метаданных. Основным назначением общих модулей является содержание общих алгоритмов конфигурации, доступных из разных модулей. В общих модулях отсутствует раздел определения переменных и раздел основной программы, то есть они содержат только раздел процедур и функций (см. раздел «Структура программного модуля»).

В любом общем модуле возможно объявление и описание процедур и функций, которые будут доступны в любом модуле конфигурации.

Подробнее про общие модули см. [здесь](#).

4.2.3.5. Модули прикладных объектов

Набор прикладных объектов имеет собственные модули. К таким объектам относятся:

- Менеджеры значения константы,
- Справочники,
- Документы,
- Отчеты,
- Обработки,
- Планы видов характеристик,
- Планы счетов,
- Планы видов расчетов,
- Планы обмена,
- Бизнес-процессы,
- Задачи,
- Регистры.

Модули располагаются в ветках конфигурации, в которых содержатся сами объекты, и являются свойствами объектов. Каждый объект имеет свой индивидуальный модуль. В этих модулях возможно объявление переменных, процедур и функций, которые будут доступны при работе с объектом извне во встроенном языке, дополняя контекст объекта.

В контексте модуля прикладного объекта имеется доступ к реквизитам и табличным частям объекта, а также его методам и событиям.

4.2.3.6. Модули менеджеров объектов

Каждый прикладной объект имеет менеджера, предназначенный для управления этим объектом как объектом конфигурации. С помощью менеджера можно создавать объекты, работать с формами и макетами. Модуль менеджера позволяет расширить функциональность менеджеров за счет введения процедур и функций на встроенном языке. Фактически это позволяет описать методы для объекта конфигурации (например, справочника), которые относятся не к конкретному экземпляру объекта базы данных, а к самому объекту конфигурации.

Контекст модуля менеджера образуется:

- свойствами и методами глобального контекста;
- экспортируемыми процедурами и функциями глобальных общих модулей (если эти модули компилируются на сервере);
- экспортируемыми процедурами и функциями неглобальных общих модулей (если эти модули компилируются на сервере);
- локальным контекстом самого модуля.

Модуль менеджера не может иметь переменных и тела модуля.

Если функции или процедуры модуля менеджера объявлены как экспортируемые, к ним можно будет получить доступ через менеджер объекта.

Например, опишем функцию в модуле менеджера справочника `Контрагенты`:

```
Функция ПолучитьСписокДебиторов() Экспорт
...
КонецФункции
```

[Копировать в буфер обмена](#)

Тогда вызов этой функции из прикладного кода будет выглядеть следующим образом:

```
Дебиторы = Справочники.Контрагенты.ПолучитьСписокДебиторов();
```

[Копировать в буфер обмена](#)

4.2.3.7. Модули форм

Эти модули содержатся в формах конфигурации (см. [здесь](#)). Каждая форма имеет свой индивидуальный модуль. В этих модулях возможно объявление переменных, процедур и функций, которые будут доступны при работе с формой извне во встроенном языке, дополняя контекст формы.

Контекст формы будет образован:

- локальным контекстом самого модуля формы, реквизитами формы, которой «принадлежит» модуль;
- свойствами и методами объекта `УправляемаяФорма` встроенного языка;
- свойствами и методами расширения формы, определяемого типом того объекта, данные которого содержатся в основном реквизите формы;
- глобальным контекстом, в том числе неглобальными общими модулями и экспортируемыми функциями и процедурами глобальных общих модулей, при этом нужно обеспечивать согласованность того, как описана процедура в модуле формы (`&НаКлиенте`, `&НаСервере` и т. д.), и того, какие свойства установлены у общего модуля (`Клиент (управляемое приложение)`, `Сервер` и т. д.);
- экспортируемыми переменными, процедурами и функциями модуля управляемого приложения.

4.2.3.8. Модули команд

Модуль команды предназначен для того, чтобы описать в нем на встроенном языке те действия, которые должна выполнить система при вызове команды. Модуль команды может содержать только описание процедур и функций. Модуль команды не может иметь переменных и тела модуля.

Обработчик `ОбработкаКоманды()` обязательно должен предваряться директивой препроцессора `&НаКлиенте`, т. к. именно там начинается исполнение команды.

Контекст клиентских процедур модуля команды образуется:

- глобальным контекстом, в том числе неглобальными общими модулями и экспортируемыми функциями и процедурами глобальных общих модулей, при этом нужно обеспечивать согласованность того, как описана процедура в модуле формы (`&НаКлиенте`, `&НаСервере` и т. д.), и того, какие свойства установлены у общего модуля (`Клиент (управляемое приложение)`, `Сервер` и т. д.);
- локальным контекстом самого модуля команды.

Контекст серверных процедур модуля команды образуется:

- свойствами и методами глобального контекста;
- экспортируемыми процедурами и функциями глобальных общих модулей, если эти модули компилируются на сервере;
- неглобальными общими модулями, если эти модули компилируются на сервере; доступны экспортируемые методы таких модулей;
- серверными методами модуля команды.

В модуле команды можно описывать методы с ключевым словом `Экспорт`. Однако как-либо использовать их за пределами этого модуля нельзя. Из встроенного языка невозможно получить доступ к командам и, следовательно, к их контексту тоже.

4.2.4. Формат программного модуля

Структуру программного модуля можно подразделить на следующие разделы:

- раздел определения переменных,
- раздел процедур и функций,
- раздел основной программы.

В конкретном программном модуле любой из разделов может отсутствовать.

Раздел определения переменных размещается от начала текста модуля до первого оператора `Процедура`, или оператора `Функция`, или любого исполняемого оператора. В этом разделе могут находиться только операторы объявления переменных `Перем`.

Раздел процедур и функций размещается от первого оператора `Процедура` или оператора `Функция` до любого исполняемого оператора вне тела описания процедур или функций.

Раздел основной программы размещается от первого исполняемого оператора вне тела последней процедуры или функции до конца модуля. В этом разделе могут находиться только исполняемые операторы. Раздел основной программы исполняется в момент инициализации модуля. Обычно в разделе основной программы имеет смысл размещать операторы инициализации переменных какими-либо конкретными значениями, которые необходимо провести до первого вызова любой из процедур или функций модуля.

Исходный текст программного модуля может состоять из операторов и комментариев.

4.2.4.1. Комментарии

Комментарий используется для размещения в исходном тексте программного модуля всякого рода пояснений к работе модуля. Хорошим тоном программирования считается, когда исходный текст содержит исчерпывающий комментарий с описанием алгоритма. В режиме исполнения программы комментарии пропускаются. В тексте программного модуля комментарий начинается парой символов `«//»` и заканчивается концом строки. Это значит, что комментарий можно начинать с начала строки или записывать его после оператора на той же строке. После начала комментария нельзя писать оператор на той же строке, необходимо закончить комментарий концом строки.

```
A=B; // Это - комментарий
```

[Копировать в буфер обмена](#)

// Это тоже комментарий

Комментарий, предшествующий процедуре или функции, может использоваться контекстной подсказкой конфигуратора. Формат такого комментария и схема работы см. [здесь](#).

4.2.4.2. Формат операторов

Операторы имеют вид стандартного обращения к процедуре, за исключением оператора присваивания ($A = B$;) и синтаксических конструкций встроенного языка (например, таких как **Для**, **Пока**, **Если**). Между собой операторы обязательно следует разделять символом точка с запятой. Конец строки не является признаком конца оператора, т. е. операторы могут свободно переходить через строки и продолжаться на другой строке. Можно располагать произвольное число операторов в одной строке, разделяя их символом точка с запятой.

Операторы языка в программном модуле можно подразделить на две категории: операторы объявления переменных и исполняемые операторы.

Операторы объявления переменных создают имена переменных, которыми манипулируют исполняемые операторы.

Любой исполняемый оператор может иметь метку, используемую в качестве точки перехода в операторе **Перейти**.

В общем случае формат оператора языка следующий:

```
~метка:Оператор[{параметры}] [ДобКлючевоеСлово];
```

[Копировать в буфер обмена](#)

В качестве меток используются специальные идентификаторы, начинающиеся с символа тильда и состоящие из последовательности букв, цифр и символов подчеркивание. Чтобы пометить оператор, нужно поместить перед ним метку и следующий за ней символ двоеточие.

[Копировать в буфер обмена](#)

```
~метка:A=B;
```

4.2.4.3. Имена переменных, процедур и функций

Именем переменной, объявленной процедуры или функции может быть любая последовательность букв, цифр и знаков подчеркивания, начинающаяся с буквы или знака подчеркивания. Вновь создаваемые имена не должны совпадать с зарезервированными словами языка или именами свойств, непосредственно доступных в текущем контексте. Распознавание имен переменных, процедур и функций ведется без учета регистра букв.

4.2.4.4. Язык написания программных модулей

Встроенный язык системы «1С:Предприятие» является двуязычным. Почти все зарезервированные слова, имена типов значений, свойств, методов, событий имеют два имени: русское и английское. Исключение составляют слова, не имеющие аналогов в русском языке. В тексте программных модулей эти имена можно свободно смешивать, используя то русские, то английские имена без каких-либо ограничений.

4.2.4.5. Регистры букв при написании программных модулей

Регистр букв (строчные или заглавные) при написании имен переменных, свойств, методов, процедур, функций, а также функций встроенного языка не имеет значения.

4.2.4.6. Зарезервированные слова

Приведенные далее ключевые слова являются зарезервированными и не могут использоваться в качестве создаваемых имен переменных, реквизитов объектов конфигурации и объявляемых процедур и функций. В данном варианте языка каждое из ключевых слов имеет два представления – русское и английское.

Русское имя	Английское имя
Если	If
Тогда	Then
ИначеЕсли	ElsIf
Иначе	Else
КонецЕсли	EndIf
Для	For
Каждого	Each
Из	In
По	To
Пока	While
Цикл	Do
КонецЦикла	EndDo
Процедура	Procedure
Функция	Function
КонецПроцедуры	EndProcedure
КонецФункции	EndFunction
Перем	Var
Перейти	Goto
Возврат	Return
Продолжить	Continue
Прервать	Break
И	And
Или	Or
Не	Not
Попытка	Try
Исключение	Except
ВызватьИсключение	Raise
КонецПопытки	EndTry
Новый	New
Выполнить	Execute

ПРИМЕЧАНИЕ. Регистр букв (строчные или заглавные) при написании не имеет значения.

4.2.5. Специальные символы, используемые в исходном тексте

Символ	Описание

//	Двумя знаками косая черта начинается комментарий. Комментарием считается весь текст от символа до конца текущей строки
	Используется только в строковых константах в начале строки и означает, что данная строка является продолжением предыдущей (перенос строки)
~	Начало метки оператора
:	Окончание метки оператора
;	Символ разделения операторов
()	В круглые скобки заключается список параметров методов, процедур, функций и конструкторов. Также они используются в выражениях встроеного языка
[]	С помощью оператора квадратные скобки производится обращение к свойствам объекта по строковому представлению имени свойства. Также возможно обращение к элементам коллекций по индексу или другому параметру
,	Разделяет параметры в списке параметров методов, процедур, функций и конструкторов
" "	Обрамляет строковые литералы
' '	Обрамляет литералы даты
.	Десятичная точка в числовых литералах. Разделитель, используемый для обращения к свойствам и методам объектов встроеного языка
+	Операция сложения. Операция конкатенации строк
-	Операция вычитания
*	Операция умножения
/	Операция деления
%	Получение остатка от деления. Допускается использование дробных значений делимого и делителя
>	Логическая операция Больше
>=	Логическая операция Больше или равно
<	Логическая операция Меньше
<=	Логическая операция Меньше или равно
=	Операция присваивания. Логическая операция Равно
<>	Логическая операция Не равно

4.3. Примитивные типы данных

Во встроеном языке системы «1С:Предприятие» поддерживается набор примитивных типов данных. Для большинства примитивных типов данных предусмотрена возможность использования в тексте модуля литералов, то есть указание значения соответствующего типа непосредственно в модуле.

[Копировать в буфер обмена](#)

```
// Пример использования литерала типа Строка
A = "Моя строка";
// Пример использования литерала типа Булево
B = Истина;
// Пример использования литерала типа Число
B = 12345.6789;
```

NULL

Описание:

Значения данного типа используются исключительно для определения отсутствующего значения при работе с базой данных, например, при соединении таблиц.

Литералы:

NULL

Булево (Boolean)

Описание:

Значения данного типа имеют два значения – *Истина* и *Ложь*, задаваемых соответствующими литералами. Значения данного типа возвращаются в качестве результата вычисления логических выражений.

Примечание:

В операциях сравнения встроеного языка используются логические выражения. Это означает, что в выражении сравнения не обязательно писать так:

```
Если МояПеременная = Истина Тогда
КонецЕсли;
```

[Копировать в буфер обмена](#)

Достаточно написать так:

```
Если МояПеременная Тогда
КонецЕсли;
```

[Копировать в буфер обмена](#)

Литералы:

- *Истина* (True),
- *Ложь* (False).

Дата (Date)

Описание:

Значения данного типа содержат дату от Рождества Христова (с 01 января 0001 года) и время с точностью до секунды.

Литералы:

Строка цифр, заключенная в одинарные кавычки вида `'ГГГГММДДччммсс'`, где:

- `ГГГГ` – четыре цифры года (включая тысячелетие и век);
- `ММ` – две цифры месяца;
- `ДД` – две цифры даты;
- `чч` – две цифры часа (в 24-часовом формате);
- `мм` – две цифры минут;
- `сс` – две цифры секунд.

Во встроенном языке в литерале типа `Дата` обязательно должно задаваться значение года, месяца и дня. Для задания даты, соответствующей началу отсчета, достаточно указать `'00010101'`. Допускается при указании литералов типа `Дата` опускать последние символы (секунды, минуты, часы и т. д.). Это означает, что данные параметры будут равны нулю (для времени) или единице (для даты).

В литерале даты допускается использование различных разделителей.

Пример:

```
Дата('2008.03.23 10:45:23') = "23.03.2008 10:45:23"
```

[Копировать в буфер обмена](#)

Число (*Number*)

Описание:

Числовым типом может быть представлено любое десятичное число. Определены основные арифметические операции над данными числового типа: сложение, вычитание, умножение и деление.

ВНИМАНИЕ! Максимально допустимая разрядность числа – 32 знака.

Литералы:

Набор цифр, написанных непосредственно в тексте модуля вида:

```
[+|-]{0|1|2|3|4|5|6|7|8|9}[.{0|1|2|3|4|5|6|7|8|9}]
```

В качестве разделителя целой и дробной части используется точка.

Пример:

```
A = 15;
B = -968.612;
```

[Копировать в буфер обмена](#)

Строка (*String*)

Описание:

Значения данного типа содержат строку произвольной длины в формате Unicode. Строка закодирована в формате UTF-16. При необходимости указания в строке суррогатной пары, следует использовать два вызова функции `Символ()` с последовательным указанием значения каждого code unit из суррогатной пары.

Литералы:

Литералы строкового типа представляют собой набор символов, заключенных в кавычки. Для задания в строке символа " (кавычка) необходимо записать две кавычки подряд ("").

Кроме того, допускаются «многострочные» строковые константы. В исходном тексте многострочные константы могут задаваться двумя способами:

- Между фрагментами, представляющими отдельные строки многострочной строки, не должно встречаться никаких символов, за исключением пробелов, переводов строки и строк комментариев.
- Каждая отдельная составляющая не замыкается кавычками, а на каждой последующей строке помещен символ переноса строки «|» (вертикальная черта). В этом варианте комментарии допускаются, если строка начинается с символа комментария «//».

Пример:

```
// Пример строки
МояСтрока = "Это правильная строка";
// Пример 1 многострочной строки
МояМногострочнаяСтрока = "Это
|правильная
|многострочная
|строка";
// Пример 2 многострочной строки
МояМногострочнаяСтрока = "Это тоже" //Это комментарий
"правильная"
"многострочная"
"строка";
// Пример 3 строки с кавычками
НазваниеФирмы = "ООО "Василек" ";
// Пример 4 суррогатная пара
ТекстСсуррогатнойПарой = "Иероглиф: " + Символ(55401) + Символ(56997);
```

[Копировать в буфер обмена](#)

Результат вывода на экран или печать строки `НазваниеФирмы` (пример 3) будет выглядеть следующим образом:

```
ООО "Василек"
```

[Копировать в буфер обмена](#)

Неопределено (*Undefined*)

Описание:

Значение данного типа применяется, когда необходимо использовать пустое значение, не принадлежащее ни к одному другому типу. Например, такое значение изначально имеют реквизиты с составным типом значения. Существует одно-единственное значение данного типа, задаваемое литералом.

Литералы:

```
Неопределено (Undefined)
```

Тип (*Type*)

Значения данного типа используются для идентификации типов значений. Это необходимо для определения и сравнения типов. Данный тип не имеет литералов и возвращается функциями встроенного языка `ТипЗнч` и `Тип` (см. ниже).

4.4. Оператор присваивания

Оператор присваивания (=)

Описание:

Оператор присваивания (символ «=») означает присваивание значения `<Источник>` переменной, обозначенной как `<Назначение>`.

Синтаксис:

```
<Назначение> = <Источник>;
```

Параметры:

`<Назначение>`

В качестве назначения может выступать переменная или свойство объекта встроенного языка, которое допускает запись.

`<Источник>`

Выражение, значение которого необходимо присвоить.

Пример:

```
A = B;
Стр1 = "777";
ДатаДокумента = '20020717';
```

[Копировать в буфер обмена](#)

4.5. Выражения языка

Выражение – это математическая, логическая или строковая формула, состоящая из соответствующих операций, по которой вычисляется значение. Математическое и логическое выражение может стоять справа от знака равенства в операторах присваивания, быть параметром процедур или функций. Логическое выражение также может быть условием в управляющих конструкциях `Если`, `Пока`, `Для`. Выражения состоят из констант, переменных и функций, связанных символами логических и/или арифметических операций.

4.5.1. Арифметические операции

В языке определены следующие виды арифметических операций.

Название	Выражение
Сложение	$(Op1 + Op2)$
Вычитание	$(Op1 - Op2)$
Умножение	$(Op1 * Op2)$
Деление	$(Op1 / Op2)$
Остаток от деления	$(Op1 \% Op2)$
Унарный минус	$(-Op1)$

Арифметические операции имеют один или два операнда, в зависимости от типа которых операция имеет ту или иную семантику. Тот или иной семантический вариант операции определяется по первому операнду. В случае несовпадения типа второго операнда с требуемым значение преобразуется к требуемому типу в соответствии с правилами преобразования типов. Если тип первого операнда не соответствует ни одному из допустимых типов, то в зависимости от ситуации может производиться преобразование типов или возбуждаться состояние ошибки выполнения.

Операция	Описание действия
Сложение определено для следующих типов операндов	<code>Число + Число</code> <code>Дата + Число</code> (к дате прибавляется число секунд)
Вычитание определено для следующих типов операндов	<code>Число - Число</code> <code>Дата - Число</code> (от даты отнимается число секунд) <code>Дата - Дата</code> (результатом является разница между двумя датами, измеренная в секундах)
Умножение	<code>Число * Число</code>
Деление	<code>Число / Число</code>
Остаток от деления	<code>Число % Число</code>

При выполнении арифметических операций с типом `Дата` следует помнить, что результат может иметь точность до 4 знаков после запятой (точность времени – сотни микросекунд). Однако, при выполнении операции записи такого значения в базу данных (в реквизите соответствующего типа), дробная часть числа будет отброшена. В связи с этим рекомендуется округлять до целого значение типа `Число` при выполнении арифметических операций со значением типа `Дата`.

4.5.2. Операция конкатенации

Операция конкатенации («+») используется для того, чтобы присоединить одну строку к другой. Длина результирующей строки равна сумме длин соединяемых строк. В случае несовпадения типа данных второго или последующих операндов со строковым типом их значение преобразуется к строковому типу в соответствии с правилами преобразования типов.

```
ФИО = Фамилия + " " + Имя + " " + Отчество;
```

[Копировать в буфер обмена](#)

4.5.3. Логические операции

Логическая операция сравнивает операнды и вырабатывает значение типа `Булево`: `Истина` или `Ложь`. Существует два вида логических операций: операции сравнения и булевы операции. В операциях сравнения сравниваются два значения. Булевы операции выполняются над значениями типа `Булево`, реализуя булеву алгебру. Символы булевых операций могут комбинироваться, образуя составные операции.

Операции сравнения:

В языке определены следующие виды операций сравнения.

Операция	Выражение операции
Больше	Op1 > Op2
Больше или равно	Op1 >= Op2
Равно	Op1 = Op2
Не равно	Op1 <> Op2
Меньше	Op1 < Op2
Меньше или равно	Op1 <= Op2

Операции сравнения определены для следующих типов операндов.

Операция	Выражение операции
Больше	Число > Число Строка > Строка Дата > Дата
Больше или равно	Число >= Число Строка >= Строка Дата >= Дата
Меньше	Число < Число Строка < Строка Дата < Дата
Меньше или равно	Число <= Число Строка <= Строка Дата <= Дата
Равно	Любой тип = Любой тип
Не равно	Любой тип <> Любой тип

Булевы операции:

В языке определены следующие виды булевых операций.

И (AND)	конъюнкция (булево <i>и</i>)
ИЛИ (OR)	дизъюнкция (булево <i>или</i>)
НЕ (NOT)	логическое отрицание (булево отрицание <i>не</i>)

Логические выражения вычисляются слева направо. Для того чтобы избежать неоднозначности и управлять последовательностью операндов, следует применять круглые скобки.

Уровни старшинства логических операций:

Уровень 1	операнды, заключенные в скобки
Уровень 2	НЕ
Уровень 3	И
Уровень 4	ИЛИ

ПРИМЕЧАНИЕ. При вычислении логического выражения вычисляются только необходимые части выражения. Например, в выражении `(Цена > 0) И ПроверкаСуммы()`, если `Цена <= 0`, функция `ПроверкаСуммы()` не вызывается.

4.5.4. Приоритеты вычисления выражений встроенного языка

В следующей таблице представлен список операторов встроенного языка в порядке увеличения приоритета. Операторы с одинаковым приоритетом вычисляются слева направо.

Операция	Описание
OR/или	Логическое сложение
AND/и	Логическое умножение
NOT/не	Логическое отрицание
<, <=, =, >=, >, <>	Сравнение
+, -	Сложение, вычитание
*, /, %	Умножение, деление
-, +	Унарные плюс и минус
., ()	Разыменованное, вызов метода и т. п.

Для того чтобы избежать неоднозначности и управлять последовательностью операндов, следует применять круглые скобки.

4.6. Операторы и синтаксические конструкции

? (вычислить выражение по условию)

Описание:

Позволяет вычислить одно из двух заданных выражений в зависимости от результата вычисления логического выражения.

Синтаксис:

?(<Логическое выражение>, <Выражение 1>, <Выражение 2>)

Параметры:

<Логическое выражение>

Логическое выражение, результат вычисления которого определяет одно из результирующих выражений, которые будут вычислены. Если результат его вычисления *Истина*, то будет вычисляться <Выражение 1>. Если результат *Ложь*, то <Выражение 2>.

<Выражение 1>

Результирующее выражение, которое будет вычисляться, если результат логического выражения *Истина*.

<Выражение 2>

Результирующее выражение, которое будет вычисляться, если результат логического выражения *Ложь*.

Возвращаемое значение:

Результат вычисления одного из результирующих выражений.

Пример:

```
Статус = ?(ПолучитьСкидку() > 10, "Особый клиент", "Обычный клиент");
Предупреждение(Статус);
```

[Копировать в буфер обмена](#)

ВызватьИсключение (Raise)

Описание:

При использовании данной формы оператора вызывается новое исключение.

Синтаксис:

ВызватьИсключение <Выражение>

Англоязычный синтаксис:

Raise <Expression>

Параметры:

<Выражение>

Результат вычисления выражения преобразуется к строке, и данная строка используется в качестве описания исключения.

Пример:

```
ВызватьИсключение "Документ не может быть проведен";
```

[Копировать в буфер обмена](#)

См. также:

Описание оператора *Попытка*.

Выполнить (Execute)

Описание:

Позволяет выполнить фрагмент кода, который передается ему в качестве строкового значения.

ВНИМАНИЕ! Не рекомендуется реализовывать с помощью этого метода существенную часть функциональности прикладных решений.

ПРИМЕЧАНИЕ. Исполняемый код не может содержать в себе отдельных процедур или функций, т. к. исполнение кода само по себе идет в рамках процедуры или функции, в которой использован этот оператор. А также не может содержать явного объявления переменных.

Синтаксис:

Выполнить (<Строка>)

Англоязычный синтаксис:

Execute(<Строка>)

Параметры:

<Строка>

Строка, содержащая текст исполняемого кода.

Пример:

```
// Выводит в окно сообщений текущую дату
Выполнить("Сообщить(ТекущаяДата())");
```

[Копировать в буфер обмена](#)

ДобавитьОбработчик (AddHandler)

Описание:

Добавляет обработчик события.

При добавлении обработчика события производится проверка соответствия числа параметров события числу параметров метода, назначаемого в качестве обработчика.

Синтаксис:

ДобавитьОбработчик <Событие>, <ОбработчикСобытия>;

Англоязычный синтаксис:

AddHandler <Событие>, <ОбработчикСобытия>;

Параметры:

<Событие>

Событие, которому добавляется обработчик.

Событие задается в форме <Выражение>. <ИмяСобытия>, где:

- <Выражение> – произвольное выражение на встроенном языке. Его результатом должен быть объект, к событию которого добавляется обработчик.
- <ИмяСобытия> – идентификатор (имя) события.

<ОбработчикСобытия>

Процедура/функция-обработчик события.

Обработчиком события может являться метод объекта встроенного языка. Тогда <ОбработчикСобытия> задается как <Выражение>. <ИмяОбработчика>, где:

- <Выражение> – произвольное выражение на встроенном языке. Его результатом должен быть объект, метод которого служит обработчиком события.
- <ИмяОбработчика> – имя метода обработчика события.

Также в качестве обработчика события может быть задана процедура/функция, находящаяся в области видимости. В этом случае обработчик события задается как имя процедуры/функции.

Имеется возможность оформлять подписку на одноименные (в СОМ-объектах) события, но с разным числом параметров. Для этого на встроенном языке необходимо создать несколько обработчиков (каждый с указанием уникального имени и с указанием нужного количества параметров), а механизм подписки сам выберет нужный обработчик для нужной подписки.

Пример:

```
Обработка = Обработки.КонтрольДокумента.Создать();
Накладная = Документы.Накладная.СоздатьДокумент();
ДобавитьОбработчик Накладная.ПриЗаписи, Обработка.ПриЗаписиДокумента;
msword = Новый СОМОбъект("Word.Application");
ДобавитьОбработчик msword.DocumentChange, ПриИзмененииДокумента;
Процедура ПриИзмененииДокумента()
    Сообщить("Документ изменен");
КонечПроцедуры
```

[Копировать в буфер обмена](#)**Пример с разными параметрами:**

```
// Обработчик без параметров
Процедура ОбработкаСобытия()
КонечПроцедуры
// Обработчик с одним параметром
Процедура ОбработкаСобытия2(Параметр)
КонечПроцедуры
// Объект может генерировать события как с параметром,
// так и без параметров
Объект = Новый СОМОбъект("Test.Events");
ДобавитьОбработчик Объект.TestEvent, ОбработкаСобытия
ДобавитьОбработчик Объект.TestEvent, ОбработкаСобытия2
```

[Копировать в буфер обмена](#)**Для (For)****Описание:**

Оператор цикла Для предназначен для циклического повторения операторов, находящихся внутри конструкции Цикл – КонечЦикла. Перед началом выполнения цикла значение <Выражение 1> присваивается переменной <Имя переменной>. Значение <Имя переменной> автоматически увеличивается при каждом проходе цикла. Величина приращения счетчика при каждом выполнении цикла равна 1. Цикл выполняется, пока значение переменной <Имя переменной> меньше или равно значению <Выражение 2>. Условие выполнения цикла всегда проверяется вначале, перед выполнением цикла.

Синтаксис:

Для <Имя переменной> = <Выражение 1> По <Выражение 2> Цикл

// Операторы

[Прервать;]

// Операторы

[Продолжить;]

// Операторы

КонечЦикла;

Англоязычный синтаксис:

For <Имя переменной> = <Выражение 1> To <Выражение 2> Do

// Операторы

[Break;]

// Операторы

[Continue;]

// Операторы

EndDo;

Параметры:

<Имя переменной>

Идентификатор переменной (счетчика цикла), значение которой автоматически увеличивается на 1 при каждом повторении цикла. Так называемый счетчик цикла.

<Выражение 1>

Числовое выражение, которое задает начальное значение, присваиваемое счетчику цикла при первом проходе цикла.

Синтаксическая связка для параметра <Выражение 2>.

<Выражение 2>

Максимальное значение счетчика цикла. Когда переменная <Имя переменной> становится больше чем <Выражение 2>, выполнение оператора цикла **Для** прекращается.

Цикл

Операторы, следующие за ключевым словом **Цикл**, выполняются, пока значение переменной <Имя переменной> меньше или равно значению <Выражение 2>.

// Операторы

Исполняемый оператор или последовательность таких операторов.

Прервать

Позволяет прервать выполнение цикла в любой точке. После выполнения этого оператора управление передается оператору, следующему за ключевым словом **КонецЦикла**.

Продолжить

Немедленно передает управление в начало цикла, где производится вычисление и проверка условий выполнения цикла. Операторы, следующие в теле цикла за ним, на данной итерации обхода не выполняются.

КонецЦикла

Ключевое слово, которое завершает структуру оператора цикла.

Пример:

```
// Перебор дней текущего месяца
ПоследнийДеньМесяца = День(КонецМесяца(ТекущаяДата()));
Для ТекДень = 1 по ПоследнийДеньМесяца Цикл
    Состояние("Обрабатывается день: "+ ТекДень);
    // Операторы обработки очередного дня месяца
    . . .
КонецЦикла;
```

[Копировать в буфер обмена](#)

Для каждого (For each)

Описание:

Оператор цикла **Для каждого** предназначен для циклического обхода коллекций значений. При каждой итерации цикла возвращается новый элемент коллекции. Обход осуществляется до тех пор, пока не будут перебраны все элементы коллекции, или может быть завершен досрочно при выполнении оператора **Прервать**.

Синтаксис:

Для каждого <Имя переменной 1> Из <Имя переменной 2> Цикл

// Операторы

[Прервать;]

// Операторы

[Продолжить;]

// Операторы

КонецЦикла

Англоязычный синтаксис:

For each <Имя переменной 1> In <Имя переменной 2> Do

// Операторы

[Break;]

// Операторы

[Continue;]

// Операторы

EndDo;

Параметры:

<Имя переменной 1>

Переменная, которой при каждом повторении цикла присваивается значение очередного элемента коллекции.

Из

Синтаксическая связка для параметра <Имя переменной 2>.

<Имя переменной 2>

Переменная или выражение, предоставляющее коллекцию. Элементы этой коллекции будут присваиваться параметру <Имя переменной 1>.

Цикл

Операторы, следующие за ключевым словом **Цикл**, выполняются до тех пор, пока не будут перебраны все элементы коллекции.

// Операторы

Исполняемый оператор или последовательность таких операторов.

Прервать

Позволяет прервать выполнение цикла в любой точке. После выполнения этого оператора управление передается оператору, следующему за ключевым словом **КонецЦикла**.

[Продолжить](#)

Немедленно передает управление в начало цикла, где производится вычисление и проверка условий выполнения цикла. Операторы, следующие в теле цикла за ним, на данной итерации обхода не выполняются.

[КонецЦикла](#)

Ключевое слово, которое завершает структуру оператора цикла.

Пример:

[Копировать в буфер обмена](#)

```
// Перебор строк табличной части документа.
Документ = Документы.РасходнаяНакладная.НайтиПоКоду(12345);
// Проверим, найден нужный нам документ или нет
Если Не Документ.Пустая() Тогда
    Для каждого СтрокаСостава из Документ.Состав Цикл
        Состояние("Строка: " + Документ.Состав.Индекс(СтрокаСостава)+1);
        // Операторы обработки очередной строки табличной части
        ...
    КонецЦикла;
КонецЕсли;
```

Если (If)

Описание:

Оператор **Если** управляет выполнением программы, основываясь на результаты одного или более логических выражений. Оператор может содержать любое количество групп операторов, возглавляемых конструкциями **ИначеЕсли** – **Тогда**.

Синтаксис:

Если <Логическое выражение> Тогда

// Операторы

[ИначеЕсли <Логическое выражение> Тогда]

// Операторы

[Иначе]

// Операторы

КонецЕсли;

Англоязычный синтаксис:

If <Логическое выражение> Then

// Операторы

[ElseIf <Логическое выражение> Then]

// Операторы

[Else]

// Операторы

EndIf;

Параметры:

<Логическое выражение>

Логическое выражение.

Тогда

Операторы, следующие за **Тогда**, выполняются, если результатом логического выражения является значение **Истина**.

// Операторы

Исполняемый оператор или последовательность таких операторов.

ИначеЕсли

Логическое выражение, следующее за ключевым словом **ИначеЕсли**, вычисляется только тогда, когда условия в **Если** и всех предшествующих **ИначеЕсли** оказались равны **Ложь**. Операторы, следующие за конструкцией **ИначеЕсли** – **Тогда**, выполняются, если результат логического выражения в данном **ИначеЕсли** равен **Истина**.

Иначе

Операторы, следующие за ключевым словом **Иначе**, выполняются, если результаты логических выражений в конструкции **Если** и всех предшествующих конструкциях **ИначеЕсли** оказались равны **Ложь**.

КонецЕсли

Ключевое слово, которое завершает структуру оператора условного выполнения.

Пример:

[Копировать в буфер обмена](#)

```
Если ДеньНедели(ТекущаяДата()) = 6 Тогда
    Сообщить("Сегодня суббота.");
ИначеЕсли ДеньНедели(ТекущаяДата()) = 7 Тогда
    Сообщить("Сегодня воскресенье.");
Иначе
    Сообщить("Сегодня рабочий день.");
КонецЕсли;
```

Новый (New)

Описание:

Оператор позволяет создать значение указанного типа. Допустим только для тех типов, для которых разрешено создание новых значений. Для прикладных объектов

необходимо использовать функциональную форму оператора **Новый** (вариант 2), так как при проверке модулей в конфигураторе не определены типы для прикладных объектов.

Синтаксис (вариант 1):

```
Новый <Имя типа>{(<Парам 1>, ..., <Парам N>)}
```

Параметры:

Имя типа

Указывается имя типа, значение которого создается.

<Парам 1>, ..., <Парам N>

После имени типа в скобках могут указываться параметры, если они определены в конструкторах для данного типа. Допустимое количество параметров и их назначение указываются в описании конструкторов объекта.

Пример:

```
// Пример создания массива из трех элементов.
Массив = Новый Массив(3);
```

[Копировать в буфер обмена](#)

Синтаксис (вариант 2):

```
Новый (<Тип>[, <Параметры конструктора>])
```

Параметры:

Тип

Имя типа или значение типа *Тип*.

<Параметры конструктора>

Массив параметров конструктора.

Пример:

```
ТипЗначения = Тип("КвалификаторыСтроки");
Параметры = Новый Массив(2);
Параметры[0] = 20;
Параметры[1] = ДопустимаяДлина.Переменная;
КвалифСтр = Новый(ТипЗначения, Параметры);
```

[Копировать в буфер обмена](#)

Перейти (Goto)

Описание:

Безусловная передача управления на другой оператор программы. Передает управление от одного оператора к другому.

Область действия оператора ограничивается программным модулем, процедурой или функцией; он не может передать управление за пределы программного модуля, процедуры или функции.

ПРИМЕЧАНИЕ 1. Метка в этом операторе не должна быть меткой перехода на оператор *Процедура* или *Функция*.

ПРИМЕЧАНИЕ 2. Оператор безусловного перехода не может быть использован для передачи управления на операторы, находящиеся внутри конструкций: *Пока – КонечЦикла*, *Для – КонечЦикла*, *Для каждого – КонечЦикла*, *Если – КонечЕсли*, *Попытка – Исключение – КонечПопытки* **извне** этих конструкций.

Синтаксис:

```
Перейти <Метка>;
```

Пример:

```
Перейти ~Метка1;
...
~ Метка1: Сообщить("Осуществлен переход по метке.");
```

[Копировать в буфер обмена](#)

Перем (Var)

Описание:

Позволяет в явном виде объявить переменную.

Синтаксис:

```
Перем <Имя переменной 1> [Экспорт] [, <Имя переменной 2>, ...];
```

Англоязычный синтаксис:

```
Var <Имя переменной 1> [Export] [, <Имя переменной 2>, ...];
```

Параметры:

<Имя переменной 1>[, <Имя переменной 2>, ...]

Задается имя или имена объявляемых переменных.

Экспорт

Необязательное ключевое слово. Указывает, что данная переменная доступна при обращении к контексту этого модуля из других модулей. Данное ключевое слово необходимо указывать для каждой объявляемой переменной отдельно. Не имеет смысла при объявлении переменных отдельных процедур или функций.

Пример:

```
// Пример объявления одной переменной
Перем А Экспорт;
Перем В;
// Пример объявления нескольких переменных одним оператором
Перем А, В Экспорт;
```

[Копировать в буфер обмена](#)

Неявное объявление переменных:

В языке необязательно объявлять переменные в явном виде. Неявным определением переменной является первое ее появление в левой части оператора присваивания. Тип переменной определяется типом присвоенного ей значения. Не допускается использование в выражениях переменных, не объявленных ранее в явном или неявном виде.

Область использования переменной:

Область использования переменных зависит от места их определения в конфигурации. Существует три области, в которых можно объявить переменные:

- В разделе определения переменных программного модуля управляемого приложения. Это глобальные переменные.
- В разделе определения переменных модуля. Это переменные модуля.
- В процедуре или функции. Это локальные переменные.

Глобальные переменные, объявленные с ключевым словом **Экспорт**, доступны для использования в исполняемых операторах, выражениях, в любой процедуре и функции любого клиентского программного модуля конфигурации.

Переменные модуля доступны для использования в исполняемых операторах, выражениях, в любой процедуре и функции того программного модуля, в пределах которого они объявлены. Если они объявлены с ключевым словом **Экспорт**, то они доступны из других модулей через контекст модуля, в котором они объявлены.

Локальные переменные доступны в пределах той процедуры или функции, в которой они объявлены.

Если переменная определена как глобальная, то она видна из всех процедур и функций любого клиентского программного модуля конфигурации. Если же переменная определена внутри процедуры, то ее областью видимости является данная процедура или функция.

Таким образом, если две переменные с одинаковыми именами используются в двух различных процедурах модуля и имя этой переменной не упоминается как глобальное, то это две различные переменные, локальные для процедур. Если же переменная определена как глобальная переменная, то любое использование имени этой переменной будет приводить к обращению к одной и той же переменной.

Единственный способ создать для процедуры локальную переменную с именем, совпадающим с именем переменной, определенной как глобальная, – это объявить ее явно при помощи оператора **Перем**.

Пока (While)

Описание:

Оператор цикла **Пока** предназначен для циклического повторения операторов, находящихся внутри конструкции **Цикл – КонечЦикла**. Цикл выполняется, пока логическое выражение равно **Истина**. Условие выполнения цикла всегда проверяется вначале, перед выполнением цикла.

Синтаксис:

Пока <Логическое выражение> Цикл

// Операторы

[Прервать;]

// Операторы

[Продолжить;]

// Операторы

КонечЦикла

Англоязычный синтаксис:

While <Логическое выражение> Do

// Операторы

[Break;]

// Операторы

[Continue;]

// Операторы

EndDo;

Параметры:

<Логическое выражение>

Логическое выражение.

Цикл

Операторы, следующие за ключевым словом **Цикл**, выполняются, пока результат логического выражения равен **Истина**.

// Операторы

Исполняемый оператор или последовательность таких операторов.

Прервать

Позволяет прервать выполнение цикла в любой точке. После выполнения этого оператора управление передается оператору, следующему за ключевым словом **КонечЦикла**.

Продолжить

Немедленно передает управление в начало цикла, где производится вычисление и проверка условий выполнения цикла. Операторы, следующие в теле цикла за ним, на данной итерации обхода не выполняются.

КонечЦикла

Ключевое слово, которое завершает структуру оператора цикла.

Пример:

```
ВыборкаДок = Документы.РасходнаяНакладная.Выбрать();
// Цикл по всем документам
```

```
Пока ВыборкаДок.Следующий() Цикл
// Отообразим Документ в панели состояния
Состояние("Обрабатывается документ №" + ВыборкаДок.Номер);
// Операторы выполнения действий над документом
КонечЦикла;
```

Попытка (Try)

Описание:

Оператор **Попытка** управляет выполнением программы, основываясь на возникающих при выполнении модуля ошибочных (исключительных) ситуациях, и определяет обработку этих ситуаций.

В качестве ошибочных (исключительных) ситуаций воспринимаются ошибки времени выполнения модуля. Не предусмотрено определяемых пользователем исключений.

Если при выполнении последовательности операторов попытки произошла ошибка времени выполнения, то выполнение оператора, вызвавшего ошибку, прерывается и управление передается на первый оператор последовательности операторов исключения. При этом управление будет передано даже в том случае, если ошибку вызвал оператор, находящийся в процедуре или функции, вызванной из операторов попытки. Если ошибка произошла в вызванной процедуре или функции, то ее выполнение будет прервано, а локальные переменные – уничтожены. Это справедливо для любой вложенности вызовов. После выполнения последовательности операторов исключения управление передается на следующий за ключевым словом **КонечПопытки** оператор. Если же последовательность операторов попытки выполнена без ошибок, то последовательность операторов исключения будет пропущена и управление также будет продолжено с оператора, следующего за ключевым словом **КонечПопытки**.

Конструкции **Попытка – Исключение – КонечПопытки** могут быть вложенными. При этом при возникновении исключительной ситуации управление передается на тот обработчик, в попытке которого произошла ошибка. Если же в последовательности операторов исключения этого обработчика выполняется оператор **ВызватьИсключение**, выполнение передается вышестоящему обработчику исключения и так далее. Если вышестоящего обработчика нет, исключительная ситуация обрабатывается системно с прекращением выполнения программного модуля.

В выдаче диагностики помощь могут оказать встроенные функции **ОписаниеОшибки()** и **ИнформацияОбОшибке()** (см. описание функций встроенного языка). Также следует учитывать, что попадание во вложенный обработчик исключения приводит к тому, что информация о «родительском» исключении теряется.

Синтаксис:

Попытка

```
// Операторы попытки
```

Исключение

```
// Операторы исключения
```

```
[ВызватьИсключение; ]
```

```
// Операторы исключения
```

```
КонечПопытки;
```

Англоязычный синтаксис:

Try

```
// Операторы попытки
```

Excerpt

```
// Операторы исключения
```

```
[Raise: ]
```

```
// Операторы исключения
```

```
EndTry;
```

Параметры:

```
// Операторы попытки
```

Исполняемый оператор или последовательность таких операторов.

Исключение

Операторы, следующие за ключевым словом **Исключение**, выполняются, если при выполнении последовательности операторов произошла ошибка времени выполнения.

```
// Операторы исключения
```

Исполняемый оператор или последовательность операторов, которые обрабатывают исключительную ситуацию.

ВызватьИсключение

Оператор позволяет вызвать исключение в тех случаях, когда, несмотря на отработку исключительной ситуации, необходимо прервать выполнение модуля с ошибкой времени выполнения. Оператор допустим только внутри операторных скобок **Исключение – КонечПопытки**.

Выполнение данного оператора прекращает выполнение последовательности операторов исключения, и производится поиск более «внешнего» обработчика исключения (при вложенных попытках). Если таковой есть, то управление передается на его первый оператор. Если нет, то исключительная ситуация обрабатывается системно, выдается сообщение о первоначально возникшей ошибке, а выполнение модуля прекращается.

КонечПопытки

Ключевое слово, которое завершает структуру оператора обработки исключительных ситуаций.

Пример:

```
Процедура СформироватьВExcel()
Попытка
// Пытаемся обратиться к программе MS Excel
Табл = Новый ComObject("Excel.Application");
Исключение
Предупреждение(ОписаниеОшибки());
Возврат;
КонечПопытки;
// Операторы формирования отчета
...
КонечПроцедуры
```

[Копировать в буфер обмена](#)

Процедура (Procedure)

Описание:

Ключевое слово **Процедура** начинает секцию исходного текста, выполнение которого можно инициировать из любой точки программного модуля, просто указав **ИмяПроцедуры()** со списком параметров (если параметры не передаются, то круглые скобки, тем не менее, обязательны). Если в модуле приложения или общем программном модуле в теле описания процедуры использовано ключевое слово **Экспорт**, то это означает, что данная процедура является доступной из всех других программных модулей конфигурации.

При выполнении оператора **Возврат** процедура заканчивается и возвращает управление в точку вызова. Если в тексте процедуры не встретился оператор **Возврат**, то после выполнения последнего исполняемого оператора происходит выполнение неявного оператора **Возврат**. Конец программной секции процедуры определяется по оператору **КонецПроцедуры**.

Переменные, объявленные в теле процедуры в разделе **Объявления локальных переменных**, являются локальными переменными данной процедуры, поэтому доступны только в этой процедуре (за исключением случая передачи их как параметров при вызове других процедур, функций или методов).

ПРИМЕЧАНИЕ. Ключевые слова **Процедура**, **КонецПроцедуры** являются не операторами, а операторными скобками, поэтому не должны заканчиваться точкой с запятой (это может приводить к ошибкам выполнения модуля).

Синтаксис:

```
Процедура <ИмяПроцедуры>([[Знач] <Парам 1> [=<ДефЗнач>, ... , [Знач] <Парам N> [=<ДефЗнач>]])[Экспорт]
```

```
// Объявления локальных переменных?
```

```
// Операторы;
```

```
...
```

```
[Возврат;]
```

```
// Операторы;
```

```
...
```

```
КонецПроцедуры
```

Англоязычный синтаксис:

```
Procedure <ИмяПроцедуры>([[Val] <Парам 1> [=<ДефЗнач>, ... , [Val] <Парам N> [=<ДефЗнач>]])[Export]
```

```
// Объявления локальных переменных?
```

```
// Операторы;
```

```
...
```

```
[Return;]
```

```
// Операторы;
```

```
...
```

```
EndProcedure
```

Параметры:

<ИмяПроцедуры>

Назначает имя процедуры.

Знач

Необязательное ключевое слово, которое указывает на то, что следующий за ним параметр передается по значению, т. е. изменение значения формального параметра при выполнении процедуры никак не повлияет на фактический параметр, переданный при вызове процедуры. Если это ключевое слово не указано, то параметр процедуры передается по ссылке, то есть изменение внутри процедуры значения формального параметра приведет к изменению значения соответствующего фактического параметра.

<Парам 1>, ..., <Парам N>

Необязательный список формальных параметров, разделяемых запятыми. Значения формальных параметров должны соответствовать значениям передаваемых при вызове процедуры фактических параметров. В этом списке определяются имена каждого из параметров так, как они используются в тексте процедуры. Список формальных параметров может быть пуст.

=<ДефЗнач>

Необязательная установка значения параметра по умолчанию. Параметры с установленными значениями по умолчанию можно располагать в любом месте списка формальных параметров (подробнее см. раздел «Передача параметров процедур и функций»).

Экспорт

Необязательное ключевое слово, которое указывает на то, что данная процедура является доступной из других программных модулей.

```
// Объявления локальных переменных
```

Объявляются локальные переменные, на которые можно сослаться только в рамках этой процедуры (см. оператор **Перем**).

```
// Операторы
```

Исполняемые операторы процедуры.

Возврат

Необязательное ключевое слово, которое завершает выполнение процедуры и осуществляет возврат в точку программы, из которой было обращение к процедуре. Использование данного оператора в процедуре необязательно.

КонецПроцедуры

Обязательное ключевое слово, обозначающее конец исходного текста процедуры, завершение выполнения процедуры. Возврат в точку, из которой было обращение к процедуре.

Пример:

```
Перем Глоб;
// Описание процедуры
Процедура МояПроцедура(Пар1, Пар2, Пар3) Экспорт
```

```

Глоб = Глоб + Пар1 + Пар2 + Пар3;
Возврат;
КонецПроцедуры
Глоб = 123;
МояПроцедура(5, 6, 7); // Вызов процедуры

```

УдалитьОбработчик (RemoveHandler)

Описание:

Удаляет обработчик события.

При удалении обработчика события производится проверка соответствия числа параметров события числу параметров метода, назначенного в качестве обработчика.

Синтаксис:

```
УдалитьОбработчик <Событие>, <ОбработчикСобытия>;
```

Англоязычный синтаксис:

```
RemoveHandler <Событие>, <ОбработчикСобытия>;
```

Параметры:

<Событие>

Событие, обработчик которого удаляется.

Событие задается в форме <Выражение>.<ИмяСобытия>, где:

- <Выражение> – произвольное выражение на встроенном языке. Его результатом должен быть объект, обработчик события которого удаляется.
- <ИмяСобытия> – идентификатор (имя) события.

<ОбработчикСобытия>

Процедура/функция-обработчик события.

Обработчиком события может являться метод объекта встроенного языка. Тогда <ОбработчикСобытия> задается как <Выражение>.<ИмяОбработчика>, где:

- <Выражение> – произвольное выражение на встроенном языке. Его результатом должен быть объект, метод которого служит обработчиком события.
- <ИмяОбработчика> – имя метода обработчика события.

Также в качестве обработчика события может быть задана процедура/функция, находящаяся в области видимости. В этом случае обработчик события задается как имя процедуры/функции.

Пример:

```
УдалитьОбработчик Накладная.ПриЗаписи, Обработка.ПриЗаписиДокумента;
```

[Копировать в буфер обмена](#)

Функция (Function)

Описание:

Ключевое слово **Функция** начинает секцию исходного текста функции, выполнение которой можно инициировать из любой точки программного модуля, просто указав **ИмяФункции** со списком параметров (если параметры не передаются, то круглые скобки, тем не менее, обязательны). Если в модуле приложения или общем программном модуле в теле описания функции использовано ключевое слово **Экспорт**, то это означает, что данная функция является доступной из всех других программных модулей конфигурации.

Выполнение функции заканчивается оператором **Возврат**. Функции отличаются от процедур только тем, что возвращают **ВозвращаемоеЗначение**. Конец программной секции функции определяется по оператору **КонецФункции**.

Вызов любой функции в тексте программного модуля можно записывать как вызов процедуры, т. е. в языке допускается не принимать от функции возвращаемое значение.

Если ключевое слово **Возврат** в теле функции не указано или строка модуля, его содержащая, не выполнена, то функция возвращает значение типа **Неопределено**.

Переменные, объявленные в теле функции в разделе **Объявления локальных переменных**, являются локальными переменными данной функции, поэтому доступны только в этой функции (за исключением случая передачи их как параметров при вызове других процедур, функций или методов).

ПРИМЕЧАНИЕ. Ключевые слова **Функция**, **КонецФункции** являются не операторами, а операторными скобками, поэтому не должны заканчиваться точкой с запятой (это может приводить к ошибкам выполнения модуля).

Синтаксис:

```
Функция <ИмяФункции>([[Знач] <Парам 1>[=<ДефЗнач>], ... , [Знач] <Парам N>[=<ДефЗнач>]])[Экспорт]
```

```
// Объявления локальных переменных;
```

```
// Операторы;
```

```
...
```

```
Возврат <Возвращаемое значение>;
```

```
// Операторы;
```

```
...
```

```
КонецФункции
```

Англоязычный синтаксис:

```
Function <ИмяФункции>([[Val] <Парам 1>[=<ДефЗнач>], ... , [Val] <Парам N>[=<ДефЗнач>]])[Export]
```

```
// Объявления локальных переменных;
```

```
// Операторы;
```

```
...
```

```
Return <Возвращаемое значение>;
```

```
// Операторы;
```

```
...
```

```
EndFunction
```

Параметры:

```
<ИмяФункции>
```

Назначает имя функции.

```
Знач
```

Необязательное ключевое слово, которое указывает на то, что следующий за ним параметр передается по значению, т. е. изменение значения формального параметра при выполнении функции никак не повлияет на фактический параметр, переданный при вызове функции. Если это ключевое слово не указано, то параметр функции передается по ссылке, то есть изменение внутри функции значения формального параметра приведет к изменению значения соответствующего фактического параметра.

```
<Парам 1>, ..., <Парам N>
```

Необязательный список формальных параметров, разделяемых запятыми. Значения формальных параметров должны соответствовать значениям передаваемых при вызове функции фактических параметров. В этом списке определяются имена каждого из параметров так, как они используются в тексте функции. Список формальных параметров может быть пуст.

```
=<ДефЗнач>
```

Необязательная установка значения параметра по умолчанию. Параметры с установленными значениями по умолчанию можно располагать в любом месте списка формальных параметров (подробнее см. раздел «Передача параметров процедур и функций»).

```
Экспорт
```

Необязательное ключевое слово, которое указывает на то, что данная функция является доступной из других программных модулей.

```
// Объявления локальных переменных
```

Объявляются локальные переменные, на которые можно сослаться только в рамках этой функции (см. оператор [Перем](#)).

```
// Операторы
```

Исполняемые операторы функции.

```
Возврат <Возвращаемое значение>
```

Ключевое слово, которое завершает выполнение функции и возвращает указанное значение в выражение, в котором используется функция.

В качестве возвращаемого значения может выступать выражение или переменная, значение которого содержит результат обращения к функции.

```
КонецФункции
```

Ключевое слово, обозначающее конец исходного текста функции.

Пример:

```
Перем Глоб;
// Описание функции
Функция МояФункция(Пар1, Пар2, Пар3) Экспорт
    Глоб = Глоб + Пар1 + Пар2 + Пар3;
    Возврат Глоб;
КонецФункции
Глоб = 123;
Рез = МояФункция(5, 6, 7); // Вызов функции
```

[Копировать в буфер обмена](#)

4.7. Основные приемы работы

4.7.1. Обращение к свойствам объектов

Помимо основного обращения через точку, в языке системы «1С:Предприятие» предусмотрен механизм обращения к свойствам объектов по строке с именем свойства с помощью оператора `[]` (квадратные скобки).

Свойство объекта ([])

Описание:

Такая конструкция позволяет обращаться к свойствам объектов так же, как это делается через точку с указанием имени свойства.

Синтаксис:

```
<Объект>[<Имя свойства>]
```

Параметры:

```
<Объект>
```

Объект, к свойству которого идет обращение.

```
<Имя свойства>
```

Тип [Строка](#). Имя свойства, к которому необходимо обратиться.

Пример:

```
Спр = Справочники.Номенклатура.НайтиПоКоду(ИскомыйКод);
// Обращение к наименованию справочника по строке с именем свойства
А = Спр["Наименование"];
// Обращение к наименованию справочника по имени свойства
А = Спр.Наименование;
// Оба эти обращения к свойству абсолютно равнозначны
```

[Копировать в буфер обмена](#)

4.7.2. Дополнение контекста объектов и форм

В языке есть возможность при обращении к объектам и формам извне, из других программных модулей, обращаться к переменным, процедурам и функциям этих модулей, как к свойствам и методам самих объектов и форм. Можно обращаться к тем переменным, процедурам и функциям, которые объявлены с ключевым словом [Экспорт](#). Для форм

Пример:

```
// Пример использования процедуры печати документов из журнала
// документов. Допустим, у нас есть несколько различных документов,
// причем у всех есть процедура Печать(). В модуле журнала
// документов располагается кнопка "Печать", которая
// вызывает процедуру печати текущего документа журнала.
Процедура ПечатьНажатие(Элемент)
    // Получим текущий документ, на котором установлен курсор.
    ТекДок = ЭлементыФормы.ЖурналСписок.ТекущаяСтрока;
    // Получим основную форму текущего документа.
    ФрмТекДок = ТекДок.ПолучитьФорму();
    // Вызовем процедуру печати, расположенную
    // в модуле формы документа.
    ФрмТекДок.Печать();
КонецПроцедуры
```

[Копировать в буфер обмена](#)

4.7.3. Передача параметров процедур и функций

Передача параметров процедур и функций выполняется двумя способами. Один способ называется передачей **по ссылке** и представляет собой передачу не конкретного значения параметра, а адреса памяти (ссылки на переменную), где расположено это значение. Изменение переданного значения в вызываемой процедуре или функции приведет к изменению передаваемой переменной в вызывающем методе.

Другой способ называется передачей **по значению** и представляет собой передачу копии значения параметра. В этом случае изменение переданного значения в вызываемой процедуре или функции не приведет к изменению значения передаваемой переменной в вызывающем методе.

В то же время механизм передачи параметров процедур и функций зависит от того, какой вызов будет выполняться:

- вызов без передачи управления между клиентом и сервером (только на клиенте или только на сервере);
- вызов с передачей управления между клиентом и сервером.

Подробнее рассмотрим каждый из способов вызова.

4.7.3.1. Вызов без передачи управления с клиента на сервер

Если вызов происходит без передачи управления между клиентом и сервером (вызов происходит только на клиентской или только на серверной стороне), то по умолчанию параметры передаются по ссылке. При этом изменение формального параметра ведет к изменению соответствующего фактического параметра. С помощью модификатора **Знач** перед именем формального параметра можно указать, что параметр передается по значению. В этом случае нельзя изменить значение фактического параметра путем присвоения формальному параметру какого-либо значения.

[Копировать в буфер обмена](#)

```
&НаКлиенте
Процедура МояПроцедура()
    А = 100;
    ПоСсылке(А);
    // Переменная А = 40, так как в теле процедуры значение
    // параметра Параметр1 изменено на 40.
    // Изменение переменной А произошло потому, что параметр передавался по ссылке
    А = 100;
    ПоЗначению(А);
    // Переменная А = 100, несмотря на то, что в теле процедуры
    // значение параметра Параметр1 изменено на 40.
    // Изменение переменной А не произошло, так как параметр передавался по значению
КонецПроцедуры
&НаКлиенте
Процедура ПоСсылке(Параметр1)
    Параметр1 = 40;
КонецПроцедуры
&НаКлиенте
Процедура ПоЗначению(Знач Параметр1)
    Параметр1 = 40;
КонецПроцедуры
```

Однако необходимо помнить следующую особенность: если параметром передается агрегатный объект, то невозможно присвоить фактическому параметру другое значение, но возможно изменить сам переданный объект. Например, если в процедуру по значению передан массив, то можно очистить этот массив методом **Очистить()**, но нельзя изменить в вызывающей процедуре само значение параметра.

[Копировать в буфер обмена](#)

```
&НаКлиенте
Процедура МояПроцедура()
    Массив = Новый Массив;
    Массив.Добавить(12);
    Массив.Добавить(18);
    // В массиве есть два элемента
    ПоЗначению(Массив);
    // Массив пустой, но это по-прежнему массив, а не Число
КонецПроцедуры
// Параметр передается по значению
&НаКлиенте
Процедура ПоЗначению(Знач Параметр)
    // В массиве два значения
    Параметр.Очистить();
    // В массиве нет значений!
    // Меняем формальный параметр
    Параметр = 14;
    // Изменено значение только формального параметра
КонецПроцедуры
```

Также следует учитывать особенности хранения переменных при организации вызовов:

[Копировать в буфер обмена](#)

```
&НаКлиенте
Процедура ТестоваяПроцедура()
    ЗначениеА = Новый Массив;
    ЗначениеА.Добавить(1);
    ЗначениеА.Добавить(2);
    ЗначениеБ = ЗначениеА;
    // обращение к любой переменной: ЗначениеА или ЗначениеБ вызывает
    // изменение одного и того же массива
    МояПроцедура(ЗначениеА, ЗначениеБ);
    // После завершения вызова процедуры обе переменные (ЗначениеА и ЗначениеБ)
    // ссылаются на один массив.
    // В массиве находятся следующие значения:
    // ЗначениеА[0] = ЗначениеБ[0] = "А"
    // ЗначениеА[1] = ЗначениеБ[1] = "Б"
КонецПроцедуры
&НаКлиенте
Процедура МояПроцедура(Параметр1, Параметр2)
    // Несмотря на то, что параметров два, фактически работа происходит с одним массивом
```

```

Параметр1[0] = 5;
Параметр1[1] = 6;
Параметр2[0] = "А";
Параметр2[1] = "Б";
КонечПроцедуры

```

В данном примере следует обратить внимание на тот факт, что операция присваивания `ЗначениеБ = ЗначениеА;` (в процедуре `ТестоваяПроцедура()`) не приведет к созданию копии массива, который находится в переменной `ЗначениеА`. Обе переменные будут ссылаться на **один и тот же массив!**

При этом в процедуру `МояПроцедура()` параметры передаются по ссылке, следовательно, при изменении двух разных формальных параметров происходит изменение одного физического массива.

4.7.3.2. Вызов с передачей управления с клиента на сервер

Вызов процедур и функций с передачей управления между клиентом и сервером характерен тем, что в общем случае при таком вызове изменяется компьютер, на котором происходит работа вызываемого метода. Это происходит потому, что клиент работает на одном компьютере, а сервер – на другом. Следовательно, нельзя говорить о передаче параметров по ссылке, т. к. один компьютер не может получить прямого доступа к памяти другого компьютера. При работе в файл-серверном варианте клиент и сервер представляют собой один компьютер, но на логику взаимодействия это не влияет. Поэтому передача параметров в случае клиент-серверного взаимодействия происходит особым образом:

- При передаче управления с клиента на сервер (и обратно) **всегда** передаются **копии** параметров. При вызове серверной процедуры или функции с клиента происходит создание копии фактического параметра и передача этой копии на сторону сервера. При возврате управления с сервера на клиента также происходит создание копии формального параметра (с которым происходила работы в вызванной процедуре или функции) для передачи обратно на клиента.
- Если формальный параметр указан с модификатором `Знач`, то значение параметра будет передаваться только при вызове процедуры или функции и не будет передаваться обратно при возврате управления на клиента.
- Если для нескольких формальных параметров указывается одно и то же фактическое значение, то создается столько копий фактического значения, для скольких формальных параметров используется значение.
- Если в качестве формальных параметров указано значение одной и той же переменной, то после возврата управления с сервера значение этой переменной будет установлено в значение самого правого формального параметра (без модификатора `Знач`), который изменялся в вызываемой функции.

[Копировать в буфер обмена](#)

```

&НаКлиенте
Процедура ТестоваяПроцедура(Команда)
    ЗначениеА = Новый Массив;
    ЗначениеА.Добавить(1);
    ЗначениеА.Добавить(2);
    ЗначениеБ = ЗначениеА;
    // обращение к любой переменной: ЗначениеА или ЗначениеБ вызывает
    // изменение одного и того же массива
    // При вызове процедуры создается копия массива для каждого формального параметра
    СервернаяПроцедура(ЗначениеА, ЗначениеБ);
    // После завершения вызова процедуры:
    // 1. Переменные ЗначениеА и ЗначениеБ ссылаются на разные массивы
    // 2. В массивах находятся разные значения
    // ЗначениеА[0] = 5
    // ЗначениеА[1] = 6
    // ЗначениеБ[0] = "А"
    // ЗначениеБ[1] = "Б"
КонечПроцедуры
&НаСервере
Процедура СервернаяПроцедура(Параметр1, Параметр2)
    // Так как передаются копии, для каждого формального параметра на сервере создан "свой" массив
    Параметр1[0] = 5;
    Параметр1[1] = 6;
    Параметр2[0] = "А";
    Параметр2[1] = "Б";
    // С сервера будут возвращаться копии переменных Параметр1 и Параметр2
КонечПроцедуры

```

Следует всегда помнить о том, что при передаче параметров на сервер создается копия значения. Это позволит избежать труднодиагностируемых ошибок при работе системы. Рассмотрим пример:

[Копировать в буфер обмена](#)

```

&НаКлиенте
Процедура ТестоваяПроцедура(Команда)
    ЗначениеА = Новый Массив;
    ЗначениеА.Добавить(1);
    ЗначениеА.Добавить(2);
    ЗначениеБ = ЗначениеА;
    Структура = Новый Структура("Ключ1, Ключ2", ЗначениеА, ЗначениеА);
    СервернаяПроцедура(Структура);
    ЗначениеА[0] = 9;
    ЗначениеА[1] = 8;
    ЗначениеБ[0] = "С";
    ЗначениеБ[1] = "D";
КонечПроцедуры
&НаСервере
Процедура СервернаяПроцедура(Параметр1)
    Параметр1.Ключ1[0] = "0";
    Параметр1.Ключ1[1] = "1";
    Параметр1.Ключ2[0] = "2";
    Параметр1.Ключ2[1] = "3";
КонечПроцедуры

```

После завершения вызова процедуры `ТестоваяПроцедура()` будет наблюдаться следующее состояние переменных:

- Массивы `ЗначениеА` и `ЗначениеБ` идентичны и содержат значения «С» в первом элементе и значение «D» во втором элементе, т. к. переменная `ЗначениеБ` указывает на массив из переменной `ЗначениеА` и значения этих переменных не передавались на сервер.
- В переменной `Структура` будут находиться два массива:
 - Со значениями «0» и «1» для элементов массива, находящегося в элементе структуры с ключом `Ключ1`.
 - Со значениями «2» и «3» для элементов массива, находящегося в элементе структуры с ключом `Ключ2`.
- Такое поведение будет обусловлено тем, что при вызове процедуры происходит создание копии не только самого параметра (переменная `Структура`), но и всех объектов, находящихся в этой структуре: два массива, которые изначально указывали на один массив со значениями элементов «0» и «1».

4.7.3.3. Общие особенности указания параметров

При передаче параметров в метод (процедуру или функцию) различаются две ситуации: когда параметр **опущен** и когда параметр **пропущен**. Пропущенный параметр – это параметр, значение которого явным образом не указано. Если пропускается несколько параметров, то в строке с вызовом метода каждый из пропускаемых параметров должен быть отмечен «своей» разделительной запятой.

[Копировать в буфер обмена](#)

```

Процедура МояПроцедура(Параметр1, Параметр2)
...

```

```

КонецПроцедуры
// Параметр Параметр1 - пропущен
МояПроцедура( ,1);

```

Опущенным может быть только один или несколько параметров, указанных справа в списке формальных параметрах при описании метода. При этом опускать можно только параметры, которые имеют значение по умолчанию. Если опущенными будут параметры, не имеющие значений по умолчанию, то при компиляции данного программного кода будет выдана ошибка о несоответствии числа параметров.

Значение опущенного параметра (одного или нескольких) определяется значениями по умолчанию для опускаемых параметров. Значение пропускаемого параметра равно значению по умолчанию (если таковое задано) или значению **Неопределено**, если для формального параметра не задано значение по умолчанию.

[Копировать в буфер обмена](#)

```

Процедура МояПроцедура(Параметр1, Параметр2 = "по умолчанию", Параметр3 = 16)
...
КонецПроцедуры
// Параметр Параметр1 пропущен, значение параметра = Неопределено
// Параметр Параметр2 указан, значение параметра = 1
МояПроцедура( ,1);
// Параметр Параметр1 пропущен, значение параметра = Неопределено
// Параметр Параметр2 пропущен, значение параметра = "по умолчанию"
// Параметр Параметр3 опущен, значение параметра = 16
МояПроцедура( , );
// Ошибка несоответствия количества параметров
МояПроцедура();

```

Если при вызове метода параметры не передаются (пустой список параметров), то, тем не менее, круглые скобки обязательно требуется ставить.

Не следует передавать с клиента на сервер (и обратно), а также выполнять сериализацию данных, содержащих циклические ссылки. В этом случае будет выдаваться ошибка и сеанс будет завершаться.

При передаче с клиента на сервер и обратно строковых значений следует помнить, что эти значения не должны содержать символы, недопустимые с точки зрения спецификации XML версии 1.0 (<http://www.w3.org/TR/xml/>).

СОВЕТ. На стороне сервера такую проверку можно выполнить с помощью функции [НайтиНедопустимыеСимволыXML\(\)](#).

Для конструкторов различных объектов понятие необязательности параметра трактуется особым образом. Если в описании конструктора указано, что параметр является необязательным, то это означает, что параметр может быть опущен, но не гарантируется, что этот параметр может быть пропущен. Опущенный параметр всегда имеет значение по умолчанию, а пропущенный параметр всегда получит значение **Неопределено**. В этом случае поведение конструктора зависит от реализации. Некоторые конструкторы могут воспринимать значение **Неопределено** как указание использовать некоторое значение по умолчанию, некоторые – воспринимают это значение буквально (с возможным формированием ошибки времени исполнения). Можно сформулировать следующую рекомендацию по использованию необязательных параметров:

- Не рекомендуется пропускать необязательные параметры.
- Если конструктор обладает необязательными параметрами – рекомендуется опускать все необязательные параметры, если не требуется явное указание какого-либо из них. В противном случае рекомендуется явно указывать все необязательные параметры, предшествующие параметру, который вы хотите указать явно.

4.7.4. Работа с коллекциями значений

Ряд объектов в языке системы «1С:Предприятие» представляет собой коллекции значений. Большинство коллекций имеют набор схожих методов и свойств, таких как [Количество\(\)](#), [Индекс\(\)](#), [Добавить\(\)](#), [Удалить\(\)](#) и т. д. В качестве свойств коллекции, как правило, выступают ее элементы. Для коллекций доступен обход элементов коллекции посредством конструкции **Для каждого – Из – Цикл**. Для большинства коллекций доступно обращение к элементам коллекции с помощью оператора [**<Аргумент>**] (квадратные скобки). Как правило, в качестве аргумента передается индекс элемента коллекции. Индексирование элементов коллекции начинается с 0. Это означает, что индекс последнего элемента равен количеству элементов в коллекции минус 1.

Если в процессе обхода коллекции происходит удаление или другие изменения состава элементов, то дальнейшее поведение системы не определено.

Подробнее описание конкретных коллекций, их свойств, методов и приемов работы с ними см. в описаниях конкретных объектов.

4.7.5. Использование номеров и индексов

В языке системы «1С:Предприятие» есть ряд объектов, отдельные части которых имеют нумерацию. К таким объектам, например, относится строка, символы которой имеют номер в строке, или табличный документ, строки и колонки которого имеют номер, и т. п. При обращении к частям объектов обычно используется понятие **Номер**. Номера начинаются с 1.

При обращении к элементам коллекций используется понятие **Индекс**. Индексирование элементов коллекций начинается с 0.

4.7.6. Работа с системными перечислениями

Во встроенном языке системы «1С:Предприятие» существует понятие системных перечислений. Они предназначены для определения некоторого ограниченного набора предопределенных значений. Доступ к системным перечислениям осуществляется как к свойствам глобального контекста его имени. Конкретные значения указываются через точку от имени системного перечисления. Системные перечисления используются, как правило, для задания значений параметров системных методов или свойств объектов, а также в качестве возвращаемых значений методов.

4.7.7. Работа с предопределенными значениями

4.7.7.1. С помощью менеджера объекта

Получить предопределенное значение на стороне сервера «1С:Предприятия» можно с помощью менеджера соответствующего объекта. Строка, определяющая получаемый реквизит, имеет следующий вид:

[Копировать в буфер обмена](#)

```
ТипПредопределенногоЗначения.ИмяОбъектаМетаданных.Значение
```

Рассмотрим составляющие этой строки подробнее:

- **ТипПредопределенногоЗначения** – для получения предопределенных значений доступно указание следующих типов данных (написание во множественном числе):
 - **Справочники**,
 - **ПланыВидовХарактеристик**,
 - **ПланыСчетов**,
 - **ПланыВидовРасчета**,
 - **Перечисления**.
- **ИмяОбъектаМетаданных** – указывается имя объекта метаданных так, как оно задано в конфигураторе.
- **Значение** – может быть одним из следующих:
 - для перечислений указывается имя значения перечисления;
 - для получения предопределенного значения указывается его имя, как оно задано в конфигураторе;

- `ТочкиМаршрута.ИмяТочки` – точка маршрута бизнес-процесса.

В случае, если требуется получить точку маршрута бизнес-процесса, строка, описывающая получаемое значение, будет выглядеть следующим образом:

```
БизнесПроцессы.ИмяОбъектаМетаданных.ТочкиМаршрута.ИмяТочкиМаршрута
```

[Копировать в буфер обмена](#)

Пример:

```
// Получение значения перечисления.
Вид = Перечисления.ВидыТоваров.Товар;
// Получение предопределенных данных справочника.
Элемент = Справочники.Валюта.Рубль;
// Точка маршрута бизнес-процесса
Точка = БизнесПроцесс.Согласование.ТочкиМаршрута.Одобрение;
```

[Копировать в буфер обмена](#)

4.7.7.2. С помощью функции `ПредопределенноеЗначение()`

В связи с тем, что на стороне клиента недоступны прикладные объекты, получение предопределенных реквизитов с помощью менеджеров объектов становится невозможным. Поэтому для их получения существует метод глобального контекста `ПредопределенноеЗначение()`. Параметром этого метода является строка, описывающая то, какое предопределенное значение требуется получить. Синтаксис описания предопределенного значения совпадает с синтаксисом оператора `ЗНАЧЕНИЕ` языка запросов (см. [здесь](#)).

Строка, определяющая получаемый реквизит, имеет следующий вид:

```
ТипПредопределенногоЗначения.ИмяОбъектаМетаданных.Значение
```

[Копировать в буфер обмена](#)

Рассмотрим составляющие этой строки подробнее:

- `ТипПредопределенногоЗначения` – для получения предопределенных значений доступно указание следующих типов данных (написание в единственном числе):
 - `Справочник`,
 - `ПланВидовХарактеристик`,
 - `ПланСчетов`,
 - `ПланВидовРасчета`,
 - `Перечисление`,
 - `БизнесПроцесс`.
- `ИмяОбъектаМетаданных` – указывается имя объекта метаданных так, как оно задано в конфигураторе.
- `Значение` – может быть одним из следующих:
 - для перечислений указывается имя значения перечисления;
 - для получения предопределенного значения указывается его имя, как оно задано в конфигураторе;
 - `ТочкаМаршрута.ИмяТочки` – точка маршрута бизнес-процесса;
 - `ПустаяСсылка` – для получения пустой ссылки.

В случае необходимости получить значение системного перечисления параметр метода будет выглядеть следующим образом:

```
ИмяСистемногоПеречисления.ЗначениеСистемногоПеречисления.
```

Например:

```
ТипДиаграммы = ПредопределенноеЗначение("ТипДиаграммы.ВогнутаяПоверхность");
```

[Копировать в буфер обмена](#)

В случае, если требуется получить точку маршрута бизнес-процесса, строка, описывающая получаемое значение, будет выглядеть следующим образом:

Пример:

```
// Получение значения перечисления.
Вид = ПредопределенноеЗначение("Перечисление.ВидыТоваров.Товар");
// Получение значения пустой ссылки.
ПустаяСсылка =
    ПредопределенноеЗначение("Документ.РасходнаяНакл.ПустаяСсылка");
// Получение предопределенных данных справочника.
Элемент = ПредопределенноеЗначение("Справочник.Валюта.Рубль");
// Точка маршрута бизнес-процесса
Точка = ПредопределенноеЗначение("БизнесПроцесс.Согласование.ТочкаМаршрута.Одобрение");
```

[Копировать в буфер обмена](#)

4.7.8. Синхронные и асинхронные методы работы

4.7.8.1. Общая информация

При работе с объектами, которые реализуют некоторую, достаточно специфичную (работа с файлами, блокирующими окнами, криптографией и внешними компонентами), функциональность, можно выделить два различных подхода: **синхронный** и **асинхронный**. Отличие в этих подходах определяется их названиями.

При синхронном подходе вызов какого-либо метода объекта приведет к тому, что работа вызывающего кода будет остановлена до тех пор, пока не завершится вызов метода объекта (со всеми вложенными вызовами). Обычный вызов процедуры или функции встроенного языка является синхронным вызовом.

При асинхронном подходе вызов метода объекта выполняется как обычно, но управление сразу возвращается вызывающему коду и его исполнение продолжается. При этом вызванный метод гарантирует, что после завершения своей работы он оповестит о своем завершении заинтересованное «лицо» (если такое оповещение требуется). Оповещение выполняется с помощью вызова специального метода, который передается в вызываемый метод объекта. Все вышеописанное верно только для тех методов, для которых заявлена поддержка асинхронной работы.

В системе «1С:Предприятие» асинхронную технику можно использовать для работы:

1. С блокирующими окнами (см. [здесь](#));
2. Расширением работы с файлами (см. [здесь](#));
3. Расширением криптографии (см. [здесь](#));
4. Внешними компонентами (см. [здесь](#)).

Все, указанные выше, варианты предполагают асинхронное использование только на стороне клиентского приложения.

Рассмотрим организацию асинхронного вызова на примере отображения предупреждения. Для асинхронного вывода предупреждения используется метод `ПоказатьПредупреждение()`. Для того, чтобы сообщить о том, что диалог предупреждения закрыт, в этот метод передается **описание оповещения**. Описание оповещения содержит имя и месторасположения процедуры, которая будет вызвана системой после закрытия диалога. Соответственно, метод `ПоказатьПредупреждение()` отличается от метода `Предупреждение()` тем, что в методе `ПоказатьПредупреждение()` существует еще один параметр, через который и передается описание оповещения. Эта разница означает, что метод поддерживает асинхронную работу.

Пример:

```
&НаКлиенте
Процедура ВыполнитьКоманду(Команда)
    ОбратныйВызов = Новый ОписаниеОповещения("ПредупреждениеЗавершение", ЭтотОбъект);
    ПоказатьПредупреждение(ОбратныйВызов, "Закрытие формы обрабатывается отдельно");
КонiecПроцедуры
&НаКлиенте
Процедура ПредупреждениеЗавершение(ДополнительныеПараметры) Экспорт
    // какие-то действия
КонiecПроцедуры
```

Копировать в буфер обмена

Для описания оповещения используется специальный объект `ОписаниеОповещения`. При его создании необходимо указать имя процедуры оповещения, место ее (процедуры) расположения и дополнительные параметры, если необходимо. Дополнительные параметры могут использоваться процедурой обработки оповещения для своей работы. В приведенном выше примере дополнительные параметры не указываются.

Также следует отметить, что процедуры обработки оповещений (обратного вызова) могут располагаться только на стороне клиентского приложения в модуле формы, общем модуле и модуле команды. Процедура обратного вызова должна быть описана с ключевым словом `Экспорт`.

Таким образом, из приведенного примера видно, что работа с асинхронными методами существенно изменяет подходы к разработке. Основное изменение заключается в том, что единый фрагмент текста на встроенном языке (для случаев использования синхронных методов) разбивается на несколько изолированных фрагментов (в случае использования асинхронных методов).

Так, если в приведенном примере после отображения предупреждения должны были выполняться какие-то действия (именно после реакции пользователя, а не после вызова метода!), то эти действия следует перенести в обработчик `ПредупреждениеЗавершение()`.

Более сложные алгоритмы, очевидно, приведут и к более сложному рефакторингу (см. [здесь](#)) исходного текста. Возможно, что переработки потребует сам алгоритм, а не только его реализация, которая может существенно усложниться. Например, в синхронном режиме, если алгоритму требуется какая-либо реакция пользователя, то эту реакцию получают непосредственно в том месте, где эта реакция требуется. Если перерабатывать такой алгоритм с использованием асинхронной техники, то логично разработать отдельный диалог, в котором пользователь ответит на все возможные вопросы, а затем результат этого «опроса» будет передан методу, реализующему алгоритм. При этом вопросы «по месту» больше задавать не требуется, а вместо этого следует анализировать результаты «опроса» пользователя.

Кроме изменения подходов к разработке, при использовании асинхронных методов несколько изменяется и подход к обработке ошибок. Например, ошибки могут возникать во время, когда код на встроенном языке не исполняется и нельзя использовать конструкцию `Попытка ... Исключение`. Для обработки таких ситуаций при создании обработчика оповещения можно указать процедуру, которая будет вызываться системой при возникновении ошибки. Примером такой ошибки может быть ошибка, возникающая в процессе удаления файлов.

Пример:

```
&НаКлиенте
Процедура УдалитьВсеФайлыКаталога(ПутьККаталогу)
    ОбратныйВызов = Новый ОписаниеОповещения("УдалитьЗавершение", ЭтотОбъект, "УдалитьОшибка", ЭтотОбъект);
    НачатьУдалениеФайлов(ОбратныйВызов, ПутьККаталогу, ПолучитьМаскуВсеФайлыКлиента());
КонiecПроцедуры
&НаКлиенте
Процедура УдалитьЗавершение(ДополнительныеПараметры) Экспорт
    Сообщить("Удаление успешно завершено");
КонiecПроцедуры
&НаКлиенте
Процедура УдалитьОшибка(ИнформацияОбОшибке, СтандартнаяОбработка, ДополнительныеПараметры) Экспорт
    Сообщить("При удалении обнаружена ошибка: " + КраткоеПредставлениеОшибки(ИнформацияОбОшибке));
КонiecПроцедуры
```

Копировать в буфер обмена

В приведенной реализации есть одна особенность: если во время удаления файлов произойдет ошибка, то наряду с сообщением об ошибке в окне сообщений платформы, пользователь увидит также и системный диалог с сообщением об ошибке. В том случае, если системный диалог не нужен – следует в обработчике `УдалитьОшибка()` установить параметр `СтандартнаяОбработка` в значение `Ложь`.

Также следует учитывать, что указание обработчика ошибок в описании оповещения игнорируется при использовании асинхронных методов работы с блокирующими окнами (см. [здесь](#)).

Еще одной особенностью работы с асинхронными вызовами является их отладка. В данный момент отладчик умеет отлаживать действия, которые синхронно выполняются в предмете отладки (фактически, по инициативе отладчика).

В случае использования асинхронных вызовов асинхронный вызов выполняется в момент времени, отличный от фактического выполнения строки кода на встроенном языке, которая инициирует данный вызов. В связи с этим, если во время фактического исполнения асинхронного вызова в этом вызове случается ошибка, отладчик не может корректно определить состояние отлаживаемого приложения. Поэтому недоступна текущая строка исполнения, стек вызовов и так далее.

4.7.8.2. Работа в веб-клиенте

Работа в веб-клиенте имеет некоторые особенности. Эти отличия вызваны особенностями реализации веб-браузеров (отсутствие поддержки модальных окон, синхронных вызовов, особенности модели безопасности).

Использование любой техники работы с файлами (см. [здесь](#)) и криптографией (см. [здесь](#)) требует специальных расширений, установленных в используемом веб-браузере. Расширения для работы синхронной и асинхронных техник различаются. При использовании веб-браузера Google Chrome, перед установкой расширений платформы, необходимо выполнить установку расширения веб-браузера [Расширение для работы с 1С:Предприятием](#), которое устанавливается из [Интернет-магазина Chrome](#) (переход на страницу установки будет выполнен автоматически, при необходимости). Без установки этого расширения будет невозможно использование расширений платформы, предназначенных для использования асинхронных способов работы.

При разработке прикладных решений рекомендуется использовать асинхронные способы работы с блокирующими диалогами, файлами, криптографией и внешними компонентами, т. к. использование синхронных методов (даже с использованием соответствующих расширений) может приводить к проблемам в современных веб-браузерах.

Для работы с расширениями необходимо:

- Соответствующим образом настроить веб-браузер (подробнее см. книгу «1С:Предприятие 8.3. Руководство администратора»).
- Расширение работы с файлами:
 - Установить расширение – с помощью метода `НачатьУстановкуРасширенияРаботыСФайлами()` или `УстановитьРасширениеРаботыСФайлами()`. Это интерактивное действие, которое необходимо выполнить один раз для каждого пользователя локального компьютера, использующего расширение.
 - Подключить расширение – с помощью метода `НачатьПодключениеРасширенияРаботыСФайлами()` или `ПодключитьРасширениеРаботыСФайлами()`.
- Расширение работы с криптографией:
 - Установить расширение – с помощью метода `НачатьУстановкуРасширенияРаботыСКриптографией()` или `УстановитьРасширениеРаботыСКриптографией()`. Это интерактивное действие, которое необходимо выполнить один раз для каждого пользователя локального компьютера, использующего расширение.
 - Подключить расширение – с помощью метода `НачатьПодключениеРасширенияРаботыСКриптографией()` или `ПодключитьРасширениеРаботыСКриптографией()`.

- Внешние компоненты – более подробно о работе с внешними компонентами см. [здесь](#).

4.8. Особенности различных вариантов запуска системы

Система «1С:Предприятие» может использоваться в файловом и клиент-серверном вариантах, во внешнем соединении, а также в виде Web-сервисов (см. [здесь](#)).

Конфигуратор позволяет настроить использование процедур и функций общих модулей и модулей объектов для каждого варианта.

4.8.1. Исполнение процедур и функций

Для указания разрешения применения процедур и функций различных модулей (про виды модулей см. [здесь](#)) используют инструкции препроцессору и директивы компиляции.

4.8.1.1. Различие инструкций препроцессора и директив компиляции

Инструкции препроцессора и директивы компиляции предназначены для того, чтобы оставить в скомпилированном модуле только то, что действительно должно присутствовать в том или ином контексте. При этом инструкции препроцессора действуют на исходный текст модуля (т. е. удаляют из модуля текст, который не может там находиться), а директивы компиляции действуют на структурные единицы программного кода (методы – процедуры или функции, а также объявления переменных).

Фактически все программные модули, присутствующие в системе, можно разделить на две большие группы модулей:

- Модули, которые могут существовать (и выполняться) только в одном контексте, например, модуль управляемого приложения может выполняться только на стороне клиента (тонкого или веб-клиента).
- Модули, которые могут существовать (и выполняться) в нескольких контекстах. Такими модулями является модуль управляемой формы, модуль команды и общий модуль. Например, модуль управляемой формы может существовать в четырех экземплярах, при этом в процессе работы управление может передаваться между различными экземплярами модуля:
 - клиентский контекстный модуль,
 - серверный контекстный модуль,
 - клиентский внеконтекстный модуль,
 - серверный внеконтекстный модуль.

При этом у модуля команды и общего модуля не может существовать контекстных экземпляров (в отличие от модуля формы). Поэтому некоторые модули могут компилироваться несколько раз, в зависимости от того, есть ли в этом модуле фрагменты кода, которые могут выполняться в том или ином контексте.

Рассмотрим, какое влияние оказывают директивы компиляции и инструкции препроцессора на исходный программный текст модуля в процессе его преобразования в «исполняемый» код.

Рассмотрим это на примере модуля управляемой формы.

В момент создания формы формируется четыре экземпляра модуля формы и выполняется определение всех необходимых инструкций препроцессору для каждого экземпляра модуля. Затем происходит обработка каждого экземпляра модуля и исключение текста, обрамленного инструкциями препроцессора.

Затем происходит компиляция получившегося программного кода в соответствии с контекстом и директивами компиляции. В результате получается исполняемый модуль.

Предположим, в исходном тексте модуля формы встречается конструкция:

```
&НаКлиенте
Процедура РаботаСФайлами()
    #Если ВебКлиент Тогда
        // программный текст 1
        // выполняется только на веб-клиенте
    #Иначе
        // программный текст 2
        // выполняется на других типах клиента
    #КонецЕсли
КонецПроцедуры
```

[Копировать в буфер обмена](#)

Тогда будет выполнено следующее:

- Данная процедура будет компилироваться на любом клиенте (как следует из директивы компиляции).
- В то же время инструкции препроцессора будут определять, какой именно текст и на каком клиенте будет оставлен в модуле (и как следствие скомпилирован). В данном примере на веб-клиенте будет доступен только «программный текст 1», а на других клиентах – «программный текст 2».

Однако если в модуле формы будет использован следующий код:

```
#Если НаСервере Тогда
&НаКлиенте
Процедура Клиентская()
КонецПроцедуры
#КонецЕсли
```

[Копировать в буфер обмена](#)

Тогда произойдет следующее:

- на сервере будет присутствовать исходный текст нашей процедуры, но он не будет скомпилирован, т. к. директива компиляции `&НаКлиенте` исключает возможность появления процедуры на стороне сервера;
- на клиенте не будет даже исходного текста нашей процедуры, т. к. его «вырежет» инструкция препроцессора, а значит, эта процедура не будет скомпилирована и не будет доступна для вызова.

Исходя из описанного механизма и следует подходить к пониманию взаимодействия директив и инструкций, а также к их совместному применению.

Также нужно отметить, что методы, отмеченные в программном модуле директивами `&НаКлиентеНаСервереБезКонтекста` и `&НаКлиентеНаСервере`, попадут одновременно в разные экземпляры программных модулей.

4.8.1.2. Инструкции препроцессора

Синтаксис инструкций препроцессору следующий:

Инструкция препроцессору

```
#Если <Логическое выражение> Тогда
#ИначеЕсли <Логическое выражение> Тогда
...
#Иначе
```

Логическое выражение

<Логическое выражение> = [НЕ] <Символ препроцессора> [<Булева операция> [НЕ] <Символ препроцессора> [<Булева операция> [НЕ] <Символ препроцессора>]...]

Символ препроцессора

<Символ препроцессора> = { Сервер | НаСервере | Клиент | НаКлиенте | ТонкийКлиент | ВебКлиент | ВнешнееСоединение | ТолстыйКлиентУправляемоеПриложение | То:

Булева операция

<Булева операция> = {И | ИЛИ}

Русское имя	Английское имя
#Если	#If
#Тогда	#Then
#Иначе	#Else
#ИначеЕсли	#ElsIf
#КонецЕсли	#EndIf
И	AND
ИЛИ	OR
НЕ	NOT
Сервер	Server
НаСервере	AtServer
Клиент	Client
НаКлиенте	AtClient
ТонкийКлиент	ThinClient
ВебКлиент	WebClient
ВнешнееСоединение	ExternalConnection
ТолстыйКлиентУправляемоеПриложение	ThickClientManagedApplication
ТолстыйКлиентОбычноеПриложение	ThickClientOrdinaryApplication
МобильноеПриложениеКлиент	MobileAppClient
МобильноеПриложениеСервер	MobileAppServer

Далее приведен перечень инструкций препроцессора и их краткое описание:

- **Сервер, НаСервере** – определяет сервер;
- **Клиент, НаКлиенте** – определяет любого клиента;
- **ТонкийКлиент** – определяет тонкого клиента;
- **ВебКлиент** – определяет веб-клиента;
- **ВнешнееСоединение** – определяет внешнее соединение;
- **ТолстыйКлиентУправляемоеПриложение** – определяет режим управляемого приложения толстого клиента;
- **ТолстыйКлиентОбычноеПриложение** – определяет обычный режим толстого клиента;
- **МобильноеПриложениеКлиент** – определяет клиентскую часть мобильного приложения;
- **МобильноеПриложениеСервер** – определяет серверную часть мобильного приложения.

Далее приведена таблица, указывающая, какие инструкции препроцессора определены в каких режимах работы «1С:Предприятия»:

Режим	Сервер, НаСервере	Клиент, НаКлиенте	ТонкийКлиент	ВебКлиент	ТолстыйКлиентУправляемоеПриложение	ТолстыйКлиентОбычноеПриложение	ВнешнееСое
В клиент-серверном варианте							
На стороне сервера	+						
В толстом клиенте в обычном режиме		+				+	
В толстом клиенте в управляемом режиме		+			+		
В тонком клиенте		+	+				
В веб-клиенте		+		+			
Во внешнем соединении							+
В файл-серверном варианте							
В толстом клиенте в обычном режиме	+	+				+	
В толстом клиенте в управляемом режиме	+	+			+		
В тонком		+	+				

клиенте							
На серверной стороне тонкого клиента	+						
В веб-клиенте	+			+			
Во внешнем соединении							+
В клиенте мобильного приложения		+					
На серверной стороне мобильного приложения	+						

Если используется неглобальный общий модуль, для которого установлено использование на клиенте (любом) и на сервере, то методы, заключенные в условие `#Если Сервер Тогда #КонецЕсли`, будут доступны только в том случае, если вызов этих методов выполняется со стороны сервера. Вызов таких методов со стороны клиента недоступен.

В модуле управляемой формы, выполняемой в файловом варианте в толстом клиенте, будет различаться инструкции `Сервер` и `НаСервере`, а также `Клиент` и `НаКлиенте`. Так, директивы `Сервер` и `Клиент` будут определены и в клиентских и в серверных экземплярах модуля формы, в том время как директива `НаСервере` будет определена только в серверных экземплярах модуля формы, а директива `НаКлиенте` – только в клиентских экземплярах модуля формы. Подробнее о разделении модуля формы на различные экземпляры см. [здесь](#).

В остальных случаях инструкции `Сервер` и `НаСервере` полностью идентичны и инструкции `Клиент` и `НаКлиенте` также полностью идентичны.

ПРИМЕЧАНИЕ. Перед передачей программного модуля на тонкий или веб-клиент сервер выполняет обработку инструкций препроцессора, находящихся в модуле. В ходе обработки текст на встроенном языке, который не исполняется на стороне вышеуказанных клиентов, заменяется на пробелы (т. е. фактически удаляется). При этом символы переноса строк и табуляции остаются на своих местах.

4.8.1.3. Директивы компиляции

Синтаксис директивы компиляции следующий:

Директива компиляции

`&<Директива>`

`<Конструкция языка>`

Директива

`<Директива> = { НаКлиенте | НаСервере | НаСервереБезКонтекста | НаКлиентеНаСервереБезКонтекста | НаКлиентеНаСервере | Область | КонецОбласти }`

Конструкция языка

`<Конструкция языка> = { <Описание переменной> | <Описание процедуры> | <Описание функции> }`

Подробное описание конструкций языка см. [здесь](#).

Русское имя	Английское имя
НаКлиенте	AtClient
НаСервере	AtServer
НаСервереБезКонтекста	AtServerNoContext
НаКлиентеНаСервереБезКонтекста	AtClientAtServerNoContext
НаКлиентеНаСервере	AtClientAtServer
Область	Region
КонецОбласти	EndRegion

Далее приведен перечень директив компиляции и их краткое описание:

- `НаКлиенте` – означает, что метод выполняется на стороне клиента в контексте формы.

Переменная существует все время жизни клиентской части формы.

Из метода доступны клиентские переменные модуля формы.

Допустимы вызовы любых методов.

- `НаСервере` – означает, что метод выполняется на стороне сервера в контексте формы.

Переменная существует только во время вызова выполнения серверного вызова.

Из метода доступны серверные переменные модуля формы.

Допустимы вызовы:

- серверных;
- серверных внеконтекстных;
- клиент-серверных внеконтекстных методов;
- методов неглобальных серверных общих модулей.
- `НаСервереБезКонтекста` – означает, что метод исполняется на сервере вне контекста формы.

Переменные не могут предваряться такой директивой компиляции.

Из метода недоступны переменные модуля формы.

Допустимы вызовы:

- серверных внеконтекстных;
 - клиент-серверных внеконтекстных методов;
 - методов не глобальных серверных общих модулей.
- **НаКлиентеНаСервереБезКонтекста** – означает, что метод выполняется как на клиенте, так и на сервере, вне контекста формы.

Переменные не могут предваряться такой директивой компиляции.

Из метода недоступны переменные модуля формы.

Допустимы вызовы:

- серверных внеконтекстных;
- клиент-серверных внеконтекстных методов;
- методов неглобальных серверных общих модулей;
- методов неглобальных общих модулей с флажками **Сервер** и **Клиент (управляемое приложение)**.

Не поддерживается вызов экспортных процедур, обозначенных такой директивой, как методов объекта **ЭтотОбъект**.

- **НаКлиентеНаСервере** – означает, что методы выполняются на клиенте и на сервере. Переменные не могут предваряться такой директивой компиляции.

Допустимы вызовы:

- серверных внеконтекстных;
- клиент-серверных внеконтекстных методов;
- методов неглобальных серверных общих модулей;
- методов неглобальных общих модулей с флажками **Сервер** и **Клиент (управляемое приложение)**.
- **Область, КонецОбласти** – не предназначены для указания места выполнения кода на встроенном языке. Служат для выделения фрагмента текста, который можно будет свернуть (аналогично процедурам, функциям и т. д.). Подробнее см. [здесь](#).

Ниже приведена таблица, показывающая, какие директивы компиляции доступны в каких модулях системы «1С:Предприятие»:

	Модуль формы	Переменные модуля формы	Модуль команды	Общий модуль
НаКлиенте	+	+	+	+
НаСервере	+	+	+	+
НаСервереБезКонтекста	+			
НаКлиентеНаСервереБезКонтекста	+			
НаКлиентеНаСервере			+	

4.8.2. Особенности использования объектов, их свойств и методов

Каждый объект, метод или свойство встроенного языка (далее в этом разделе – объект) обладает определенной доступностью (см. синтакс-помощник), которая определяет, где можно использовать объект, метод или свойство. Кроме того, в синтакс-помощнике указываются некоторые вспомогательные данные, которые могут помочь разработчику.

Тонкий клиент – указывает, что объект доступен в тонком клиенте.

Веб-клиент – указывает, что объект доступен в веб-клиенте.

Сервер – указывает, что объект доступен на сервере «1С:Предприятия».

Внешнее соединение – указывает, что объект доступен в режиме внешнего соединения.

Толстый клиент – указывает, что объект доступен в толстом клиенте.

Мобильное приложение (клиент) – указывает, что объект доступен в клиентской части мобильной платформы.

Мобильное приложение (сервер) – указывает, что объект доступен на стороне сервера мобильной платформы.

ВНИМАНИЕ! Если для объекта указано, что он недоступен для какого-либо из режимов запуска, то свойства и методы данного объекта также недоступны. Поэтому специального упоминания об этом при описании свойств и методов не приводится.

ПРИМЕЧАНИЕ. Если объект недоступен на стороне какого-либо клиентского приложения, то в этом клиентском приложении также недоступен и тип этого объекта. Например, если в тонком клиенте недоступен объект **СправочникОбъект.Контрагенты**, то **Тип ("СправочникОбъект.Контрагенты")** также недоступен в тонком клиенте.

Сериализуется. Указывает возможность сохранения значения объектов (например, методами **СохранитьЗначение()** и **ЗначениеВФайл()**), сохранения параметров форм отчетов и обработок), а также помещения в **ХранилищеЗначения**.

Данный объект может быть сериализован в/из XML. Указывает возможность поддержки чтения/записи значений данных системы «1С:Предприятие» в/из XML. Подробнее см. [здесь](#).

Возможен обмен с сервером. Указывает возможность обмена значениями данного типа между клиентом и сервером.

ПРИМЕЧАНИЕ. Для управляемого режима запуска обмен между клиентом и сервером возможен для объектов, которые поддерживают XDTO-сериализацию.

Данный объект может быть сериализован в/из XDTO. Указывает на то, что данный тип имеет возможность отображения в модель данных XDTO. При этом указывается квалифицированное имя типа (типов) (указывается URI пространство имен и имя типа), в который отображается данный тип. Например, для типа **ХранилищеЗначения**: (<http://v8.1c.ru/8/data/core>)ValueStorage.

Синтакс-помощник содержит информацию о кешировании результатов работы некоторых методов «1С:Предприятия». Например, для метода глобального контекста **ПредопределенноеЗначение()** будет указана следующая информация: **Результат выполнения кешируется при первом обращении до изменения конфигурации или версии платформы.**

Глава 5. Объекты конфигурации

В этой главе рассказывается о работе с конфигурацией в целом и о тех режимах и механизмах, которые используются для всех объектов конфигурации.

Описание действий по созданию и настройке основных объектов конфигурации (константы, справочники, документы, последовательности, журналы, перечисления, отчеты, обработки, регистры), а также некоторых объектов, располагающихся в ветви конфигурации **Общие** (критерии отбора, стили), производится в документации на примере использования окна редактирования объекта (подробнее см. [здесь](#)). Аналогичные действия можно выполнять и в палитре свойств объектов.

5.1. Свойства конфигурации

Конфигурация в целом имеет свои свойства, которые можно редактировать. Палитра свойств открывается для корневой ветки дерева конфигурации.

Помимо основных свойств (см. [здесь](#)), присущих каждому объекту конфигурации, конфигурация имеет следующие рассмотренные ниже свойства.

5.1.1. Категория свойств «Основные»

Основной режим запуска – выбирается режим запуска системы по умолчанию (**Управляемое приложение** или **Обычное приложение**). Для новой конфигурации устанавливается режим запуска **Управляемое приложение**. Также режим запуска можно изменить для пользователя системы (см. книгу «1С:Предприятие 8.3. Руководство администратора»). Данное свойство нельзя изменить, если свойство **Режим совместимости** установлено в значение **Версия 8.1**.

Назначения использования – указывает назначение использования прикладного решения (**Мобильное устройство** или **Персональный компьютер**). Свойство доступно только в том случае, если свойство **Основной режим запуска** установлено в значение **Управляемое приложение**. В зависимости от выбранного назначения использования, изменяются возможности конфигурирования:

- **Мобильное устройство** – возможна разработка прикладного решения для работы на мобильном устройстве (подробнее см. [здесь](#)). При этом разработчику будет доступен ограниченный набор прикладных объектов (список ограничений см. [здесь](#));
- **Персональный компьютер** – возможна разработка прикладного решения, не предназначенного для функционирования на мобильном устройстве и являющееся собой управляемое приложение;
- Установлены оба флажка (**Мобильное устройство** и **Персональный компьютер**) – возможна разработка прикладного решения, которое будет содержать объекты, доступные и на мобильном устройстве и на персональном компьютере в управляемом приложении.

Следует помнить о следующих особенностях:

- Если установлены оба флажка (**Мобильное устройство** и **Персональный компьютер**), то возможно изменение значения свойства конфигурации **Режим совместимости интерфейса**. Перед изменением будет задан вопрос. Изменение будет выполнено только в случае утвердительного ответа на вопрос.

Значение свойства **Режим совместимости интерфейса** будет изменяться только в том случае, если это свойство установлено в значение **Такси**. Свойство будет установлено в значение **Такси**. [Разрешить 8.2](#).

- Если устанавливается только флажок **Мобильное устройство**, то свойство **Режим совместимости интерфейса** устанавливается в значение **Версия 8.2**.

- Если среди назначений использования отсутствует **Персональный компьютер**, то:

- недоступна возможность изменения основного режима запуска;
- в палитре свойств (см. [здесь](#)) объектов конфигурации недоступны свойства, неиспользуемые на мобильной платформе;
- в качестве типов реквизитов могут выступать только типы объектов, доступные на мобильной платформе.

- Если среди назначений конфигурации присутствует **Мобильное приложение**, то:

- проверка модулей использует настройку для режима **Мобильное приложение**, этот режим, по умолчанию, предлагается для изменения при редактировании параметров;
- проверка конфигурации использует отдельно хранимую настройку выполняемых проверок;
- синтакс-помощник использует отдельно хранимую настройку наложенных отборов.

Вариант встроеного языка – выбирается основной язык программирования (русский или английский). Выбор определяет, на каком языке будут формироваться языковые конструкции в модулях (например, при использовании синтакс-помощника), формироваться имена свойств для объектов, создаваемых платформой в качестве результата работы, а также имена компонентов формы (элементы, команды, реквизиты, параметры) для форм создаваемых платформой (как в режиме «1С:Предприятие», так и в конфигурировании). Вне зависимости от значения свойства можно использовать как русский, так и английский вариант языковых конструкций. При смене значения свойства вариант написания введенных языковых конструкций не изменяется.

Основные роли – список ролей, которые будут использоваться в том случае, когда список пользователей прикладного решения пустой. В этом случае не выполняется авторизация доступа при начале работы системы и права доступа определяются набором ролей (подробнее о правиле сочетания ролей см. [здесь](#)), указанных в свойстве. При этом считается, что пользователь обладает административными правами вне зависимости от значения права **Администрирование** у всех ролей, указанных в качестве основных. Если не указаны основные роли конфигурации и список пользователей пуст, то пользователь работает без ограничения прав доступа. Роли задаются в ветви дерева конфигурации **Общие – Роли** (см. [здесь](#)).

Основной режим запуска, указанный у прикладного решения, должен соответствовать правам на допустимые приложения, указанные у всех ролей, которые выбраны в свойстве **Основные роли**. Так, если свойство **Основной режим запуска** установлено в значение **Управляемое приложение**, а набор основных ролей не предоставляет право **Тонкий клиент**, то пользователь не сможет работать с прикладным решением с помощью тонкого клиента.

Модуль управляемого приложения – по ссылке **Открыть** открывается окно редактирования модуля управляемого приложения (см. [здесь](#)).

Модуль сеанса – по ссылке **Открыть** открывается окно редактирования модуля сеанса (см. [здесь](#)).

Модуль внешнего соединения – по ссылке **Открыть** открывается окно редактирования модуля внешнего соединения (см. [здесь](#)).

Использовать управляемые формы в обычном приложении – указывает необходимость использования управляемых форм в обычном режиме толстого клиента. При установке этого флажка изменяются правила выбора формы толстым клиентом (см. [здесь](#)), а также изменяются правила централизованной проверки конфигурации (см. [здесь](#)).

ПРИМЕЧАНИЕ. Данное свойство доступно только в том случае, если режим редактирования конфигурации установлен в **Управляемое приложение** и **обычное приложение**.

Использовать обычные формы в управляемом приложении – указывает необходимость использования обычных форм в управляемом режиме толстого клиента. При установке этого флажка изменяются правила выбора формы толстым клиентом (подробнее см. [здесь](#)), а также изменяются правила централизованной проверки конфигурации (подробнее см. [здесь](#)).

ПРИМЕЧАНИЕ. Данное свойство доступно только в том случае, если режим редактирования конфигурации установлен в **Управляемое приложение** и **обычное приложение**.

Дополнительные словари полнотекстового поиска – выбор общих макетов или констант, которые будут выступать в роли дополнительных словарей для полнотекстового поиска (см. [здесь](#)).

Требуемые разрешения мобильного приложения – описывает, какие разрешения ОС Android будет требовать мобильное приложение. Фактические разрешения для мобильного приложения устанавливает прикладное решение, выполняющее сборку мобильного приложения. На работу мобильного приложения под ОС iOS данное свойство влияния не оказывает. Установка того или иного разрешения не приводит к включению или отключению использования соответствующего механизма при работе под управлением ОС Android. Поведение мобильного приложения будет зависеть от реакции ОС Android на попытку использования «неразрешенной» возможности.

Хранилище общих настроек – данное хранилище предназначено для хранения различных настроек прикладного решения. Платформа самостоятельно не записывает в данное

хранилище никаких настроек. Хранилище, указанное в данном свойстве, должен использовать разработчик из встроенного языка, для того чтобы выполнять сохранение/восстановление прикладных настроек пользователя.

Хранилище пользовательских настроек – в данное хранилище помещаются пользовательские настройки отчетов.

Хранилище вариантов отчетов – в данное хранилище помещаются варианты отчетов.

Хранилище настроек данных форм – это хранилище сохраняются данные форм. Этим хранилищем можно пользоваться, например, для сохранения реквизитов обработок. При этом можно выбрать индивидуальное хранилище для каждого отчета и обработки.

Хранилище пользовательских настроек динамических списков – в это хранилище сохраняются настройки динамических списков.

Подробнее о хранилищах и работе с ними см. [здесь](#).

5.1.2. Категория свойств «Представление»

Командный интерфейс – по ссылке [Открыть](#) открывается редактор, который позволяет задавать видимость подсистем по умолчанию на начальной странице (в том числе в разрезе ролей).

Рабочая область начальной страницы – по ссылке [Открыть](#) открывается форма настройки, которая позволяет указать, какие формы располагаются на начальной странице и по какому шаблону будет строиться рабочая область.

Командный интерфейс основного раздела – по ссылке [Открыть](#) будет открыт диалог настройки командного интерфейса основного раздела.

Картинка основного раздела – с помощью данного свойства можно изменить картинку основного раздела панели разделов.

Интерфейс клиентского приложения – позволяет настроить расположение панелей по умолчанию для интерфейса [Такси](#). Интерфейс будет принимать такой вид в том случае, когда в редакторе панелей (см. [здесь](#)) пользователь нажмет кнопку [Стандартные](#).

Основной язык – указывается основной язык конфигурации.

Краткая информация – краткая информация о конфигурации.

Подробная информация – подробная информация о конфигурации (допускается использование многострочного текста).

Логотип – выбор логотипа. Выбор осуществляется в стандартном окне выбора картинки:

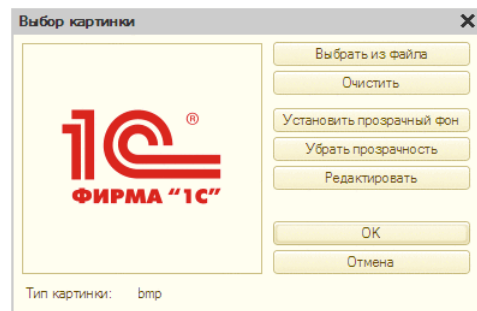


Рис. 68. Выбор картинки

ПРИМЕЧАНИЕ. Размер картинки для логотипа **64x64 точки или меньше**. Тип картинки может быть любым из поддерживаемых системой «1С:Предприятие».

Заставка – выбор заставки. Выбор осуществляется в стандартном окне выбора картинки. Картинка, используемая в качестве заставки, должна иметь размер 600x255 точек. Прозрачность не используется. Тип картинки может быть любым из поддерживаемых системой «1С:Предприятие».

В отдельных случаях предоставляется возможность полной замены заставки, отображаемой при запуске «1С:Предприятия». Для решения по предоставлению такой возможности необходимо обращаться в фирму «1С».

Авторские права – информация об авторе конфигурации.

Адрес информации о поставщике – ссылка на информацию о поставщике конфигурации. Указывается в свойстве [Авторские права](#). Может задаваться как с префиксом схемы ([http://](#)), так и без него.

Адрес информации о конфигурации – ссылка на информацию о конфигурации. Может задаваться как с префиксом схемы ([http://](#)), так и без него. В окне [О программе](#) отображается следующая информация: [Синоним конфигурации](#), свойство [Адрес информации о конфигурации](#), свойство [Авторские права](#), свойство [Адрес информации о поставщике конфигурации](#).

Основная форма констант – выбор основной формы для ввода и редактирования констант конфигурации. Эта форма выбирается из общих форм (см. [здесь](#)) в ветви конфигурации [Общие – Общие формы](#). Подробнее про различные формы см. [здесь](#).

ПРИМЕЧАНИЕ. Данное свойство доступно только в том случае, если режим редактирования конфигурации установлен в [Управляемое приложение и обычное приложение](#).

Свойства [Основная форма отчета](#), [Основная форма настроек отчета](#) и [Основная форма варианта отчета](#) позволяет указать общие формы, которые будут использоваться отчетами, у которых не указаны соответствующие формы. Например, можно создать общую форму отчета, содержащую некоторые возможности, которые должны присутствовать во всех отчетах прикладного решения, например, отправка сформированного отчета по электронной почте. Для решения такой задачи следует создать общую форму отчета, реализовать в этой форме необходимые команды и указать ее в свойстве конфигурации [Основная форма отчета](#). После этого все отчеты, для которых не указана основная форма отчета, будут использовать созданную общую форму.

Свойство [Основная форма настроек динамического списка](#) позволяет указать общую форму, с помощью которой будет выполняться редактирование настроек динамического списка в прикладном решении. Если данное свойство не заполнено – для редактирования настроек динамических списков будет использоваться автоматически генерируемая форма настроек.

Свойство [Основная форма поиска](#) позволяет указать общую форму, которая будет использоваться вместо системной формы полнотекстового поиска, которая может быть вызвана в интерфейсе [Такси](#) из панели инструментов или с помощью сочетания клавиш.

5.1.3. Категория свойств «Разработка»

В свойствах категории описываются данные о поставщике конфигурации и версии конфигурации (подробнее см. [здесь](#)).

Адрес каталога обновлений – содержит адрес ресурса, который может использоваться для обновления прикладного решения.

5.1.4. Категория свойств «Справочная информация»

Включать в содержание справки – если свойство установлено, то содержимое справочной информации включается в общее описание конфигурации.

Справочная информация – по ссылке [Открыть](#) открывается окно редактирования описания конфигурации.

5.1.5. Категория свойств «Совместимость»

Режим управления блокировкой данных – вариант управления блокировкой данных в транзакции (см. [здесь](#)).

Режим автонумерации объектов – определяет, использовать повторно или нет автоматически полученные номера объектов, если они не записаны в базу данных.

Значение данного свойства **ОсвободитьАвтоматически** используется для обеспечения режима работы нумерации, аналогичного версии «1С:Предприятие 8.0». Полученные автоматически номера и коды будут в дальнейшем использоваться, если объект, для которого они получены, не записан.

Значение данного свойства **НеОсвободитьАвтоматически** используется для выбора режима работы, когда для объектов, требующих непрерывной нумерации, будет реализовано получение номеров при записи, а не при открытии формы.

Режим использования модальности – указывает, можно в прикладном решении использовать методы, приводящие к открытию модальных окон или нельзя. Если свойство имеет значение **Использовать** – модальные окна можно использовать без ограничений. Если свойство имеет значение **Не использовать** – модальное окно нельзя использовать в данном прикладном решении. При попытке использования методов будет диагностироваться ошибка (в том числе и при синтаксическом контроле текста на встроенном языке). Вместо модальных окон следует использовать блокирующие окна (см. [здесь](#)). Если свойство имеет значение **Использовать с предупреждением**, то при использовании модальных окон ошибки диагностироваться не будут, но в окно сообщений будет выводиться сообщение о том, что использование модальных окон в данном режиме запрещено. При работе в таком режиме также следует заменить модальные окна блокирующие.

Начиная с «1С:Предприятия» версии 8.3.5 поведение тонкого и толстого клиентских приложений несколько изменяется. Описанное выше поведение (только для толстого и тонкого клиентских приложений) будет наблюдаться только при использовании параметра `/EnableCheckModal` командной строки запуска системы. Если этот параметр не указан, то модальные методы сохраняют работоспособность вне зависимости от свойства конфигурации **Режим использования модальности**. В частности, данное поведение позволяет использовать внешние обработки, написанные без учета свойства конфигурации **Режим использования модальности**, в конфигурациях, которые написаны из расчета на то, что данное свойство будет установлено в значение **Не использовать**.

Если «1С:Предприятие» запускается из конфигуризатора (см. [здесь](#)), то параметр командной строки запуска `/EnableCheckModal` устанавливается принудительно, без возможности настройки такого поведения.

Режим использования синхронных вызовов расширений и внешних компонент – свойство управляет возможностью использовать синхронные вызовы для работы с расширениями работы с файлами, криптографией и внешними компонентами. Если свойство установлено в значение **Использовать**, то на стороне клиента доступны синхронные методы работы с расширениями и внешними компонентами. В том случае, если свойство установлено в значение **Не использовать**, синхронные методы становятся недоступны в синтакс-помощнике, контекстной подсказке при редактировании модулей и расширенная проверка конфигурации выдает ошибку при обнаружении синхронных вызовов на стороне клиента. Вместо синхронных вызовов стоит использовать асинхронные аналоги. В режиме 1С:Предприятие использование синхронных методов работы с расширениями и внешними компонентами приводит в генерации исключения. Режим **Использовать с предупреждениями** является промежуточным вариантом, когда применение синхронных методов допускается, но сопровождается предупреждающими сообщениями в режиме 1С:Предприятие и во время расширенной проверки конфигурации.

При этом описанное поведение будет наблюдаться только в том случае, если используется параметр `/EnableCheckExtensionsAndAddInsSyncCalls` командной строки запуска клиентского приложения.

Если «1С:Предприятие» запускается из конфигуризатора (см. [здесь](#)), то параметр командной строки запуска `/EnableCheckExtensionsAndAddInsSyncCalls` устанавливается принудительно, без возможности настройки такого поведения.

Режим совместимости интерфейса – свойство управляет режимом интерфейса клиентского приложения:

- **Версия 8.2** – клиентское приложение работает в интерфейсе версии 8.2. Переключение в интерфейс **Такси** невозможно.
- **Версия 8.2. Разрешить Такси** – по умолчанию используется интерфейс версии 8.2. Возможно переключение на интерфейс **Такси** с помощью диалога параметров, объект `НастройкиКлиентскогоПриложения` или командной строки.
- **Такси. Разрешить Версия 8.2** – по умолчанию используется интерфейс **Такси**. Возможно переключение на интерфейс версии 8.2 с помощью диалога параметров, объект `НастройкиКлиентскогоПриложения` или командной строки.
- **Такси** – клиентское приложение работает в интерфейсе **Такси**. Переключение в режим интерфейса 8.2 невозможно.

При использовании интерфейса версии 8.2 допустимо переключение между интерфейсом в отдельных окнах и интерфейсом в закладках.

Если свойство **Режим совместимости** установлено в значение, большее значения **Версия 8.3.2** (**Не использовать**, **Версия 8.3.3** и выше), а свойство **Режим совместимости интерфейса** установлено в значение **Версия 8.2** или **Версия 8.2. Разрешить Такси**, то для новых пользователей прикладного решения будет автоматически установлен интерфейс в закладках.

Картинка `БиблиотекаКартинок.Справка` отображается различными способами, если свойство **Режим совместимости интерфейса** установлено в значение **Такси** или **Такси. Разрешить Версия 8.2** и в остальных случаях.

Режим совместимости – свойство управляет поведением механизмов, которое в новой версии системы изменено по сравнению с предыдущими версиями. Данное свойство может принимать значения **Версия 8.3.5**, **Версия 8.3.4**, **Версия 8.3.3**, **Версия 8.3.2**, **Версия 8.3.1**, **Версия 8.2.16**, **Версия 8.2.13**, **Версия 8.1** и **Не использовать**. Особенности работы системы в режиме совместимости с какой-либо версией см. [здесь](#).

Не поддерживается работа конфигурации с неизвестным режимом совместимости. Неизвестными режимами совместимости являются те, которые соответствуют функциональности, реализуемой в последующих версиях системы «1С:Предприятие». Например, если в версии 8.3.1 установить режим совместимости **Не использовать**, то он будет считаться неизвестным при открытии конфигурации в версии 8.2.16. При этом, если в версии 8.3.1 установить режим совместимости **Версия 8.2.16**, то при открытии конфигурации в версии 8.2.16 он будет отображаться как **Не использовать**. Если в конкретной версии не будет вводиться нового режима совместимости, то действие режима **Не использовать** будет аналогично предыдущей версии. При попытке запустить или загрузить конфигурацию с неизвестным режимом совместимости, будет выдаваться ошибка с указанием требуемой версии. Запрещена загрузка файлов `1cv8.dt`, сформированных в версии 8.3.1 и выше, в «1С:Предприятии» младших версий (младше чем 8.3.1). Исключением является ситуация, когда в версии 8.3.1 свойство конфигурации **Режим совместимости** установлено в значение **Версия 8.2.16**.

При конвертации конфигураций версии «1С:Предприятие 8.1» (и более ранних) свойство принимает значение **Версия 8.1**. В общем случае можно сказать, что при открытии конфигурацией более поздней версией «1С:Предприятия», свойство **Режим совместимости** будет устанавливаться в значение совместимости с предыдущей версией, если в новой версии введен такой режим.

Если необходимо обеспечить работу прикладного решения сразу в нескольких версиях «1С:Предприятия» (в том числе и для которых есть режим совместимости), то в местах различного вызова кода рекомендуется вначале получать текущую версию платформы и затем сравнивать ее с режимом совместимости (если это необходимо). При этом не рекомендуется проверять режим совместимости на значение **НеИспользовать**.

Режим совместимости мобильной платформы (см. [здесь](#)) имеет несколько отличий:

- Режим совместимости для мобильной платформы не управляется свойствами конфигурации (в том числе и свойством **Режим совместимости**). Однако данное свойство существует «внутри» мобильного приложения.
- Установка значения режима совместимости происходит в момент создания информационной базы. Значение режима совместимости будет соответствовать версии мобильной платформы, с помощью которой выполнялась операция создания информационной базы.
- Обновление версии мобильной платформы не приводит к обновлению значения режима совместимости.
- При обновлении конфигурации информационной базы мобильного приложения, режим совместимости устанавливается в значение, равное версии мобильной платформы, с помощью которой выполняется обновление конфигурации.
- Соответственно, все методы встроенного языка, поведение которых учитывает режим совместимости, будут учитывать режим совместимости, сформированный по вышеизложенным правилам.

5.2. Модуль управляемого приложения

Более подробное описание модуля управляемого приложения см. [здесь](#).

Модуль управляемого приложения, являясь частью конфигурации, сохраняется только в составе конфигурации. Использование пункта **Файл – Сохранить** приведет к выполнению процедуры сохранения сделанных изменений применительно ко всей конфигурации.

5.3. Модуль внешнего соединения

Более подробное описание модуля внешнего соединения см. [здесь](#).

5.4. Модуль сеанса

Более подробное описание модуля сеанса см. [здесь](#).

5.5. Ветвь конфигурации «Общие»

В этом разделе описываются такие объекты конфигурации, как **Подсистемы**, **Общие модули**, **Параметры сеанса**, **Роли**, **Планы обмена**, **Критерии отбора**, **Подписки на события**, **Регламентные задания**, **Функциональные опции**, **Параметры функциональных опций**, **Хранилища настроек**, **Общие формы**, **Общие команды**, **Группы команд**, **Интерфейсы**, **Общие макеты**, **Общие картинки**, **пакеты XDTO**, **Web-сервисы**, **WS-ссылки**, **Стили**, **Языки**. Эти объекты не описывают структуру данных и механизмы их обработки. Они предназначены для установки правил работы пользователей с данными, описания вспомогательных объектов, используемых для формирования различных форм, в механизме обмена данными, а также содержат общие модули и макеты печатных форм, доступные из любого модуля конфигурации.

5.5.1. Подсистемы

Описание назначения подсистем см. [здесь](#).

ПРИМЕЧАНИЕ. Не рекомендуется задавать подсистеме имя **Desktop**.

На структуру объектов ветви **Подсистемы** не накладывается никаких ограничений по числу и вложенности.

Для просмотра объектов конфигурации, относящихся к определенному набору подсистем, в окне **Конфигурация** можно произвести настройку фильтра отбора объектов. Следует выбрать пункт **Действия – По подсистемам** окна **Конфигурация** и указать необходимый набор подсистем, а также установить дополнительные признаки отбора **Включать объекты подчиненных подсистем** и **Включать объекты родительских подсистем**.

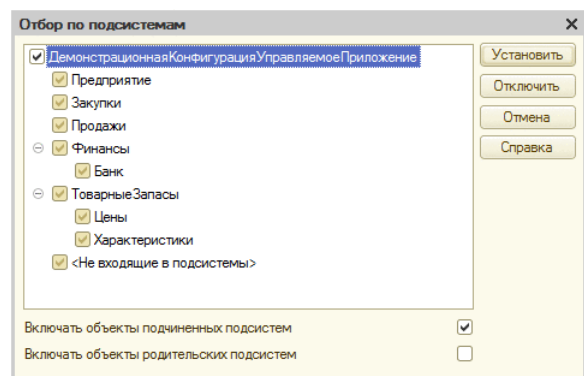


Рис. 69. Отбор по подсистемам

В списке подсистем существует специальный пункт **<Не входящие в подсистемы>**, с помощью которого можно отобразить только те объекты, которые не принадлежат ни одной подсистеме.

Для быстрого отбора объектов, входящих в конкретную подсистему, можно использовать специальную команду контекстного меню: **Объекты подсистемы**. Выполнение данной команды эквивалентно исполнению команды **Действия – По подсистемам** с указанием только одной подсистемы в открывшемся диалоге.

ПРИМЕЧАНИЕ. При установке отбора по подсистемам не показываются ключевые ветви дерева объектов конфигурации, в которых нет отобранных объектов.

Принадлежность объектов конфигурации к определенной подсистеме определяет пользовательский интерфейс.

Значение свойства объекта конфигурации **Подсистемы** доступно в программе с помощью средств встроенного языка. Это предоставляет дополнительные возможности отбора данных.

Команда контекстного меню **Переместить подсистему** позволяет изменить подчиненность подсистемы в иерархии подсистем.

Для того чтобы «привязать» объекты метаданных к конкретной подсистеме (подсистемам), служит закладка **Состав** редактора подсистемы.

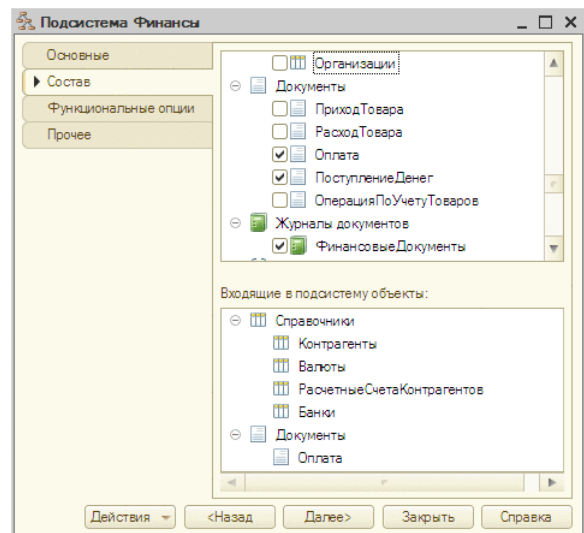


Рис. 70. Состав подсистемы

В верхней части окна перечислены все объекты конфигурации, которые могут быть отнесены к различным подсистемам. Отметка объекта галочкой (например, **Организации** на [рис. 70](#)) означает, что объект начинает входить в подсистему и появляется в нижней части окна. В нижней части окна перечислены все объекты, которые входят в редактируемую подсистему.

Если свойство **Включать в содержание справки** установлено, то в содержание справки будет включена ветвь, отображающая справку по подсистеме и всем объектам, входящим в ее состав. Если свойство сброшено, то такая ветвь (описывающая подсистему и объекты, входящие в ее состав) в содержании справки будет отсутствовать, но справка по объектам, входящим в подсистему, будет доступна непосредственно в формах самих объектов.

5.5.2. Общие модули

Объекты, расположенные в ветви дерева конфигурации **Общие модули**, предназначены для размещения в них текстов функций и процедур, которые могут вызываться из любого другого модуля конфигурации.

ВНИМАНИЕ! Общий модуль может содержать только определения процедур и функций.

Процедуры и функции общего модуля, для которых в заголовках указано ключевое слово **Экспорт**, являются одними из составных частей **глобального контекста**. Подробнее о написании процедур в общем модуле можно узнать в разделах «Формат исходных текстов программных модулей» и «Операторы» справки по встроенному языку.

Для редактирования общего модуля необходимо в палитре свойств объекта типа **Общие модули** окна **Конфигурация** в свойстве **Модуль** щелкнуть мышью ссылку **Открыть**. Текст общего модуля будет выдан для редактирования в редакторе текстов системы «1С:Предприятие» в режиме редактирования текста программного модуля.

Общий модуль, являясь частью конфигурации, сохраняется только в составе конфигурации.

Свойство **Глобальный** определяет, являются ли экспортируемые методы общего модуля частью глобального контекста.

Если свойство **Глобальный** установлено в значение **Истина**, то экспортируемые методы общего модуля доступны как методы глобального контекста.

Если свойство **Глобальный** установлено в значение **Ложь**, то в глобальном контексте создается свойство с именем, соответствующим имени общего модуля в метаданных. Данное свойство доступно только для чтения. Значением данного свойства является объект **ОбщийМодуль**. Через данный объект доступны экспортируемые методы данного общего модуля. Таким образом, обращение к методам неглобальных общих модулей выглядит как **xxxxx.yyyyy**, где **xxxxx** – это имя свойства, соответствующее контексту общего модуля, а **yyyyy** – имя экспортируемого метода общего модуля.

Пример:

```
РаботаСТорговымОборудованием.ПодключитьСканерШтрихкодов();
```

[Копировать в буфер обмена](#)

5.5.2.1. Различные контексты и общие модули

С помощью свойств общих модулей и инструкций препроцессора можно организовать выполнение различных методов общих модулей в нужном контексте.

Каждое свойство общего модуля отвечает за возможность компиляции (и исполнения) общего модуля в том или ином контексте.

Доступны следующие свойства, отвечающие за контекст, в котором доступны методы общего модуля:

- **Клиент (обычное приложение)** – методы общего модуля будут доступны для толстого клиента в режиме обычного приложения;
- **Клиент (управляемое приложение)** – методы общего модуля будут доступны для тонкого клиента, веб-клиента, а также для толстого клиента в режиме управляемого приложения;
- **Сервер** – методы общего модуля будут доступны на сервере;
- **Внешнее соединение** – методы общего модуля будут доступны во внешнем соединении.

Если устанавливаются одновременно несколько свойств, то это означает, что методы общего модуля будут доступны в нескольких контекстах.

Если у общего модуля установлено свойство **Сервер** и еще какое-либо свойство, то это означает, что общий модуль будет доступен одновременно на сервере и в выбранном клиенте. При этом необходимо понимать, что фактически это будет несколько вариантов скомпилированного кода (по числу выбранных клиентов и собственно для сервера).

При этом если метод, расположенный в таком общем модуле, вызывается со стороны клиента, то будет использована клиентская копия общего модуля, а если с сервера – серверная. В этом случае с помощью директив препроцессора (подробнее см. [здесь](#)) можно «оградить» сервер от того кода, который на нем не может исполняться.

Рассмотрим пример. В общем модуле (который может исполняться на тонком клиенте и на сервере) есть метод, который имеет несколько различное поведение на стороне тонкого клиента и на стороне сервера. Посмотрим, как это можно сделать:

```
Процедура МетодОбщегоМодуля() Экспорт
    // Тут размещается различный важный код
#Если ТонкийКлиент Тогда
    // Покажем предупреждение
    ПоказатьОповещениеПользователя("На клиенте");
#КонецЕсли
КонецПроцедуры
```

[Копировать в буфер обмена](#)

Тогда на стороне сервера код приобретет следующий вид:

```
Процедура МетодОбщегоМодуля() Экспорт
    // Тут размещается различный важный код
КонецПроцедуры
```

[Копировать в буфер обмена](#)

А на стороне тонкого клиента код будет иметь следующий вид:

```
Процедура МетодОбщегоМодуля() Экспорт
    // Тут размещается различный важный код
    // Покажем предупреждение
    ПоказатьОповещениеПользователя("На клиенте");
КонецПроцедуры
```

[Копировать в буфер обмена](#)

Для передачи управления с клиента на сервер существует несколько способов:

- вызвать метод серверного общего модуля;
- в модуле формы или команды вызвать метод, который предваряется директивами компиляции **&НаСервере**, **&НаСервереБезКонтекста** (подробнее о модуле формы см. [здесь](#)).

При этом из серверных процедур невозможно вызвать методы клиентских общих модулей (у которых не установлено свойство **Сервер**) и клиентские методы модуля формы или модуля команды. Управление вернется на клиента после того, как будет завершен самый внешний вызов серверного метода.

Исключение составляют методы модуля формы и модуля команды, которые предваряются директивами компиляции **&НаКлиентеНаСервере**, **&НаКлиентеНаСервереБезКонтекста** (подробнее см. [здесь](#)).

Также следует упомянуть следующие моменты:

- Если общий модуль доступен более чем для одного клиента, то при написании программного кода следует учитывать максимальные ограничения, которые могут накладываться клиентами, либо использовать инструкции препроцессора для «изоляции» кода, специфичного для того или иного клиента.
- Инструкции препроцессора также имеют смысл тогда, когда один общий модуль имеет несколько контекстов исполнения, например, внешнее соединение и тонкий клиент

или (что встречается значительно чаще) какой-либо клиент и сервер. В этом случае инструкции препроцессора будут обрамлять интерактивный код, который невозможно использовать на сервере, но возможно на клиенте (см. пример выше).

Подробнее об инструкциях препроцессора и директивах компиляции см. [здесь](#) и в разделе «Исполнение процедур и функций» справки по встроенному языку.

Свойство **Вызов сервера** предназначено для управления возможностью вызова экспортируемых методов серверного общего модуля из клиентского кода. Если свойство установлено, то экспортируемые методы серверного общего модуля доступны для вызова со стороны клиента. Если свойство не установлено, то такие экспортируемые методы можно вызывать только из серверных методов (как методов серверных общих модулей, так и серверных методов модуля формы и модулей команд).

СОВЕТ. Рекомендуется устанавливать в значение *Ложь* свойство **Вызов сервера** в тех случаях, когда серверный общий модуль содержит методы, которые нежелательно вызывать с клиента (например, по причинам безопасности).

ПРИМЕЧАНИЕ. Если одновременно установлены свойства **Клиент (обычное приложение)**, **Клиент (управляемое приложение)**, **Внешнее соединение**, то свойство **Вызов сервера** автоматически сбрасывается. Если устанавливается свойство **Вызов сервера**, то автоматически сбрасываются свойства **Клиент (обычное приложение)**, **Клиент (управляемое приложение)** и **Внешнее соединение**, если эти свойства были установлены одновременно.

Свойство **Привилегированный** предназначено для отключения контроля прав доступа при выполнении методов общего модуля.

ПРИМЕЧАНИЕ. Если свойство **Привилегированный** установлено, то общему модулю автоматически устанавливается свойство **Сервер** и сбрасываются остальные свойства (**Клиент (обычное приложение)**, **Клиент (управляемое приложение)** и **Внешнее соединение**). Привилегированный общий модуль может исполняться только на сервере.

Подробнее о привилегированном режиме см. [здесь](#).

5.5.2.2. Повторное использование возвращаемых значений

Если общий модуль не является глобальным, то становится доступно свойство **Повторное использование возвращаемых значений**. Это свойство может принимать следующие значения:

- **Не использовать** – повторное использование возвращаемых значений для функций этого общего модуля не используется.
- **На время вызова** и **На время сеанса** – для общего модуля используется метод определения повторного использования данных. Суть этого метода заключается в том, что в ходе выполнения кода система запоминает параметры и результат работы функций после первого вызова функции. При повторном вызове функции с такими же параметрами, происходит возврат запомненного значения (из первого вызова) без выполнения самой функции. Если функция во время своего выполнения меняет значения параметров, то повторный вызов функции не будет это делать.

Можно выделить следующие особенности сохранения результатов вызова:

- если функция выполняется на сервере и вызывается из серверного кода, то значения параметров и результат вызова запоминаются для текущего сеанса на стороне сервера;
- если функция выполняется на толстом или тонком клиенте, то значения параметров и результатов вызова запоминаются на стороне клиента;
- если функция выполняется на стороне сервера, а вызывается из клиентского кода, то значения параметров вызова запоминаются и на стороне клиента, и на стороне сервера (для текущего сеанса).

Сохраненные значения удаляются:

- если свойство установлено в значение **На время вызова**:
 - на стороне сервера – при возврате управления с сервера;
 - на стороне клиента – при завершении работы процедуры или функции встроенного языка верхнего уровня (вызванной системой из интерфейса, а не из другой процедуры или функции встроенного языка);
- если свойство общего модуля установлено в значение **На время сеанса**:
 - на стороне сервера – при окончании сеанса;
 - на стороне клиента – при закрытии клиентского приложения.

Сохраненные значения будут удалены:

- на сервере, в толстом клиенте, во внешнем соединении, в тонком клиенте и в веб-клиенте с обычной скоростью соединения – через 20 минут после вычисления сохраняемого значения или через 6 минут после последнего использования;
- в тонком клиенте и веб-клиенте с низкой скоростью соединения – через 20 минут после вычисления сохраняемого значения;
- при нехватке оперативной памяти в рабочем процессе сервера;
- при перезапуске рабочего процесса;
- при переключении клиента на другой рабочий процесс.

После удаления значений вызов экспортируемой функции выполняется как при первом вызове.

На выполнение процедур данное свойство общих модулей не влияет – процедуры выполняются всегда.

Если у общего модуля установлено повторное использование возвращаемых значений, то на типы параметров экспортируемых функции накладывается ряд ограничений. Типы параметров могут быть только:

- Прimitives типами (**Неопределено**, **NULL**, **Булево**, **Число**, **Строка**, **Дата**).
- Любыми ссылками на объекты базы данных.
- Структурами со значениями свойств вышперечисленных типов. В этом случае идентичность параметров контролируется «по содержимому» структур.

Если у общего модуля свойство **Повторное использование возвращаемых значений** установлено в значение **На время сеанса**, то в значениях, возвращаемых функциями такого модуля, нельзя использовать значения типа **МенеджерВременныхТаблиц**.

Если функция общего модуля, с установленным повторным использованием, вызывается из этого же самого общего модуля (например, с именем **ОбщийМодуль**), то следует помнить о следующей особенности: если функция вызывается по имени **МояФункция()**, то исполнение функции будет происходить при каждом вызове функции. Для того чтобы использовались сохраненные значения, функция следует вызывать по полному имени: **ОбщийМодуль.МояФункция()**.

Метод глобального контекста **ОбновитьПовторноИспользуемыеЗначения()** удаляет все повторно используемые значения, как на стороне сервера, так и на стороне клиента, независимо от места вызова метода. После выполнения метода **ОбновитьПовторноИспользуемыеЗначения()** первый вызов функции будет выполнен полностью.

5.5.3. Параметры сеанса

Параметры сеанса предназначены в основном для использования значений параметров в запросах и условиях ограничения доступа к данным для текущего сеанса.

Использование параметров сеанса снижает время доступа к данным за счет исключения связанных таблиц.

Настройка параметров сеанса производится в палитре свойств.

Для каждого параметра сеанса определены два права доступа – *Получение* и *Установка* (подробнее о правах см. раздел ниже). Если право *Установка* снято, то инициализация данного параметра сеанса возможна только в общем модуле с установленным свойством *Привилегированный* или в модуле сеанса.

Инициализация параметров сеанса может выполняться в модуле сеанса, в обработчике события `УстановкаПараметровСеанса()` (см. [здесь](#)).

До инициализации параметр сеанса находится в состоянии *Не установлено*. При попытке чтения такого параметра сначала вызывается обработчик события `УстановкаПараметровСеанса()`. Если после вызова состояние параметра остается *Не установлено*, то вызывается исключение.

Следует разделять области применения параметров сеанса и глобальных переменных модуля управляемого приложения (модуля внешнего соединения). Среди основных отличий параметров сеанса:

- Параметры сеанса являются объектами метаданных, что позволяет «1С:Предприятию» осуществлять повышенный контроль за их использованием.
- Параметры сеанса имеют тип. Набор типов параметров сеанса ограничен. Их важной общей чертой является невозможность изменения внутреннего состояния для объектов этих типов.
- Для установки или получения значения параметра сеанса текущий пользователь должен быть наделен соответствующим правом.
- В клиент-серверном варианте «1С:Предприятия» значения параметров сеанса хранятся на сервере и доступны как с сервера, так и с клиента.
- Параметры сеанса доступны как из встроенного языка «1С:Предприятия», например:

```
ПараметрыСеанса.ТекущийПользователь = ИмяПользователя()
```

[Копировать в буфер обмена](#)

Так и из ограничений доступа к данным, например:

```
Документ.Отчет.Пользователь = &ТекущийПользователь
```

[Копировать в буфер обмена](#)

В последнем случае для получения значения параметра сеанса у текущего пользователя наличия соответствующего права не требуется.

ПРИМЕЧАНИЕ 1. Если параметру сеанса установлен один из следующих типов: *ФиксированныйМассив*, *ФиксированнаяКоллекция* или *ФиксированнаяСтруктура*, то в качестве значения элемента коллекции может выступать значение *Неопределено*.

ПРИМЕЧАНИЕ 2. Объем данных, перемещаемых между клиентом и сервером, для установки одного параметра сеанса, не должен превышать 4 Gb (в сериализованном виде).

5.5.4. Роли и права доступа

5.5.4.1. Общая информация

Каждый пользователь системы должен иметь свободный доступ к общей информации, такой как общие справочники, константы или перечисления.

С другой стороны, необходимо, чтобы каждый пользователь имел дело только с той информацией, которая необходима ему для работы, и никак не мог своими неосторожными действиями повлиять на работу других пользователей или на работоспособность системы в целом.

Конфигуратор системы «1С:Предприятие» предоставляет разработчикам развитые средства администрирования, предназначенные для решения указанных задач.

Прежде всего, в процессе создания конфигурации создается необходимое число типовых ролей, описывающих полномочия различных категорий пользователей на доступ к информации, обрабатываемой системой. Роли могут быть заданы в достаточно широких пределах – от возможности только просмотра ограниченного числа видов документов до полного набора прав по вводу, просмотру, корректировке и удалению любых видов данных.

В системе «1С:Предприятие» различают два типа прав – **основные** и **интерактивные**. **Основные** проверяются всегда, независимо от способа обращения к объектам информационной базы. **Интерактивные** проверяются при выполнении интерактивных операций (просмотр и редактирование в форме и т. д.). Доступные права доступа см. [здесь](#) (или в описании метода глобального контекста `ПравоДоступа()` в синтаксис-помощнике).

Если для объекта, данные которого представляются в форме, установлено (разрешено) право *Просмотр*, но не установлено право *Редактирование*, то в форме данный реквизит будет показан (элемент управления, связанный с данным объектом, отображает значение реквизита), но редактирование значения будет недоступно. Если снять право *Просмотр*, то при попытке открытия формы будет выдано предупреждение: **Нарушение прав доступа!**, и форма не будет открыта.

В списке прав при редактировании роли следует обратить внимание на внутреннюю иерархию прав. Иерархия проявляется в виде «старшинства» прав. При снятии «старшего» права снимаются другие права («младшие»), связанные со «старшим» правом; и наоборот, при установке «младшего» права устанавливаются снятые «старшие» права. Так, при снятии права *Просмотр* снимается право *Редактирование*. Что вполне логично, так как нет смысла предоставлять право редактирования при невозможности показа элемента управления, связанного с данными.

В общем случае права можно задавать:

- на всю конфигурацию в целом,
- объекты,
- реквизиты объектов,
- табличные части,
- реквизиты табличных частей,
- стандартные реквизиты.

При создании новой роли устанавливаются следующие права доступа на корневой объект конфигурации: *ТонкийКлиент*, *ВебКлиент*, *СохранениеДанныхПользователя*, *Вывод*.

5.5.4.2. Привилегированный режим работы

На сервере «1С:Предприятия» фрагменты кода могут исполняться как в обычном, так и в привилегированном режиме. В привилегированном режиме не выполняется проверка доступа на уровне записей, не производится контроль прав и разрешены любые операции, что ускоряет выполнение модулей.

Для управления привилегированным режимом предназначен метод глобального контекста `УстановитьПривилегированныйРежим()`, который позволяет включать или выключать привилегированный режим.

ВНИМАНИЕ! В клиент-серверном варианте вызов метода не оказывает влияния при работе на стороне клиента.

По умолчанию привилегированный режим выключен.

Количество включений привилегированного режима должно совпадать с количеством выключений. Однако если внутри процедуры или функции происходило включение привилегированного режима (один раз или более), но не происходило его выключение, то система автоматически выполнит выключение столько раз, сколько незавершенных включений было в покидаемой процедуре или функции.

Если в процедуре или функции вызовов метода `УстановитьПривилегированныйРежим(Ложь)` сделано больше, чем вызовов метода `УстановитьПривилегированныйРежим(Истина)`, то будет вызвано исключение.

Функция `ПривилегированныйРежим()` возвращает *Истина*, если привилегированный режим еще включен, и *Ложь*, если он полностью выключен. При этом не анализируется количество установок привилегированного режима в конкретной функции.

Программная установка привилегированного режима может потребоваться в случае массированных операций с данными информационной базы, и при этом нет смысла проверять права доступа к данным. Например, существует пользователь, которому доверили выполнять пересчет цен товаров. Тогда в обработке, например, которая это выполняет, можно проверить право текущего пользователя выполнять данную обработку, а затем включить привилегированный режим и выполнить все необходимые операции с базой данных. При этом у пользователя может не быть прав на чтение цен. Но так как данная обработка не выдает пользователю самих цен, а только их пересчитывает, то поставленные задачи ограничения доступа будут также решены.

Также существует возможность стартовать привилегированный сеанс. Это сеанс, в котором привилегированный режим установлен с самого начала работы системы. При этом во время работы метод `ПривилегированныйРежим()` будет всегда возвращать *Истина*, а возможность отключить привилегированный режим не поддерживается. Стартовать привилегированный сеанс может только пользователь, которому доступны административные права (право `Администрирование`). Запуск сеанса можно выполнить с помощью ключа командной строки запуска клиентского приложения `UsePrivilegedMode` или параметра строки соединения с информационной базой `prmod`.

5.5.4.3. Безопасный режим работы

В случае необходимости использования на сервере «ненадежного» программного кода: внешние обработки или программный код, вводимый пользователем для использования в методах `Выполнить()` и `Вычислить()`, можно воспользоваться безопасным режимом работы.

В безопасном режиме:

- Привилегированный режим **отменяется**.
- Переход в привилегированный режим **игнорируется**.
- **Запрещены** операции, приводящие к использованию внешних средств по отношению к платформе «1С:Предприятие»:

- **Механизмы COM:**

- `COMОбъект()`,
- `ПолучитьCOMОбъект()`,
- `ОболочкаHTMLДокумента.ПолучитьCOMОбъект()`.

- **Загрузка внешних компонентов:**

- `ЗагрузитьВнешнююКомпоненту()`,
- `ПодключитьВнешнююКомпоненту()`.

- **Доступ к файловой системе:**

- `ЗначениеВФайл()`,
- `КопироватьФайл()`,
- `ОбъединитьФайлы()`,
- `ПереместитьФайл()`,
- `РазделитьФайл()`,
- `СоздатьКаталог()`,
- `УдалитьФайлы()`,
- `Новый Файл`,
- `Новый xBase`,
- `ЗаписьHTML.ОткрытьФайл()`,
- `ЧтениеHTML.ОткрытьФайл()`,
- `ЧтениеXML.ОткрытьФайл()`,
- `ЗаписьXML.ОткрытьФайл()`,
- `ЧтениеFastInfoSet.ОткрытьФайл()`,
- `ЗаписьFastInfoSet.ОткрытьФайл()`,
- `КаноническаяЗаписьXML.ОткрытьФайл()`,
- `ПреобразованиеXSL.ЗагрузитьИзФайла()`,
- `ЗаписьZipФайла.Открыть()`,
- `ЧтениеZipФайла.Открыть()`,
- `Новый ЧтениеТекста()`,
- `Новый ЗаписьТекста()`,
- `Новый ИзвлечениеТекста()`,
- **изменение свойства** `ИзвлечениеТекста.ИмяФайла`,
- `ИзвлечениеТекста.Записать()`,
- `Новый Картинка()`, если первый параметр – строка;
- `Картинка.Записать()`;
- `Новый ДвоичныеДанные()`;
- `ДвоичныеДанные.Записать()`;
- `ФорматированныйДокумент.Записать()`;

- ГеографическаяСхема.Прочитать();
 - ГеографическаяСхема.Записать();
 - ГеографическаяСхема.Напечатать();
 - ТабличныйДокумент.Прочитать();
 - ТабличныйДокумент.Записать();
 - ТабличныйДокумент.Напечатать();
 - ГрафическаяСхема.Прочитать();
 - ГрафическаяСхема.Записать();
 - ГрафическаяСхема.Напечатать();
 - ТекстовыйДокумент.Прочитать();
 - ТекстовыйДокумент.Записать();
- Доступ к Интернету:
- Новый ИнтернетСоединение,
 - Новый ИнтернетПочта,
 - Новый ИнтернетПрокси,
 - Новый НТТРСоединение,
 - Новый FTPСоединение.

ВНИМАНИЕ! При выполнении запрещенных операций во время выполнения генерирует исключение.

ПРИМЕЧАНИЕ. Внешние отчеты и обработки, открываемые с помощью меню **Файл – Открыть**, выполняются в безопасном режиме, если у пользователя отсутствуют административные права доступа.

Количество включений безопасного режима должно совпадать с количеством выключений. Однако если внутри процедуры или функции происходило включение безопасного режима (один раз или более), но не происходило его выключение, то система автоматически выполнит выключение столько раз, сколько незавершенных включений было в покидаемой процедуре или функции.

Если в процедуре или функции вызовов метода `УстановитьБезопасныйРежим(Ложь)` сделано больше, чем вызовов метода `УстановитьБезопасныйРежим(Истина)`, то будет вызвано исключение.

Программная установка безопасного режима может потребоваться в том случае, когда разработчик конфигурации предполагает использование стороннего (по отношению к конфигурации) программного кода, надежность которого разработчик гарантировать не может. Примером такого кода является выполнение методов `Выполнить()` и `Вычислить()` в тех случаях, когда исполняемый код получается из внешнего мира. В этом случае хорошей практикой будет установка безопасного режима до выполнения этих методов:

```
// Формируется программный код, который следует исполнить
// Возможно, что код загружается из внешних источников
// или введен вручную
ИсполняемыйКод = ПолучитьВыполняемыйКодИзВнешнегоМира();
// Включим безопасный режим
УстановитьБезопасныйРежим(Истина);
// Выполним потенциально опасный код
Выполнить(ИсполняемыйКод);
// Выключим безопасный режим
УстановитьБезопасныйРежим(Ложь);
```

Копировать в буфер обмена

5.5.4.4. Режимы удаления данных

Система «1С:Предприятие» предоставляет пользователям возможность удаления лишней или устаревшей информации в двух режимах:

- **непосредственное удаление объектов**, при котором не производится анализ использования удаляемого объекта в других объектах базы данных;
- использование **контроля ссылочной целостности**, при котором объекты сначала помечаются на удаление, а затем производится контроль наличия ссылок на эти объекты в других объектах.

Если пользователю разрешен режим непосредственного удаления, то в этом случае дополнительная ответственность ложится и на пользователя, выполняющего удаление объектов, и на разработчика (администратора системы), определяющего права пользователей и действия системы при неразрешенных ссылках. Работа системы без контроля ссылочной целостности может, например, быть использована специалистами в процессе отладки конфигурации. Если контроль ссылочной целостности не используется, то удаление объектов происходит непосредственно (без пометки на удаление) и появляется возможность образования неразрешенных ссылок.

Самым радикальным способом установки режима контроля ссылочной целостности является отключение прав непосредственного удаления объектов в целом. Таким способом полностью исключается возможность в пределах данной конфигурации непосредственно удалять объекты. Пользователи будут иметь возможность только помечать объекты на удаление.

Заметим, что также существует возможность непосредственного удаления объектов средствами встроенного языка. Поэтому элементы конкретной конфигурации могут выполнять непосредственное удаление в обход механизма контроля ссылочной целостности. В этом случае ответственность за целостность данных лежит на специалисте, выполняющем конфигурирование системы.

5.5.4.5. Правила сочетания ролей

Роли обычно указываются для каждого вида деятельности. При включении в список пользователей нового пользователя (см. книгу «1С:Предприятие 8.3. Руководство администратора») ему может быть назначена определенная роль или совокупность ролей. В случае использования нескольких ролей алгоритм предоставления доступа по каждому объекту и виду права доступа (например, **Пометка на удаление**) будет работать следующим образом:

- если **хотя бы в одной** роли есть **разрешение**, то доступ будет **открыт**;
- если **во всех** ролях есть **запрещение**, то доступ будет **закрыт**.

5.5.4.6. Редактор прав доступа

В левой части окна редактирования прав выводится дерево объектов конфигурации по всем подсистемам. В правой – список прав по выбранному объекту конфигурации в дереве конфигурации. Если для действия установлен флажок, то оно разрешено.

Так, например, для пользователя с ролью **МенеджерПоПродажам** разрешен просмотр документа **ПриходТовара** и запрещено его интерактивное добавление.

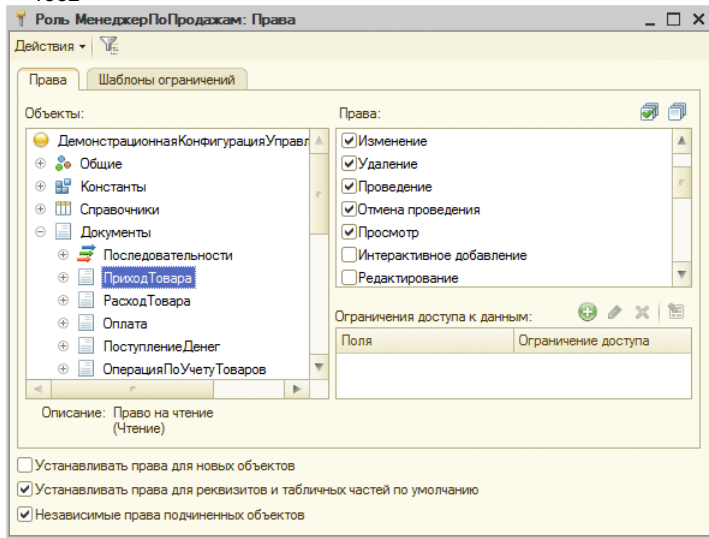


Рис. 71. Редактор прав доступа роли

Состояние флажка **Устанавливать права для новых объектов** определяет, будут ли установлены права для этой роли на новые добавляемые объекты конфигурации (снят по умолчанию для новой роли).

Состояние флажка **Устанавливать права для реквизитов и табличных частей по умолчанию** определяет, будут ли установлены права для этой роли на реквизиты (включая стандартные) и табличные части (включая стандартные) для новых объектов конфигурации (установлен по умолчанию).

При изменении состояния флажка **Устанавливать права для реквизитов и табличных частей по умолчанию** система предлагает изменить (установить или сбросить) права доступа для всех реквизитов (включая стандартные) и табличных частей (включая стандартные) всех объектов конфигурации. В случае отказа от предлагаемого действия изменений в существующих объектах не происходит, а изменяется только поведение по умолчанию для новых объектов.

При создании новой роли все права устанавливаются конфигуратором в следующее состояние:

- для объектов права не установлены;
- для реквизитов (включая стандартные) и табличных частей (включая стандартные) права установлены.

Флажок **Независимые права подчиненных объектов** определяет, каким образом система будет учитывать состояние прав на родительский объект при определении прав на подчиненный объект. Если флажок установлен, то права родительского объекта не учитываются. Если флажок снят, то при определении права подчиненного объекта анализируется соответствующее право родительского объекта. Если у родительского объекта право отсутствует, у подчиненного объекта также отсутствует право, вне зависимости от состояния права у подчиненного объекта.

Флажок **Независимые права подчиненных объектов** не отменяет того факта, что право на подчиненный объект требует наличия этого же права и на родительский. То есть для реквизита или табличной части нужно право на объект, для реквизита табличной части – на табличную часть и объект. Указанное свойство имеет влияние в том случае, если у пользователю назначено несколько ролей, права которых складываются «по ИЛИ». То есть, если в роли установлено свойство **Независимые права подчиненных объектов** и есть право только на реквизит, а в другой роли есть право только на объект, то после сложения прав ролей у пользователя будет право на реквизит

При назначении прав доступа на реквизиты (табличные части) отчетов/обработок следует помнить следующую особенность: если у роли снят флажок **Независимые права подчиненных объектов** и для реквизита (табличной части) установлено право **Редактирование**, а для отчета/обработки право **Просмотр** не установлено, то считается, что и для реквизита (табличной части) право **Редактирование** не установлено.

Если пользователю назначено несколько ролей, то проверка прав родительского объекта выполняется до того, как будет выполнено сочетание прав по ролям (для определения результирующего права на объект), см. [здесь](#).

Действие флажка **Независимые права подчиненных объектов** распространяется на следующие объекты:

- реквизиты (включая стандартные);
- табличные части (включая стандартные);
- реквизиты табличных частей (включая реквизиты стандартных табличных частей);
- команды.

Чтобы изменить право доступа, в левом списке следует выбрать объект конфигурации, а в правом изменить состояние флажка в нужной строке вида действия. Если требуется изменить доступ сразу ко всем объектам выбранной ветви, нужно указать в левой части эту ветвь и изменить установку прав доступа.

Описание каждой роли можно вывести в табличный или текстовый документ с помощью пункта **Действия – Вывести список**.

5.5.4.7. Просмотр и редактирование всех ролей

Если в конфигурации используется несколько ролей, то для удобства просмотра и редактирования прав рекомендуется использовать окно **Все роли**. Для его открытия в дереве объектов конфигурации окна **Конфигурация** нужно указать ветвь **Роли** и в контекстном меню выбрать команду **Все роли**.

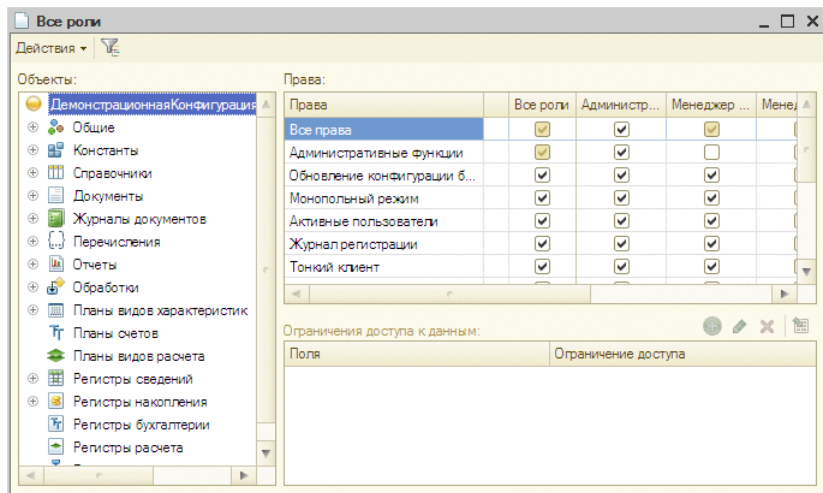


Рис. 72. Окно редактирования «Все роли»

В окне расположены три табличных поля. В первом (слева) производится выбор нужного объекта конфигурации. В первой колонке второго табличного поля выводится список прав по выбранному объекту. Другие колонки предназначены для указания использования каждого права для каждой существующей роли.

Если для какой-нибудь роли требуется установить или снять все права, то достаточно в первой строке табличного поля установить или снять флажок разрешения.

Если для какого-нибудь права требуется установить или снять его разрешение во всех ролях, то достаточно в первой колонке табличного поля установить или снять флажок разрешения.

Поддерживается возможность перестановки колонок, соответствующих ролям.

В третьем табличном поле редактируются условия доступа к данным на уровне отдельных полей и записей.

5.5.4.8. Ограничение доступа к данным

5.5.4.8.1. Общие замечания

Механизм ограничений доступа к данным позволяет управлять правами доступа не только на уровне объектов метаданных, но и на уровне объектов базы данных «1С:Предприятия». Для ограничения доступа к данным могут быть использованы следующие объекты «1С:Предприятия»:

- роли,
- параметры сеанса,
- функциональные опции,
- привилегированные общие модули,
- ключевое слово **РАЗРЕШЕННЫЕ** в языке запросов.

Совместное использование перечисленных объектов позволяет обеспечить максимальную гибкость при необходимости разграничения прав доступа к данным между пользователями, выполняющими различные функции.

Ограничения доступа к данным могут накладываться на следующие операции с данными (права доступа): чтение (право **Чтение**), добавление (право **Добавление**), изменение (право **Изменение**) и удаление (право **Удаление**). Текущий пользователь будет иметь возможность выполнить требуемую операцию в следующих случаях:

- Для операций чтения и удаления объект, находящийся в базе данных, должен соответствовать ограничению доступа к данным.
- Для операции добавления ограничению доступа к данным должен соответствовать объект, который планируется записать в базу данных.
- Для операции изменения ограничению доступа к данным должен соответствовать объект как до изменения (чтобы объект был прочитан), так и после изменения (чтобы объект был записан).

При наложении ограничений доступа к данным следует помнить, что для операций изменения, добавления и удаления можно задать только одно условие, а для операции чтения можно задать более одного ограничения доступа к данным. Это означает, что для чтения разных полей объекта могут быть заданы разные условия, причем при задании условия можно указать как имя конкретного поля, так и специальное поле **Прочие поля**. В первом случае условие будет накладываться только в том случае, если в выборке (которой выполняется чтение данных) будет присутствовать поле, для которого задано ограничение, а во втором – ограничение будет накладываться для всех полей объекта, кроме полей, для которых ограничения заданы явным образом.

При задании ограничения на конкретное поле, это поле будет считано в том случае, если ограничение выполняется, а при задании ограничения на **Прочие поля**, данные объекта будут прочитаны только в том случае, если ограничение выполняется для всех полей объекта, попавших в запрос чтения данных.

Для объектов базы данных следующих видов могут быть наложены различные ограничения на разные виды изменений (добавление, модификацию, удаление):

- Планы обмена,
- Справочники,
- Документы,
- Планы видов характеристик,
- Планы счетов,
- Планы видов расчета,
- Бизнес-процессы,
- Задачи.

Для следующих видов объектов базы данных возможно наложение ограничений на чтение не только всего объекта целиком, но и отдельных его полей:

- Планы обмена,
- Справочники,
- Документы,

- Журналы документов,
- Планы видов характеристик,
- Планы счетов,
- Планы видов расчета,
- Регистры сведений,
- Бизнес-процессы,
- Задачи.

ВНИМАНИЕ! При обращении к полям объектов базы данных посредством свойств прикладных объектов из встроенного языка «1С:Предприятия» выполняется чтение всего объекта целиком, а не только значения используемого поля. Исключением является получение представления, когда будут прочитаны только значения полей, участвующих в формировании представления.

Ограничения доступа содержатся в ролях, они могут быть указаны для большинства объектов метаданных и записываются на специальном языке, являющимся подмножеством языка запросов.

5.5.4.8.2. Язык ограничения доступа к данным

Ограничения доступа к данным описываются на специальном языке, являющимся подмножеством языка запросов (подробное описание языка запросов см. [здесь](#)). Язык ограничения доступа к данным имеет следующие изменения относительно языка запросов:

- В запросе ограничения доступа к данным всегда присутствует одна таблица в качестве источника данных – это таблица объекта, на который накладывается ограничение (основной объект ограничения).
- Сокращено описание запроса. Язык ограничения доступа к данным использует только секции **ИЗ** и **ГДЕ** языка запросов. Так, описание языка запросов выглядит следующим образом:

<Описание запроса >

```

ВЫБРАТЬ [РАЗРЕШЕННЫЕ] [РАЗЛИЧНЫЕ] [ПЕРВЫЕ <Количество>]
<Список полей выборки>
ИЗ <Список источников>
ГДЕ <Условие отбора>
[СГРУППИРОВАТЬ ПО <Поля группировки>]
[ИМЕЮЩИЕ <Условие отбора>]
[ДЛЯ ИЗМЕНЕНИЯ [ <Список таблиц верхнего уровня>]]

```

[Копировать в буфер обмена](#)

В то время как описание языка запросов ограничения доступа к данным выглядит следующим образом:

<Описание запроса >

```

[Псевдоним таблицы основного объекта ограничения]
ИЗ <Список источников>
ГДЕ <Условие отбора>

```

[Копировать в буфер обмена](#)

Описание списка источников см. [здесь](#). Описание условий отбора см. [здесь](#). При этом следует помнить, что во вложенных запросах, используемых в языке ограничения доступа к данным, ограничен набор допустимых возможностей (см. [здесь](#));

- В качестве элементов условий можно указывать параметры сеанса (см. [здесь](#)) и функциональные опции (см. [здесь](#));
- В любом месте запроса ограничения доступа к данным допустимо использование шаблонов, упрощающих написание ограничений (см. [здесь](#)).

Главной частью ограничения является условие, которое вычисляется для каждой записи таблицы базы данных, на которую накладывается ограничение доступа к данным. Запись считается доступной в том случае, если в результате работы условия для одной записи таблицы основного объекта ограничения получена не пустая таблица (т.е. таблица, в которой 1 или более записей). Если в результате работы условия получается пустая таблица – запись, для которой условие исполнилось таким образом, считается недоступной. Причем изменение записи таблицы основного объекта ограничения считается допустимым, если запись не противоречит ограничению, указанному для права, как до выполнения операции изменения, так и после выполнения этой операции.

Поля таблиц

В ограничениях доступа к данным можно использовать:

- **Поля таблиц объекта**, для которого описывается ограничение доступа к данным.

Например, если ограничение накладывается на чтение элементов справочника **Контрагенты**, то в ограничении могут использоваться поля справочника **Контрагенты** и его табличных частей. В частности, наиболее простые ограничения на чтение элементов справочника **Контрагенты** могут выглядеть так:

```
ГДЕ Наименование = "Кирпичный завод"
```

[Копировать в буфер обмена](#)

Или так:

```
ГДЕ Продукция.Наименование = "Кирпич красный"
```

[Копировать в буфер обмена](#)

Где **Продукция** – это табличная часть справочника **Контрагенты**.

- **Поля таблиц объектов, доступных по ссылкам** в основном объекте ограничения.

Например, если реквизит **ОсновнойМенеджер** справочника **Контрагенты** имеет тип ссылки на справочник **Пользователи**, то ограничение доступа может иметь, например, следующий вид:

```
ГДЕ ОсновнойМенеджер.Код = "Иванов"
```

[Копировать в буфер обмена](#)

Или:

```
ГДЕ ОсновнойМенеджер.ФизическоеЛицо.Наименование = "Петровский"
```

[Копировать в буфер обмена](#)

- **Поля таблиц объектов, связанных с основным объектом** ограничения некоторыми условиями и выражения над ними.

Например, на чтение элементов справочника **Контрагенты** может быть наложено следующее ограничение:

```
Контрагенты
ИЗ
Справочник.Контрагенты КАК Контрагенты
ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Пользователи КАК Пользователи
```

[Копировать в буфер обмена](#)

ПО Контрагенты.ОсновнойМенеджер.Наименование = Пользователи.Наименование
ГДЕ Пользователи.ФизическоеЛицо.Наименование = "Петровский"

В этом ограничении используются поля элементов справочника **Пользователи**, связанных с данным элементом справочника **Контрагенты** по значению полей **Наименование**.

Вложенные запросы

Вложенные запросы используются для формирования наборов записей, которые могут использоваться:

- для связывания с таблицей основного объекта ограничения;
- для использования в качестве операнда операций сравнения **В** или **НЕ В**.

Во вложенных запросах могут использоваться любые средства языка запросов, кроме:

- оператора **В ИЕРАРХИИ**;
- предложения **ИТОГИ**;
- результаты вложенных запросов не должны содержать табличные части;
- некоторых виртуальных таблиц, в частности **ОстаткиИОбороты**.

В следующем примере ограничения на чтение из справочника **Контрагенты**, вложенный запрос используется в качестве набора записей для связывания с основным объектом ограничения:

[Копировать в буфер обмена](#)

```
Контрагенты
ИЗ
Справочник.Контрагенты КАК Контрагенты
ЛЕВОЕ СОЕДИНЕНИЕ
(ВЫБРАТЬ
    Пользователи.Наименование, Пользователи.ФизическоеЛицо
ИЗ
    Справочник.Пользователи КАК Пользователи
ГДЕ
    Пользователи.Код > "Петечкин") КАК Пользователи
ПО Контрагенты.ОсновнойМенеджер.Наименование = Пользователи.Наименование
ГДЕ Пользователи.ФизическоеЛицо.Наименование = "Петровский"
```

В следующем примере приведено ограничение на чтение из справочника **ПаспортныеДанныеФизЛиц**, в котором вложенный запрос используется в качестве операнда операции сравнения **В**:

[Копировать в буфер обмена](#)

```
ПаспортныеДанныеФизЛиц
ГДЕ
ПаспортныеДанныеФизЛиц.ФизЛицо В
(ВЫБРАТЬ РАЗЛИЧНЫЕ
    Работники.ФизЛицо КАК ФизЛицо
ИЗ
    РегистрСведений.Работники КАК Работники)
```

Если во вложенном запросе необходимо получить данные из табличной части, то в разделе **ИЗ** вложенного запроса необходимо обращаться непосредственно к табличной части. Например, вместо:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    Ссылка КАК Ссылка,
    Продукция.Наименование КАК НаименованиеПродукции
ИЗ Справочник.Контрагенты
```

в качестве запроса, вложенного в ограничение, следует использовать:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    Ссылка КАК Ссылка,
    Наименование КАК НаименованиеПродукции
ИЗ Справочник.Контрагенты.Продукция
```

Параметры сеанса

В составе запросов ограничения доступа к данным могут входить параметры сеанса. Например, на чтение элементов справочника **ГруппыПисемЭлектроннойПочты** может быть задано следующее ограничение доступа:

[Копировать в буфер обмена](#)

```
ГДЕ Владелец.ДоступКУчетнойЗаписи.Пользователь = &ТекущийПользователь
И Владелец.ДоступКУчетнойЗаписи.Администрирование = ИСТИНА
```

Где **ТекущийПользователь** – это параметр сеанса (см. [здесь](#)).

Функциональные опции

В составе запросов ограничения доступа к данным могут входить функциональные опции. Могут использоваться только не зависящие от параметров функциональные опции. Например, если у справочника **Номенклатура** есть реквизит **ОсновнойСклад**, то ограничение на чтение данного реквизита может выглядеть следующим образом:

[Копировать в буфер обмена](#)

```
ГДЕ &УчетПоСкладам = ИСТИНА
```

Где **УчетПоСкладам** – это функциональная опция (см. [здесь](#)).

Особенности использования

В ограничениях на объекты базы данных следующих типов могут быть использованы не все поля основного объекта данных ограничения:

- в **регистрах накопления** ограничения доступа могут содержать только измерения основного объекта ограничения;
- в **регистрах бухгалтерии** в ограничениях можно использовать только балансовые измерения основного объекта ограничения.

ПРИМЕЧАНИЕ. Если в условиях ограничения доступа к данным оборотного регистра накопления используются измерения, не входящие в итоги, то при обращении к виртуальной таблице оборотов не используются хранимые итоги и запрос выполняется полностью по таблице движений.

5.5.4.8.3. Действия ограничения доступа

Ограничения доступа проверяются при любом выполнении соответствующих операций над объектами базы данных (из диалогов, из встроенного языка, посредством запросов) и могут действовать одним из двух способов:

- **Все.** Способ «все» подразумевает, что некоторая операция над данными (из диалогов, из встроенного языка или посредством запросов) должна быть выполнена над

всеми подразумеваемыми данной операцией объектами базы данных. Если при выполнении такой операции должны быть прочитаны или изменены объекты базы данных, для которых не выполняются соответствующие ограничения доступа, то операция завершается аварийно из-за нарушения прав доступа.

• **Разрешенные.** Способ «разрешенные» подразумевает, что при выполнении операции над данными должны быть прочитаны только те объекты базы данных, которые удовлетворяют соответствующим ограничениям доступа. Объекты базы данных, не удовлетворяющие ограничениям доступа, при выполнении такой операции считаются отсутствующими и на результат операции не влияют.

Ограничения доступа к данным накладываются на объекты базы данных в момент обращения «1С:Предприятия» к базе данных. В клиент-серверном варианте «1С:Предприятия» наложение ограничений выполняется на сервере «1С:Предприятия».

Способ действия ограничений, выбираемый для выполнения каждой операции над данными, определяется назначением этой операции и степенью ответственности ее результатов. В частности, способ «разрешенные» используется при отображении динамических списков и некоторых других интерактивных действиях. Способ «все» используется при выполнении любых операций с прикладными объектами из встроенного языка «1С:Предприятия», в том числе при любых изменениях объектов базы данных. Поэтому, например, могут возникнуть затруднения при построении отбора для метода `Выбрать()` менеджеров справочников, документов и других с последующим обходом результата в том случае, если на соответствующий объект установлено достаточно сложное ограничение, поскольку не всякое условие в ограничении прав доступа может быть адекватно представлено в виде отбора для метода `Выбрать()`.

В запросах способом действия ограничений доступа к данным можно управлять. Для этого в языке запросов предусмотрено ключевое слово `РАЗРЕШЕННЫЕ`. Если в запросе не указано `РАЗРЕШЕННЫЕ`, то ограничения действуют способом «все». Если слово `РАЗРЕШЕННЫЕ` указано, то выбирается способ «разрешенные».

Важно, что если в запросе не указано ключевое слово `РАЗРЕШЕННЫЕ`, то все отборы, заданные в этом запросе, не должны противоречить ни одному из ограничений на чтение объектов базы данных, используемых в запросе. При этом если в запросе используются виртуальные таблицы, то соответствующие отборы должны быть наложены и на сами виртуальные таблицы.

Пример:

```

ВЫБРАТЬ
    КонтактнаяИнформацияСрезПервых.Представление
ИЗ РегистрСведений.КонтактнаяИнформация.СрезПоследних(, Тип = &Тип)
    КАК КонтактнаяИнформацияСрезПервых
ГДЕ
    КонтактнаяИнформацияСрезПервых.Тип = &Тип
  
```

[Копировать в буфер обмена](#)

При использовании объектной техники не поддерживается получение доступа к данным в режиме `РАЗРЕШЕННЫЕ`. Предполагается, что объектная техника используется для наиболее ответственных операций над данными, в том числе для их изменения. Для получения при помощи объектной техники всех данных, независимо от установленных ограничений, можно выполнять необходимые действия в привилегированном модуле или от имени пользователя с полными правами. Средств получения только разрешенных данных в объектной технике не предусмотрено.

5.5.4.8.4. Механизм наложения ограничений

Любая операция над данными, хранимыми в базе данных, в «1С:Предприятии» в конечном счете приводит к обращению к базе данных с некоторым запросом на чтение или изменение данных. В процессе исполнения запросов к базе данных внутренние механизмы «1С:Предприятия» выполняют наложение ограничений доступа. При этом:

- Формируется список прав (чтение, добавление, изменение, удаление), список таблиц базы данных и список полей, используемых этим запросом.
- Из всех ролей текущего пользователя выбираются ограничения доступа к данным для всех прав, таблиц и полей, задействованных в запросе. При этом если какая-нибудь роль не содержит ограничений доступа к данным какой-нибудь таблицы или поля, то это значит, что в данной таблице доступны значения требуемых полей из любой записи. Иначе говоря, отсутствие ограничения доступа к данным означает наличие ограничения `ГДЕ Истина`.
- Получаются текущие значения всех параметров сеанса и функциональных опций, участвующих в выбранных ограничениях.

Для получения значения параметра сеанса от текущего пользователя не требуется наличие права на получение этого значения. Однако если значение некоторого параметра сеанса не было установлено, то произойдет ошибка и запрос к базе данных выполнен не будет.

На получение функциональных опций оказывает влияние свойство функциональной опции `Привилегированный режим` (см. [здесь](#)). Если это свойство сброшено, то текущий пользователь должен обладать правами на чтение объекта, в котором хранится функциональная опция.

- Ограничения, полученные из одной роли, объединяются операцией `И`.
- Ограничения, полученные из разных ролей, объединяются операцией `ИЛИ`.
- Построенные условия добавляются к SQL-запросам, с которыми «1С:Предприятие» обращается к СУБД. При обращении к данным со стороны условий ограничения доступа проверка прав не выполняется (ни к объектам метаданных, ни к объектам базы данных). Причем механизм добавления условий зависит от выбранного способа действия ограничений «все» или «разрешенные».

Способ «все»

При наложении ограничений способом «все» к SQL-запросам добавляются условия и поля так, чтобы «1С:Предприятие» могло получить информацию о том, были ли в процессе исполнения запроса к базе данных использованы данные, запрещенные для данного пользователя или нет. Если запрещенные данные были использованы, то инициируется аварийное завершение запроса. Наложение ограничений доступа способом «все» схематически представлено на [рис. 73](#):

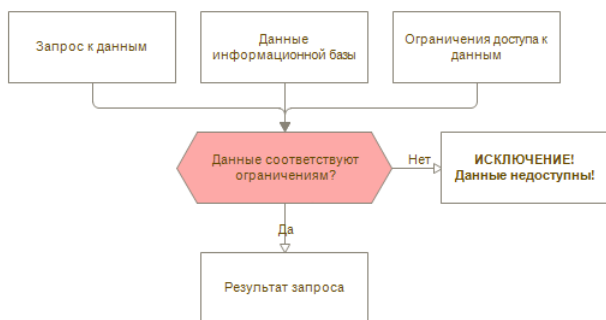


Рис. 73. Способ «все»

Способ «разрешенные»

При наложении ограничений способом «разрешенные» к SQL-запросам добавляются такие условия, чтобы запрещенные текущему пользователю записи не оказывали влияния на результат запроса. Иначе говоря, при наложении ограничений в режиме «разрешенные» запрещенные данному пользователю записи считаются отсутствующими, что схематически представлено на [рис. 74](#):

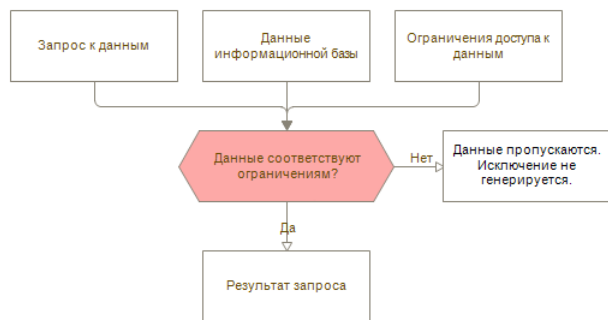


Рис. 74. Способ «разрешенные»

5.5.4.8.5. Другие объекты, связанные с ограничениями доступа к данным

При разработке конфигураций с использованием ограничений доступа к данным могут оказаться полезными такие объекты метаданных, как параметры сеанса, функциональные опции и общие модули с флажком **Привилегированный**.

Параметры сеанса

Параметры сеанса могут использоваться в ограничениях доступа к данным аналогично тому, как в запросе могут использоваться параметры запроса.

Функциональные опции

Не зависящие от параметров функциональные опции могут использоваться в ограничениях доступа к данным аналогично тому, как в запросе могут использоваться параметры запроса.

Привилегированные общие модули

Если для общего модуля установлен флажок **Привилегированный**, то исполнение процедур и функций этого модуля приобретает важную специфику:

- В клиент-серверном варианте «1С:Предприятия» привилегированным может быть только тот модуль, который исполняется на сервере.
- Исполнение процедур и функций привилегированного модуля и всего, что из них вызвано, выполняется при выключенной системе ограничения прав, как к объектам метаданных, так и к данным. Таким образом, из привилегированного модуля может быть выполнена любая операция над любыми объектами даже в том случае, если текущий пользователь не имеет соответствующих прав.

Привилегированные модули предназначены для начальной установки значений параметров сеанса, используемых в ограничениях доступа к данным. Еще общие модули могут быть использованы для некоторых целостных действий над данными со стороны пользователя с ограниченными правами. Например, если в функции пользователя входит ввод и проведение документов, но пользователь не должен иметь доступа к данным, на которые влияет проведение документа, то выполнение операции проведения может быть вынесено в привилегированный модуль. Это позволит пользователю проводить документы без предоставления ему прав на другую информацию (регистры, например).

Привилегированный режим

Имеется возможность программной установки привилегированного режима при работе с данными. Программная установка привилегированного режима может потребоваться в случае массированных операций с данными информационной базы, и при этом нет смысла проверять права доступа к данным. Описание привилегированного режима см. [здесь](#).

5.5.4.8.6. Использование препроцессора

При редактировании текста ограничения доступа к данным возможно использование инструкций препроцессора. Доступны следующие инструкции:

```
#ЕСЛИ <Выражение> #ТОГДА
#ИНАЧЕЕСЛИ <Выражение> #ТОГДА
#ИНАЧЕ
#КОНЕЦЕСЛИ
```

[Копировать в буфер обмена](#)

<Выражение> – произвольное логическое выражение на встроенном языке, результат которого имеет тип **Булево**. Выражение может содержать:

- операции сравнения <, >, <=, >=, =, <>;
- логические операции И, ИЛИ, НЕ;
- параметры сеанса – используется синтаксис &Параметр, где Параметр – имя параметра сеанса.

Если результатом выражения инструкции **#ЕСЛИ** или **#ИНАЧЕЕСЛИ** является значение **Истина**, то в результирующий текст инструкции ограничения доступа помещается текст, расположенный после ключевого слова **#ТОГДА**. Если же результатом выражения является значение **Ложь**, то текст, расположенный после ключевого слова **#ТОГДА**, не помещается в текст инструкции ограничения доступа. Текст, расположенный после инструкции **#ИНАЧЕ**, будет помещен в результирующий текст ограничения доступа, если ни одно из ранних условий не было выполнено.

ПРИМЕЧАНИЕ. Если текст ограничения доступа к данным содержит инструкции препроцессора, то такое ограничение не проходит проверку синтаксиса при редактировании и не может быть изменено при помощи конструктора.

Пример:

```
#ЕСЛИ &ТекущийПользователь <> "Климова" #ТОГДА
<текст ограничения доступа>
#КОНЕЦЕСЛИ
```

[Копировать в буфер обмена](#)

Здесь **ТекущийПользователь** – параметр сеанса типа **СправочникСсылка.Пользователи**.

Такая конструкция означает, что условие для установки ограничения доступа будет проверяться для всех пользователей из справочника, кроме пользователя **Климовой**.

5.5.4.8.7. Шаблоны текста ограничения доступа

Роль может содержать список шаблонов ограничения доступа, которые описываются на закладке **Шаблоны ограничений** формы роли. Также шаблоны ограничения доступа можно редактировать в редакторе группового редактирования ограничений доступа и шаблонов (см. [здесь](#)).

Каждый шаблон ограничения доступа имеет имя и текст. Имя шаблона подчиняется обычным правилам для имен, принятых в системе «1С:Предприятие».

Текст шаблона содержит часть текста на языке ограничения доступа к данным и может содержать параметры, которые выделяются при помощи символа "#".

После символа "#" могут следовать:

• Одно из ключевых слов:

- **Параметр**, после которого в скобках указывается номер параметра в шаблоне;
- **ТекущаяТаблица** – обозначает вставку в текст полного имени таблицы, для которой строится ограничение;
- **ИмяТекущейТаблицы** – обозначает вставку в текст полного имени таблицы (как строковое значение, в кавычках), к которой применяется инструкция, на текущем варианте встроеного языка;
- **ИмяТекущегоПраваДоступа** – содержит имя права, для которого выполняется текущее ограничение: **ЧТЕНИЕ/READ**, **ДОБАВЛЕНИЕ/INSERT**, **ИЗМЕНЕНИЕ/UPDATE**, **УДАЛЕНИЕ/DELETE**;
- имя параметра шаблона – означает вставку в текст ограничения соответствующего параметра шаблона;
- символ **"#"** – обозначает вставку в текст одного символа **"#"**.

В выражении ограничения доступа могут содержаться:

- Шаблон ограничения доступа, который указывается в формате **#ИмяШаблона("Значение параметра шаблона 1", "Значение параметра шаблона 2", ...)**. Каждый параметр шаблона заключается в двойные кавычки. При необходимости указания в тексте параметра символа двойной кавычки следует использовать две двойные кавычки.
- Функция **СтрСодержит(ГдеИщем, ЧтоИщем)**. Функция предназначена для поиска вхождения строки **ЧтоИщем** в строке **ГдеИщем**. Возвращает **Истина** в случае, если вхождение обнаружено и **Ложь** – в противном случае.
- Оператор **+** для конкатенации строк.

Для удобства редактирования текста шаблона на закладке **Шаблоны ограничений** в форме роли нужно нажать кнопку **Установить текст шаблона**. В открывшемся диалоге ввести текст шаблона и нажать кнопку **OK**.

Система «1С:Предприятие» выполняет проверку синтаксиса текстов шаблонов, проверку синтаксиса использования шаблонов и макроподстановку текстов шаблонов ограничения доступа роли в текст запроса.

Макроподстановка шаблона заключается:

- в замене вхождений параметров в тексте шаблона на значения параметров из выражения использования шаблона в тексте ограничения;
- в замене выражения использования шаблона в тексте запроса на получившийся текст шаблона.

При вызове конструктора запроса для условия, содержащего шаблоны ограничения доступа, выдается предупреждение о замене всех шаблонов.

Далее приведены примеры шаблонов ограничений:

Имя шаблона	Шаблон
Тело шаблона	Итого = #Параметр(1)
Использование	Где #Шаблон("10")
Результат	Где Итого = 10

Имя шаблона	Шаблон1(ВидДокумента)
Тело шаблона	ВидДокумента = #ВидДокумента
Использование	Где #Шаблон1("Накладная")
Результат	Где ВидДокумента = "Накладная"

Имя шаблона	Шаблон2
Тело шаблона	ВидДокумента = #Параметр(1) ## #Параметр(2)
Использование	Где #Шаблон2("Накладная", "1")
Результат	Где ВидДокумента = "Накладная # 1"

Имя шаблона	Шаблон3
Тело шаблона	ВидДокумента = #Параметр(3)
Использование	Где #Шаблон3(" ", " ", "Накладная")
Результат	Где ВидДокумента = "Накладная"

5.5.4.8.8. Общие рекомендации по ограничению прав

Чтобы гибко управлять доступом пользователей к данным в соответствии с функциями при установке ограничений доступа к данным, рекомендуется придерживаться следующих принципов:

- Нужно выбрать совокупность информации (может быть зависимой от текущего пользователя), для которой целесообразна предварительная подготовка. Выбранная информация должна, с одной стороны, максимально упростить ограничения доступа к данным, а с другой стороны, не должна иметь слишком большой объем. Распределить ее по параметрам сеанса.
- Установить значения параметров сеанса в обработчике **УстановкаПараметровСеанса()** модуля сеанса.
- Задать ограничения доступа к тем данным, для которых это оправданно (данные являются секретными или наиболее важными для сохранения целостности системы). Необходимо иметь в виду, что установка ограничения доступа может привести к замедлению любого обращения к этим данным. Излишняя сложность ограничений также может привести к замедлению.
- При необходимости обеспечить выполнение некоторого ограниченного количества операций над данными со стороны пользователя, которому полный доступ к этим данным давать нецелесообразно, вынести эти действия в привилегированные модули или явно включать и выключать привилегированный режим в соответствующих местах программного кода (см. [здесь](#)).
- Доступ к данным при различных проверках, выполняемых системой при записи объектов, выполняется в привилегированном режиме (см. [здесь](#)). Это позволяет не отключать ограничения в правах на уровне записей для соответствующих полей, если работа конфигурации с этими данными планируется **только в управляемом режиме**:
 - для справочников при проверке родителя, владельца и уникальности кода;
 - для документов, бизнес-процессов и задач при проверке уникальности номера;
 - для планов обмена отключена при проверке уникальности кода;
 - для планов счетов и планов видов характеристик при проверке родителя и уникальности кода.

При создании запроса ограничения к данным следует помнить о некоторых ограничениях и особенностях:

- Если для объектной таблицы заданы ограничения доступа к данным и в запросе к данным используется объединение с такой таблицей, то в условии соединения (секция запроса **PO**) не допускается использование табличной части объекта с заданным ограничением доступа.
 - Если в запросе указана таблица, у которой в запросе не используется ни одного поля, то на эту таблицу накладываются все ограничения доступа к данным. Например, запрос **ВЫБРАТЬ КОЛИЧЕСТВО(*) ИЗ Справочник.Контрагенты** будет исполнен с учетом всех ограничений доступа, заданными для справочника **Тест**. Ограничения накладываются «по ИЛИ». Это значит, что будут доступны все записи, доступные хотя бы по одному условию. Если для каких-то полей не задано условий, то запрос будет выполнен для всех записей таблицы.
- Если в запросе используется таблица верхнего уровня, то ограничения, заданные для колонок вложенных таблиц, не накладываются.
- Если в запросе используется вложенная таблица, то накладываются ограничения как для вложенной таблицы, так и для таблицы верхнего уровня. Например, запрос **ВЫБРАТЬ КОЛИЧЕСТВО(*) ИЗ Справочник.Контрагенты.Договора** будет исполнен с учетом всех ограничений для справочника **Контрагенты**, а также с учетом ограничений, относящихся к табличной части **Договора**.
- Если доступ к полям, необходимым для получения представления ссылочного объекта метаданных, запрещен с помощью ограничений доступа к данным или доступ к объекту запрещен на уровне прав доступа, то получение представления такого объекта не влияет на ход текущей транзакции.

5.5.4.8.9. Конструктор ограничения доступа к данным

Для вызова конструктора в табличном поле **Ограничения доступа к данным** в колонке **Ограничение доступа** нужно перейти в режим редактирования и нажать кнопку выбора.

На экран выводится форма конструктора:

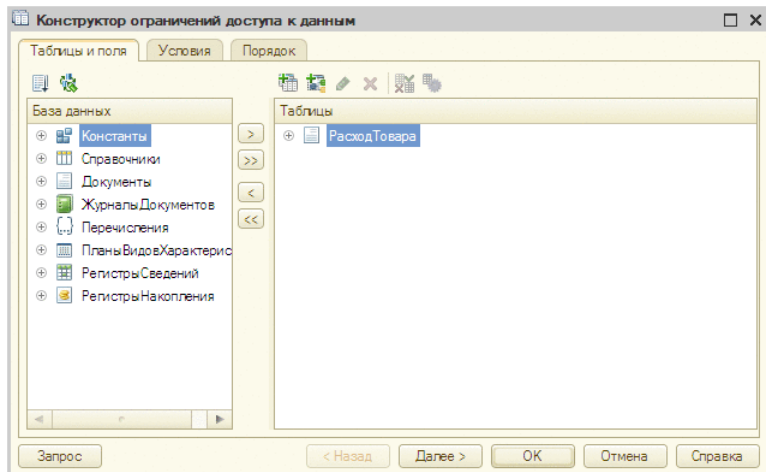


Рис. 75. Закладка «Таблицы и поля» конструктора ограничений

С его помощью производится формирование условий для установки ограничения доступа к данным.

На закладке **Таблицы и поля** следует выбрать нужные объекты и перенести их в раздел **Таблицы и Поля**. Если указано несколько таблиц, то в конструктор добавляется закладка **Связи**.

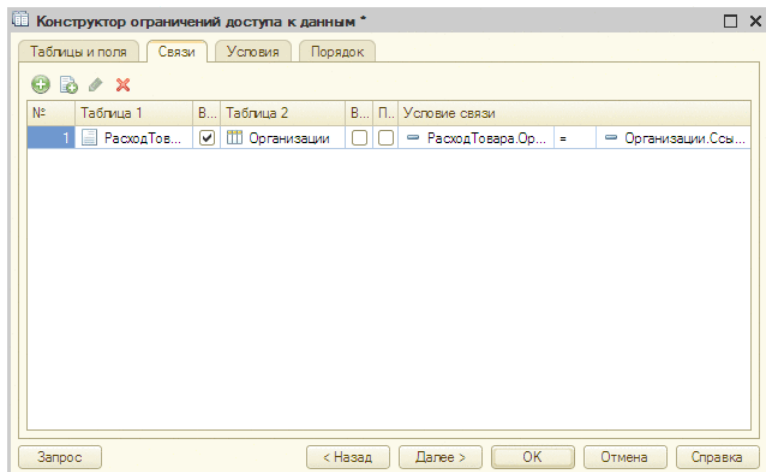


Рис. 76. Закладка «Связи» конструктора ограничений

На закладке **Связи** формируются условия, которые накладываются на связи между полями таблиц. Для ввода нового условия нужно нажать кнопку **Добавить** и в колонке **Таблица1** выбрать одну из таблиц. В колонке **Таблица2** выбрать таблицу, поля которой связаны с полями первой. Ниже списка условий расположены элементы управления, с помощью которых формируется условие связи таблиц.

Если выбран простой тип условия, то в **Поле1** и **Поле2** выбираются связанные поля указанных таблиц и задается условие сравнения. Если выбраны поля, сравнение которых не производится, то в строке списка условий в колонке **Условие связи** выводится текст: **Неверно заполненное условие**.

На закладке **Условия**, если требуется, нужно указать условия, по которым будет выполняться отбор исходных данных.

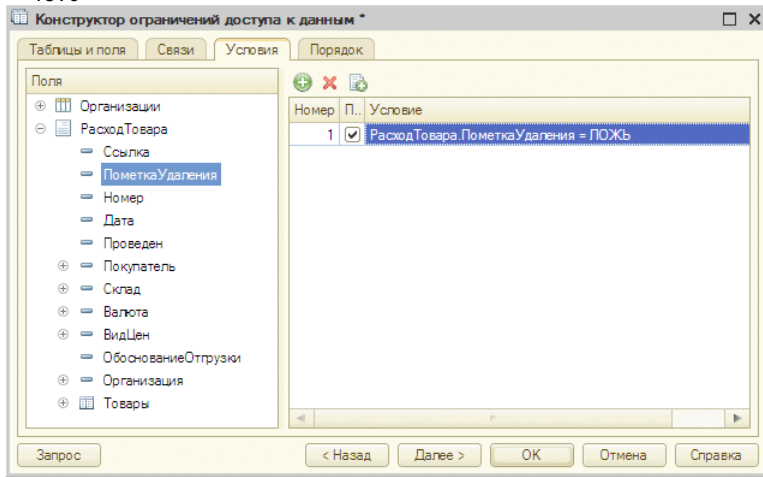


Рис. 77. Закладка «Условия» конструктора ограничений

По каждому выбранному полю необходимо выбрать вид условия и указать наименование параметра. В качестве параметра допускается использование параметра сеанса. Разрешается указывать несколько условий. В этом случае в колонке **Условие** табличного поля условий текст условия выводится в несколько строк.

В любой момент создания запроса текст запроса можно просмотреть, нажав кнопку **Запрос**.

5.5.4.8.10. Групповое редактирование ограничений прав доступа и шаблонов

Режим группового редактирования ограничений прав доступа и шаблонов вызывается командой **Все ограничения доступа** контекстного меню ветки **Роли**. В открывшейся форме присутствуют две закладки: **Ограничения доступа** и **Шаблоны ограничений**.

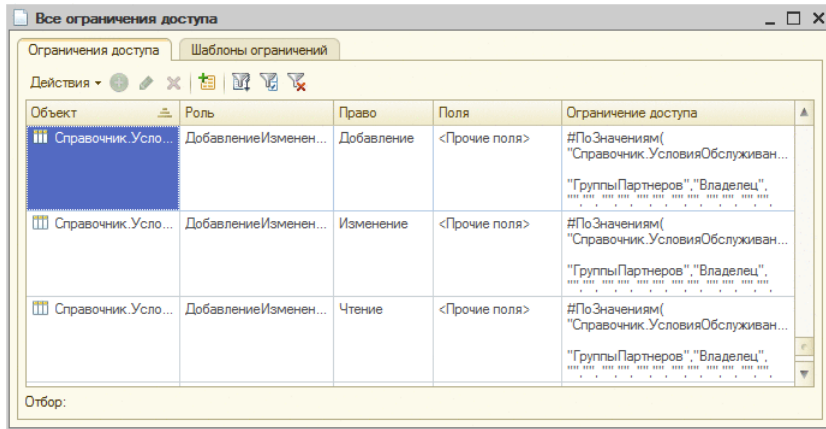


Рис. 78. Все ограничения прав доступа и шаблоны

На закладке **Ограничения доступа** можно просматривать все введенные ограничения доступа в общем списке (по всем ролям, объектам, правам, комбинациям полей).

Существует возможность добавлять ограничение доступа сразу для нескольких ролей, объектов, прав и комбинаций ролей.

Можно фильтровать список по различным критериям.

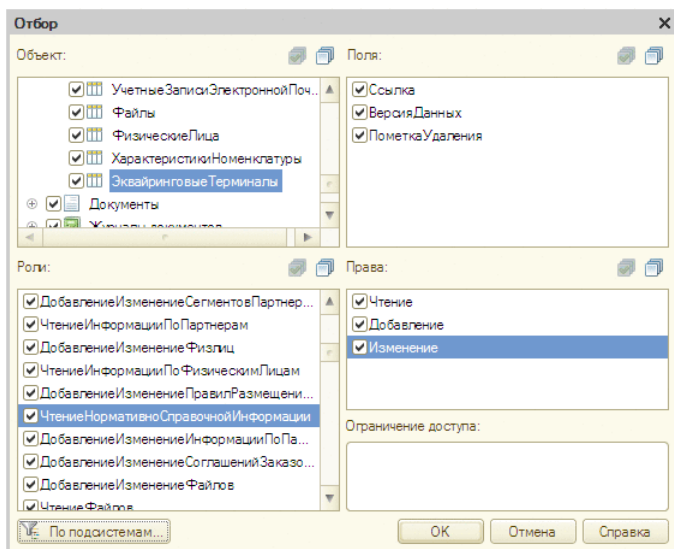


Рис. 79. Отбор ограничений доступа

Режим группового редактирования позволяет удалять выделенные в списке ограничения.

Существует возможность редактировать выделенные ограничения. При этом можно заменять состав полей и/или ограничение доступа.

Режим группового редактирования позволяет также копировать выделенные ограничения в другие роли.

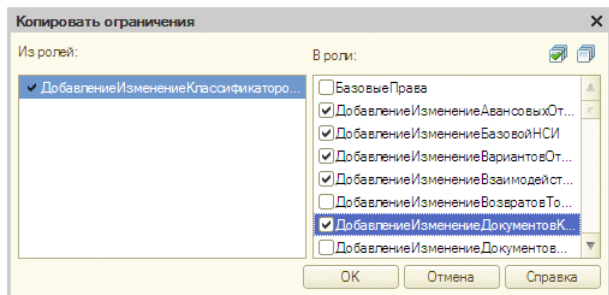


Рис. 80. Копирование ограничений

На закладке **Шаблоны ограничений** можно видеть все шаблоны ограничения доступа, присутствующие в прикладном решении, при этом из собственно текста шаблона в таблице отображаются только первые 10 строк, которые завершаются символом "...", если текст шаблона более 10 строк. В окне редактирования шаблона будет отображаться полный текст шаблона.

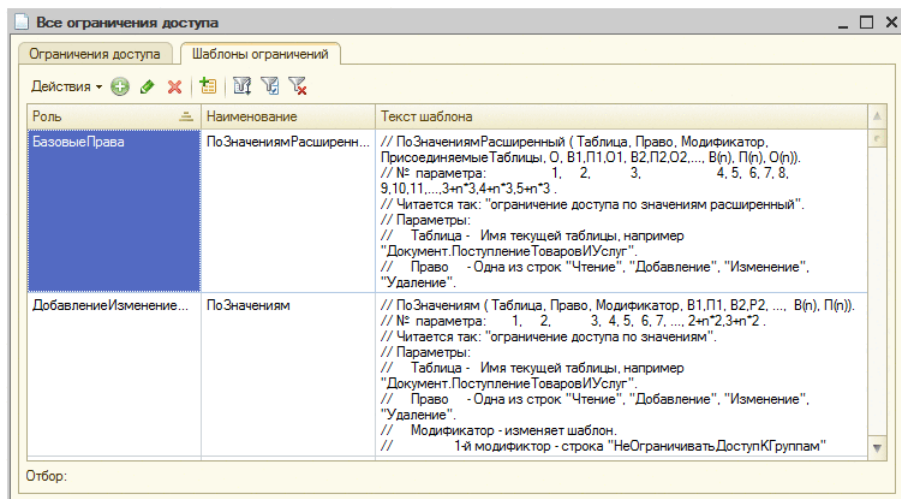


Рис. 81. Все шаблоны ограничения доступа

Существует возможность добавлять шаблон ограничения доступа сразу для нескольких ролей.

Имеется возможность отбирать необходимые шаблоны с помощью набора критериев, а также по значению текущей колонки.

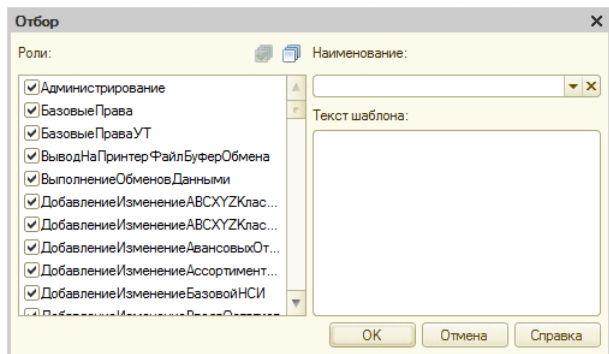


Рис. 82. Отбор шаблонов ограничения доступа

При необходимости имеется возможность выполнить копирование одного или нескольких шаблонов в другие роли.

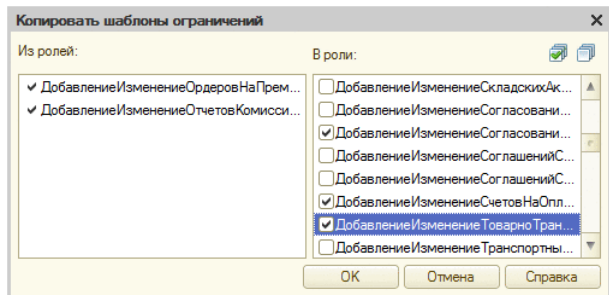


Рис. 83. Копирование шаблонов

Также редактор предоставляет возможность редактировать выделенные шаблоны. При этом можно изменять наименование и сам текст шаблона.

5.5.5. Общие реквизиты

Общий реквизит – это реквизит, добавляемый ко всем или многим объектам конфигурации. Общий реквизит может использоваться в двух сценариях:

- Как **общий реквизит**, т. е. для упрощения указания реквизита, который присутствует во всех или многих объектах конфигурации, в которых этот реквизит сохраняет свой смысл и тип. В качестве примера такого использования общего реквизита можно привести поле произвольного комментария в документах прикладного решения.
- Как составная часть **разделения данных** – специального механизма, позволяющего разделить все хранимые данные, а также работу прикладного решения на отдельные части. При этом у общего реквизита включается **разделение данных**. В качестве примера такого использования можно привести понятие абонента, когда в одной физической информационной базе могут независимо работать различные «владельцы» данных, при этом пользователи такого прикладного решения будут считать,

Для того чтобы создать общий реквизит, необходимо создать объект конфигурации **Общий реквизит**. Это можно сделать в режиме Конфигуратор обычным способом, то есть в окне конфигурации следует выбрать пункт **Общие**, далее **Общие реквизиты** и добавить новый объект.

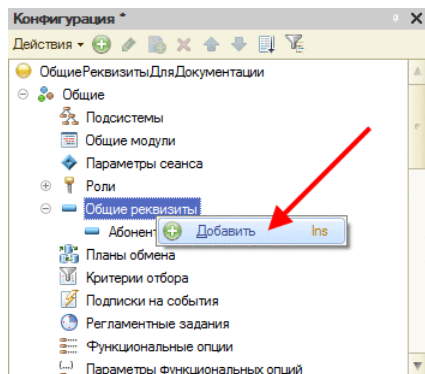


Рис. 84. Создание общего реквизита

В результате будет создан объект конфигурации **Общий реквизит**, который можно использовать для включения общего реквизита в необходимые объекты конфигурации.

Режим работы общего реквизита задается с помощью свойства **Разделение данных**. Если это свойство установлено в значение **Не использовать**, то созданный объект конфигурации будет использоваться только как реквизит, входящий в состав нескольких (или всех) объектов конфигурации. Если свойство установлено в значение **Разделять**, общий реквизит будет использоваться в качестве **разделителя** данных (см. [здесь](#)).

Список объектов конфигурации, в состав которых входит общий реквизит, определяется с помощью свойств **Состав** и **Автоиспользование** (или соответствующая закладка в окне **Дополнительно** (см. [здесь](#))).

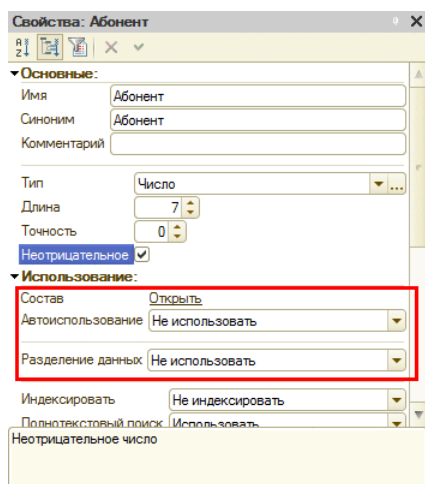


Рис. 85. Общий реквизит без разделения

Если свойство **Автоиспользование** установлено в значение **Использовать**, то созданный общий реквизит будет автоматически добавлен во все существующие объекты конфигурации (для которых возможно использование общих реквизитов), а также будет автоматически добавляться во все вновь создаваемые объекты конфигурации.

В том случае, если свойство **Автоиспользование** установлено в значение **Не использовать**, автоматического добавления реквизита происходить не будет, а для выбора объектов, в состав которых необходимо включить общий реквизит, следует воспользоваться свойством **Состав**.

Этим же свойством следует воспользоваться в том случае, если при автоматическом использовании общего реквизита существуют объекты, в которых общий реквизит добавлять не следует.

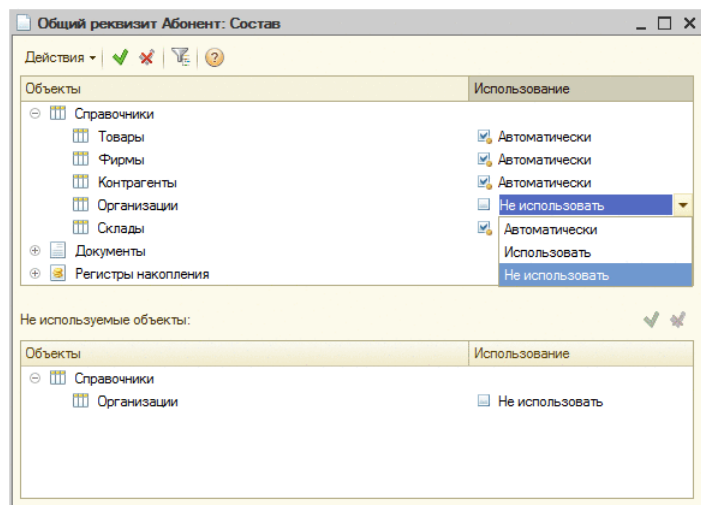


Рис. 86. Состав общего реквизита

Окно редактирования состава общего реквизита разделено на две части:

- в верхней части указаны все объекты конфигурации, которые могут входить в состав общего реквизита;
- в нижней части – объекты, настройки которых отличаются от настроек по умолчанию, определяемых значением свойства **Автоиспользование** общего реквизита:

- если свойство установлено в значение **Использовать**, то нижняя часть окна содержит список объектов, не входящих в состав общего реквизита;
- если свойство установлено в значение **Не использовать**, то нижняя часть окна содержит список объектов, входящих в состав общего реквизита.

Редактирование можно осуществлять как в верхней, так и в нижней части окна, при этом редактируемый объект конфигурации будет перемещаться между частями окна, в зависимости от значения в колонке **Использование**.

Для каждого объекта конфигурации колонка **Использование** может принимать одно из трех значений:

- **Автоматически** – означает, что назначение объекта конфигурации общему реквизиту зависит от значения свойства **Автоиспользование** общего реквизита:
- **Использовать** означает, что объект конфигурации **входит** в состав общего реквизита;
- **Не использовать** означает, что объект конфигурации **не входит** в состав общего реквизита.
- **Использовать** – означает, что объект конфигурации **входит** в состав общего реквизита вне зависимости от значения свойства **Автоиспользование**.
- **Не использовать** – означает, что объект конфигурации **не входит** в состав общего реквизита вне зависимости от значения свойства **Автоиспользование**.

Таким образом, с помощью редактора свойства **Состав** можно, например, выборочно исключать некоторые объекты из состава общего реквизита, несмотря на то, что для него установлено автоиспользование.

В состав общего реквизита (не находящегося в режиме разделения данных) могут входить следующие объекты конфигурации:

- справочники,
- документы,
- журналы документов,
- планы видов характеристик,
- планы счетов,
- планы видов расчета,
- бизнес-процессы,
- задачи,
- регистры сведений,
- регистры накопления,
- регистры бухгалтерского учета,
- регистры расчета,
- планы обмена,
- внешние источники данных.

При записи документа общему реквизиту журнала присваивается значение общего реквизита документа или значение **NULL**, если документ не входит в состав общего реквизита.

Общий реквизит может использоваться в ограничениях доступа к данным (см. [здесь](#)). Включать внешние источники данных в состав общего реквизита имеет смысл в том случае, когда общий реквизит является разделителем (см. [здесь](#)). С особенностями использования внешнего источника данных, входящего в состав разделителя, можно ознакомиться в соответствующем разделе (см. [здесь](#)).

Общий реквизит **может быть составного типа**.

СОВЕТ. Не следует использовать общие реквизиты для описания данных, которые являются частью бизнес-логики конкретных объектов.

Общие реквизиты отображаются в редакторе формы при редактировании форм объектов, входящих в состав общего реквизита, и могут быть размещены на форме.

5.5.6. Планы обмена

План обмена используется для реализации механизмов обмена данными. План обмена:

- содержит информацию об узлах, которые могут участвовать в обмене данными;
- определяет состав данных, которыми будет производиться обмен;
- указывает, следует ли задействовать механизм распределенной информационной базы при обмене.

В одном прикладном решении может существовать несколько планов обмена, каждый из которых может описывать свой порядок обмена данными. Например, если выполняется обмен данными с удаленными складами и удаленными офисами, то, скорее всего, будут существовать два плана обмена (один – для обмена со складами, другой – для офисов), поскольку состав данных, которыми производится обмен со складами, будет значительно «уже», чем состав данных, предназначенных для обмена с офисами.

Описание механизмов обмена данными см. [здесь](#).

5.5.7. Критерии отбора

Критерии отбора – одна из составляющих механизма отбора информации. С их помощью специалист, осуществляющий конфигурирование системы, создает predeterminedные правила отбора. В режиме 1С:Предприятие по этим правилам будет выполняться отбор информации в списках.

В окне редактирования объекта **Критерий отбора** нужно указать имя, синоним и комментарий.

Тип критерия отбора может быть любым из стандартных типов или определенных как объекты дерева конфигурации. Допускается включение реквизитов с составным типом, заданным как **СправочникСсылка**, **ДокументСсылка** и т. д., а также реквизитов с составным типом, определенным планом видов характеристик (**Характеристика...**).

При создании типа критерия отбора следует указать те типы, по которым требуется производить отбор. На закладке **Состав** по этому типу будет сформирован состав объектов конфигурации, содержащих данные, тип которых входит в тип критерия отбора. В списке требуется установить отметку для тех реквизитов, по которым будет производиться данный отбор.

Чтобы критерий отбора выполнял свои функции, в конфигураторе для него должен быть создан список (закладка **Состав**), состоящий из реквизитов справочников и документов. На состав списка не накладывается практически никаких ограничений: например, в отличие от графы журнала, для критерия отбора можно выбрать несколько реквизитов одного документа и реквизитов табличной части документа.

Критериев отбора может быть произвольное число, а каждый критерий может иметь несколько форм представления результатов отбора. Этот механизм полезен в случае поиска различной информации. Например, требуется отобрать все документы, в которых используется (в реквизитах и табличных частях) определенный контрагент. При этом можно учитывать и другие условия отбора информации (например, поиск ведется только среди проведенных документов или в определенном интервале дат и т. д.).

ПРИМЕЧАНИЕ. При открытии формы отбора следует передавать в нее параметр `Отбор` с установленным значением отбора (элемент `Значение`). Подробнее о параметрах формы см. [здесь](#).

Критерий отбора может иметь произвольное число форм для визуального представления результатов отбора. Для оперативности получения информации по отбору вызов формы можно разместить в пользовательском меню или на панели инструментов.

Если форм критерия отбора несколько, то в свойстве `Основная форма` указывается та форма, которая будет вызываться по умолчанию.

Если в конфигурации определено несколько подсистем, нужно выбрать ту, к которой относится данный критерий отбора. Можно указать несколько различных подсистем.

Для вызова формы критерия отбора система размещает соответствующую команду в панели навигации формы.

5.5.8. Подписки на события

Подписки на события позволяют назначать обработчики событий для одного объекта или группы объектов встроенного языка.

При добавлении новой подписки на событие, кроме общих свойств объектов конфигурации, следует указать источник события, само событие, обработчик которого назначается, и процедуру, являющуюся обработчиком этого события.

Источниками событий могут являться прикладные объекты, наборы записей регистров, менеджеры объектов конфигурации. Допускается как множественный выбор объектов, являющихся поставщиками событий, так и выбор всех объектов одного типа (например, все документы).

Выбор события осуществляется из выпадающего списка, причем список содержит те события, которые присутствуют во всех выбранных объектах. Если таких событий нет, список будет пуст.

Выбор обработчика события выполняется в окне, содержащем процедуры, которые могут быть назначены в качестве обработчика события. Такие процедуры должны удовлетворять следующим требованиям:

- процедура должна быть расположена в общем модуле;
- у общего модуля, в котором расположена процедура, должны быть заданы следующие свойства:
 - флажок `Глобальный` снят;
 - флажок `Клиент (обычное приложение)` установлен;
 - флажок `Клиент (управляемое приложение)` снят;
 - флажок `Сервер` установлен;
 - флажок `Внешнее соединение` установлен.
- количество параметров процедуры должно быть на единицу больше, чем количество параметров, которое имеет обработчик выбранного события (т. к. дополнительно к параметрам, передаваемым в обработчик события, передается еще и объект-источник данного события).

При наступлении указанного события выполняется следующая последовательность действий:

- сначала обрабатывается событие в самом объекте и вызывается обработчик события, определенный в модуле объекта или набора записей;
- если в процессе выполнения обработчика параметр `Отказ` установлен в значение `Истина` или вызвано исключение, действие прерывается;
- затем в произвольном порядке вызываются внешние обработчики, назначенные для данного события;
- если в процессе выполнения назначенного обработчика параметр `Отказ` установлен в значение `Истина` или вызвано исключение, действие прерывается.

В качестве источника в назначенный обработчик передается сам объект (набор записей), вызвавший событие.

Назначенные обработчики событий вызываются в том же контексте, что и действие, вызвавшее событие. Если выполнение назначенного обработчика нужно перенести на сервер, следует вызывать в коде обработчика процедуру общего модуля, исполняемую на сервере.

Назначение обработчиков событий доступно также и средствами встроенного языка. Для этого используются операторы `ДобавитьОбработчик` и `УдалитьОбработчик`.

У объектов, которые могут являться источниками событий, есть свойство `ДополнительныеСвойства` типа `Структура`, позволяющее хранить информацию между вызовами событий, например, новый или старый это объект.

5.5.9. Регламентные задания

5.5.9.1. Основные возможности механизма заданий

Основными возможностями механизма заданий являются:

- определение регламентных процедур на этапе программирования системы;
- выполнение заданных действий по расписанию;
- выполнение вызова заданной процедуры или функции асинхронно, т. е. без ожидания ее завершения;
- мониторинг хода выполнения заданий;
- управление заданиями (отмена, блокировка выполнения и др.);
- возможность ожидания завершения одного или нескольких заданий.

5.5.9.2. Фоновые задания

Механизм фоновых заданий реализуется средствами встроенного языка. Фоновые задания предназначены для выполнения прикладных задач асинхронно. Они могут порождать дочерние фоновые задания, например, для распараллеливания сложных вычислений по различным рабочим серверам кластера в клиент-серверном варианте работы.

Существует возможность ограничить выполнение фоновых заданий, имеющих одинаковые методы, по определенному прикладному признаку. Программное создание и управление фоновыми заданиями возможны из любого соединения пользователя с информационной базой системы «1С:Предприятие». Фоновое задание выполняется от имени пользователя, который его создал.

5.5.9.3. Регламентные задания

Регламентные задания представляют собой неотъемлемую часть конкретного прикладного решения и описываются на этапе конфигурирования.

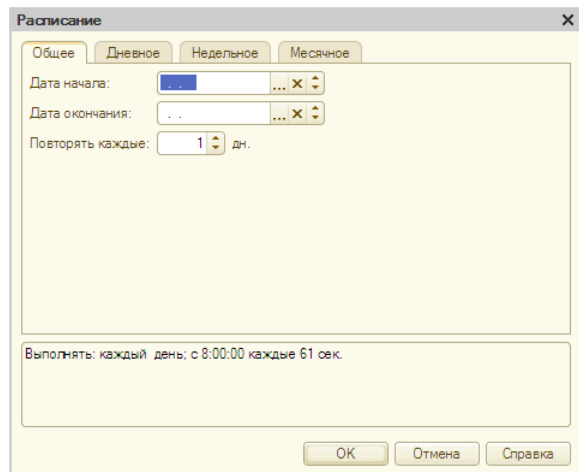


Рис. 87. Расписание фоновых заданий

Для каждого регламентного задания может быть задано расписание, в соответствии с которым регламентное задание будет автоматически запущено на исполнение. В системе «1С:Предприятие» поддерживаются однократные и периодические расписания. Можно задать дату начала и окончания выполнения, дневное, недельное и месячные расписания. Расписание можно задать как на этапе конфигурирования, так и на этапе выполнения (в режиме 1С:Предприятие).

В процессе запуска регламентное задание порождает фоновое задание, которое и выполняет реальную обработку. Регламентное задание может выполняться от имени заданного пользователя и имеет возможность перезапуска (например, в случае непредвиденного завершения работы).

В утилите администрирования клиент-серверного варианта работы автоматическое выполнение регламентных заданий может быть запрещено для конкретной информационной базы.

5.5.9.4. Особенности выполнения регламентных заданий

В клиент-серверном варианте работы запуск регламентных заданий по расписанию осуществляет менеджер кластера. Таким образом, даже если с информационной базой не установлено ни одного клиентского соединения, регламентные задания будут выполняться (при условии, что они не запрещены для конкретной информационной базы).

Особенность выполнения фоновых заданий в файловом и клиент-серверном вариантах см. [здесь](#).

5.5.10. Функциональные опции и параметры функциональных опций

5.5.10.1. Назначение

Функциональные опции позволяют разработчику описать возможности прикладного решения, которые можно оперативно включать или выключать на этапе внедрения и/или в процессе работы системы. Например, возможность работы с дополнительными свойствами товаров можно выделить в отдельную функциональную опцию. Тогда если отключить эту возможность, в интерфейсе прикладного решения «пропадут» все связанные (с дополнительными свойствами товаров) возможности.

Система способна автоматически учитывать состояние сделанных настроек – скрывать выключенные возможности, делая интерфейс приложения более ясным и понятным для пользователя.

При разработке возникают ситуации, когда значение функциональной опции должно зависеть от неких параметров, например, валютный учет ведется не у всех организаций. Для реализации такой зависимости служат [Параметры функциональных опций](#) – объекты, параметризующие функциональные опции.

5.5.10.2. На что влияют функциональные опции

Функциональные опции могут оказывать влияние:

- **На пользовательский интерфейс** – при выключении каких-либо функциональных опций система скрывает в пользовательском интерфейсе все элементы, относящиеся к ней. При этом затрагиваются следующие элементы интерфейса:
 - глобальный командный интерфейс;
 - реквизиты формы (в том числе колонки реквизита формы типа [ТаблицаЗначений](#) или [ДеревоЗначений](#));
 - команды формы;
 - отчеты, реализованные с помощью системы компоновки данных.

ВНИМАНИЕ! Если клиентское приложение работает с файловым вариантом информационной базы через веб-сервер, то изменение функциональной опции приведет к изменению пользовательского интерфейса только после перезапуска веб-сервера (перезапуск клиентского приложения не вызовет изменение пользовательского интерфейса).

- **На алгоритмы, написанные на встроенном языке** – имеется возможность программно получать значения функциональных опций и использовать их в различных условиях, например, для уменьшения объема вычислений.

ВНИМАНИЕ! Функциональные опции и их параметры не влияют на состав базы данных. Все таблицы и поля присутствуют в базе данных независимо от состояния функциональных опций.

5.5.10.2.1. Глобальный командный интерфейс

Влияние функциональных опций на глобальный командный интерфейс заключается в том, что система скрывает команды всех объектов, относящихся к выключенным опциям. Например, если значение функциональной опции [Закупки](#) равно значению [Ложь](#), то будут скрыты команды открытия раздела [Закупки](#), создания документа [ПриходТовара](#), открытия списка [ПриходТовара](#) и т. д.

В свою очередь, опция [Закупки](#) может учитывать значение параметра функциональной опции, например, [Организация](#). Изменяя с помощью методов встроенного языка значение этого параметра, можно изменять состояние функциональной опции, а, следовательно, и видимость элемента интерфейса.

Также следует учитывать следующие особенности формирования командного интерфейса:

- Команда будет исключена из командного интерфейса в том случае, если реквизит, являющийся параметром команды, отключен функциональной опцией.
- Команда будет исключена из командного интерфейса в том случае, если тип параметра команды отключен функциональной опцией. Если тип параметра команды составной, то команда становится недоступной тогда, когда отключаются все типы параметра.

5.5.10.2.2. Форма

В форме функциональные опции могут влиять на реквизиты и команды формы и (как следствие) изменять видимость связанных с ними элементов формы (поля и колонки – для реквизитов формы, кнопки – для команд формы). При разработке формы необходимо учитывать следующие особенности поведения системы:

- Отключается реквизит формы ссылочного типа, если объект конфигурации, образующий этот тип, отключен функциональной опцией. Реквизит формы составного типа отключается в том случае, если функциональные опции отключают все составляющие типы.

- Отключается реквизит формы типа <Вид>Объект (включая основной реквизит формы), если объект конфигурации, образующий этот тип, отключен функциональной опцией. Анализируются только непараметризуемые функциональные опции.

- Таблица формы будет отключена, если она отображает данные реквизита формы, отключенного функциональной опцией.

- В диалоге выбора типов (например, для полей ввода, связанных с реквизитами составного типа) отсутствуют типы, если объекты конфигурации, формирующие эти типы, отключены функциональной опцией. Информация о типах, отключенных функциональными опциями, кешируется на стороне клиента и очищается через 20 минут или во время вызова метода ОбновитьИнтерфейс().

ВНИМАНИЕ! В отличие от командного интерфейса, значения параметров функциональных опций устанавливаются только для конкретного экземпляра формы.

5.5.10.2.3. Система компоновки данных

Система компоновки данных в основном используется для построения отчетов. Функциональные опции влияют на состав данных, которые выводятся в отчет, и на состав настроек отчета, доступных пользователю. Например, если выключена функциональная опция Валютный учет, то в отчете, выводящем реестр документов Приход товара, будет отсутствовать колонка Валюта и Валютная сумма, а в настройках будет отсутствовать возможность отбора, группировки, сортировки и т. д. по полю Валюта.

Подробнее о влиянии функциональных опций на доступность полей в отчете, построенном на системе компоновки данных, см. [здесь](#).

5.5.10.2.4. Характеристики

Функциональные опции оказывают влияние на видимость полей формы, которые отображают значение характеристики объекта. Для этого необходимо включить в состав функциональной опции реквизит, хранящий значение характеристики.

Рассмотрим пример. Характеристики используются для справочника Товары, виды характеристик хранятся в плане видов характеристик Характеристики, а значения – в качестве ресурса регистра сведений ЗначенияХарактеристик. Ресурс входит в состав функциональной опции УчетХарактеристик.

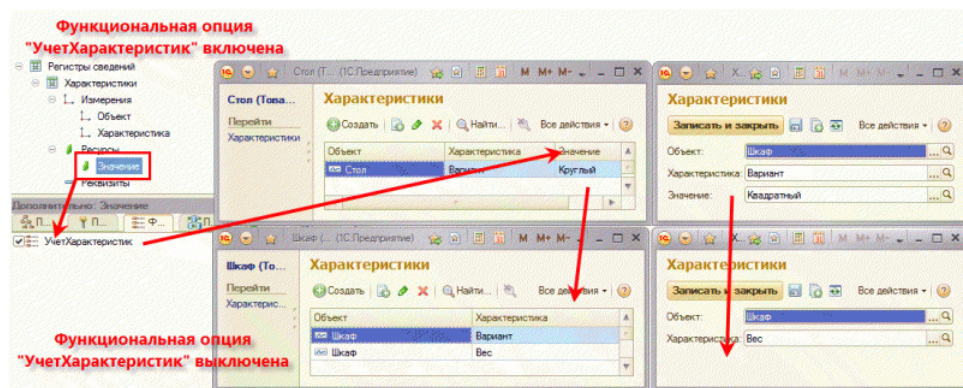


Рис. 88. Влияние функциональных опций на характеристики

В случае отключения функциональной опции УчетХарактеристик в формах отключается видимость полей (колонка Значение и поле Значение), отображающих значения характеристик, как показано на [рис. 88](#).

5.5.10.3. Общая схема работы

Механизм функциональных опций включает в себя два типа объектов метаданных: Функциональная опция и Параметры функциональных опций.

Функциональная опция представляет собой объект метаданных, который может непосредственно влиять на состав интерфейса приложения (если функциональная опция хранит свое значение в реквизите типа Булево). С помощью объектов этого типа можно скрыть элементы, которые относятся к недоступной функциональности. Например, опция Валютный учет может скрыть справочник Валюты, поле Валюта из документов, колонку Валютная сумма из отчетов. Источником значения функциональной опции является объект метаданных, выбранный в качестве свойства Хранение, например, это может быть константа.

В случае хранения значения функциональной опции в реквизите справочника или ресурсе регистра сведений требуется дополнительная информация, которая указывает на то, как именно выбрать значение опции. Для этой цели предусмотрен отдельный объект метаданных – Параметры функциональных опций.

Можно сказать, что параметры функциональных опций являются осями координат пространства значений функциональных опций. Причем один параметр функциональных опций может определять значение «своей» оси координат одновременно для множества функциональных опций.

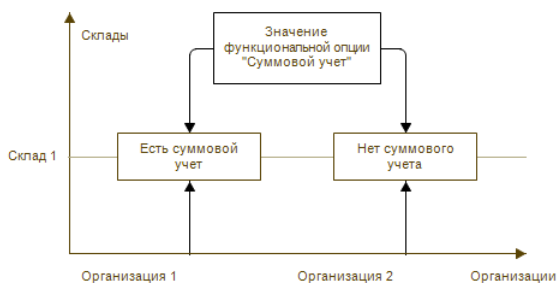


Рис. 89. Параметризуемая функциональная опция

Рассмотрим пример: допустим, суммовой учет зависит от склада, принадлежащего конкретной организации (см. [рис. 89](#)). В нашей информационной базе можно вести учет от имени разных организаций и на разных складах.

Для хранения значений функциональных опций создадим регистр сведений, где измерениями (осями координат) будут:

- Организация (соответствующего типа);
- Склад (соответствующего типа).

Ресурсом регистра сведений будет значение функциональной опции суммового учета.

Тогда общая структура конфигурации будет выглядеть следующим образом:

- Регистр сведений СуммовойУчет:

- измерение **Организация**;
- измерение **Склад**;
- ресурс **СуммовойУчет**, имеющий тип **Булево**.
- Параметр функциональных опций **Организация**. Свойство **Использование** указывает на измерение **Организация** регистра сведений **СуммовойУчет**.
- Параметр функциональных опций **Склад**. Свойство **Использование** указывает на измерение **Склад** регистра сведений **СуммовойУчет**.
- Функциональная опция **СуммовойУчет**. Свойство **Хранение** указывает на ресурс **СуммовойУчет** регистра сведений **СуммовойУчет**.

В результате для того, чтобы определить необходимость ведения суммового учета, нам необходимо в каждом конкретном случае указать значения параметров функциональных опций (**Организация** и **Склад**) и получить значение функциональной опции.

Так, в примере, показанном на [рис. 89](#), для **Организации 1** и **Склада 1** суммовой учет разрешен, а для **Организации 2** и **Склада 1** суммовой учет запрещен.

5.5.10.4. Взаимодействие с другими объектами

Функциональные опции могут быть назначены следующим объектам конфигурации:

- Подсистемы,
- Общие команды,
- Общие формы,
- Константы,
- Критерии отбора,
- Справочник,
- Документ,
- Журнал,
- План счетов,
- План видов характеристик,
- План видов расчета,
- Бизнес-процесс,
- Задача,
- Планы обмена,
- Отчет,
- Обработка,
- Регистр накопления,
- Регистр сведений,
- Регистр бухгалтерии,
- Регистр расчета,
- Команда,
- Реквизит объекта метаданных,
- Табличная часть,
- Реквизит табличной части,
- Признак учета,
- Признак учета субконто,
- Реквизиты адресации,
- Измерение регистра,
- Ресурс регистра.

Также функциональные опции могут влиять на видимость элементов формы.

5.5.10.5. Создание

5.5.10.5.1. Создание функциональной опции

Для того чтобы создать функциональную опцию, необходимо создать объект конфигурации **Функциональная опция**. Это можно сделать в режиме Конфигуратор обычным способом, то есть в окне конфигурации следует выбрать пункт **Общие**, далее **Функциональные опции** и добавить новый объект.

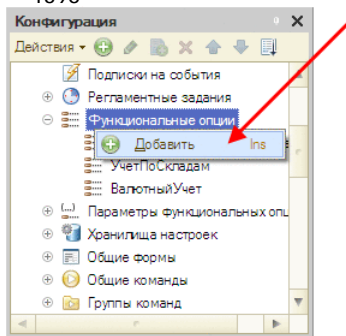


Рис. 90. Создание функциональной опции

В результате будет создан объект конфигурации **Функциональная опция**, который можно использовать для назначения функциональных опций другим объектам метаданных.

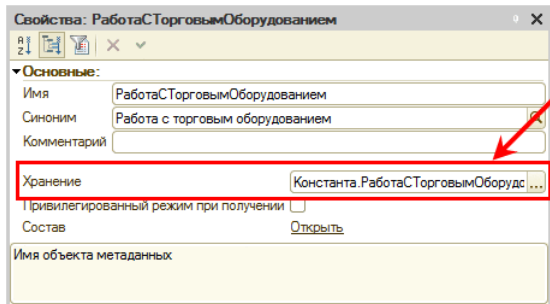


Рис. 91. Хранение значения функциональной опции

Кроме имени объект имеет обязательное для заполнения свойство – **Хранение**. В редакторе для него можно выбрать один из объектов, который будет являться источником значения опции. В список доступных объектов входят:

- константы,
- реквизиты справочников,
- ресурсы регистров сведений.

Ограничение на тип источника значения опции нет, но для управления интерфейсом пригодны только те функциональные опции, которые хранят свои значения в реквизитах, имеющих тип **Булево**. Значения функциональных опций с другими типами доступны только для анализа на встроенном языке.

Свойство **Привилегированный режим при получении** отвечает за способ получения (и кеширования) значения функциональной опции.

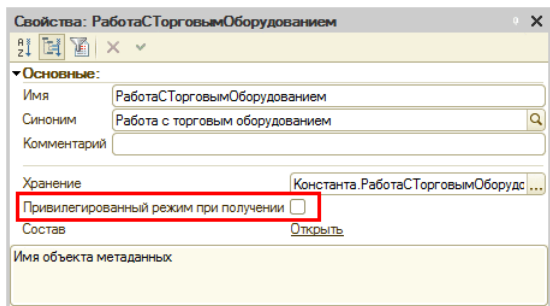


Рис. 92. Привилегированный режим при получении значения функциональной опции

Если данное свойство установлено, то значение функциональной опции получается в привилегированном режиме. Полученное значение кешируется для всех сеансов, связанных с данной информационной базой.

Если свойство **Привилегированный режим при получении** сброшено, то получение значения функциональной опции выполняется в обычном режиме. Кеширование выполняется для текущего сеанса. Кешируется как значение (если его удалось получить), так и признак невозможности получения значения (в том случае, если значение получить не удалось).

Кеш сбрасывается при изменении значений параметров сеанса.

СОВЕТ. Рекомендуется устанавливать свойство **Привилегированный режим при получении** для всех случаев, когда значение функциональной опции не содержит конфиденциальную информацию.

5.5.10.5.2. Создание параметра функциональных опций

Для того чтобы создать параметр функциональной опции, необходимо создать объект конфигурации **Параметры функциональных опций**. Это можно сделать в режиме Конфигуратор обычным способом, то есть в окне конфигурации следует выбрать пункт **Общие**, далее **Параметры функциональных опций** и добавить новый объект.

Кроме имени, параметр имеет обязательное свойство **Использование**. В нем указывается набор объектов, значения которых будут определять то, как следует выбирать значение функциональной опции. В список доступных объектов входят справочники и измерения регистра сведений. Для каждого параметра функциональных опций в данном списке можно выбрать один справочник (из всего перечня справочников) и по одному измерению каждого регистра сведений.

ВНИМАНИЕ! Нельзя использовать один и тот же объект метаданных в нескольких параметрах функциональных опций.

5.5.10.6. Использование

5.5.10.6.1. Назначение объектам метаданных

Объект метаданных (например, справочник) можно отнести к одной или нескольким функциональным опциям. Для этого служит свойство **Функциональные опции**, которое содержит ссылки на созданные в конфигурации функциональные опции.

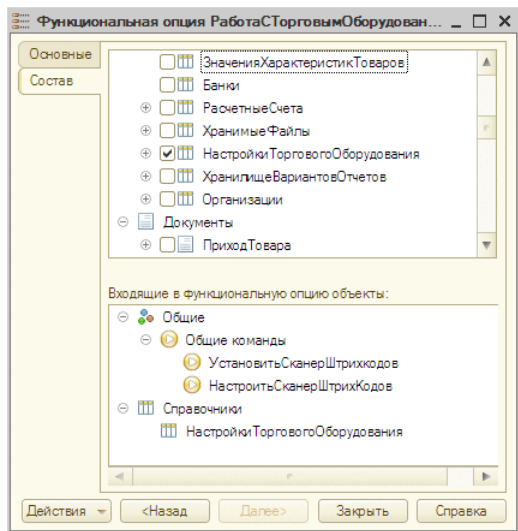


Рис. 93. Назначение функциональной опции объекту

Список доступных опций ограничен только теми опциями, для которых в свойстве **Хранение** назначен объект с типом значения **Булево**.

ВНИМАНИЕ! Если объекту не назначена ни одна функциональная опция, то он считается видимым всегда. В противном случае объект считается видимым, если хотя бы одна из назначенных ему функциональных опций является включенной (т. е. функциональные опции сочетаются «по ИЛИ»).

5.5.10.6.2. Назначение реквизитам и командам формы

Объекты, принадлежащие форме (**Реквизиты** и **Команды**), также можно задействовать в механизме функциональных опций.

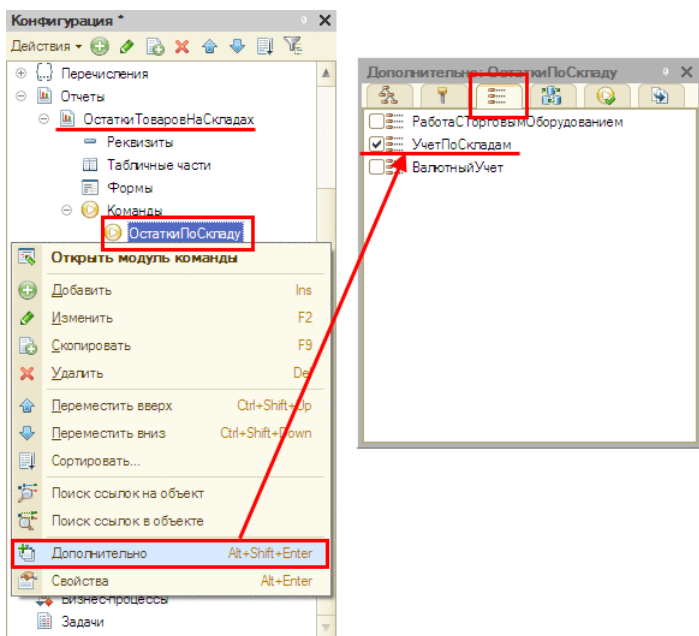


Рис. 94. Назначение функциональной опции команде

Сделать это можно в редакторе формы, установив свойство **Функциональные опции** для требуемого объекта.

Состояние функциональных опций будет влиять на отображение объектов формы точно так же, как это происходит в случае объектов метаданных. Например, в случае с командой, отключенной с помощью функциональной опции, будут убраны все связанные с ней кнопки.

Если реквизиту формы или команде не назначена ни одна функциональная опция, то реквизит формы или команда считается видимой всегда. В противном случае реквизит формы или команда считается видимой, если хотя бы одна из назначенных ему функциональных опций является включенной.

5.5.10.6.3. Использование в механизме ограничения доступа к данным

В условиях механизма ограничения доступа к данным (см. [здесь](#)) **Функциональные опции** могут использоваться точно так же, как и **Параметры сеанса** (см. [здесь](#)). Допустимо использовать только не зависящие от параметров опции, то есть те, которые привязаны к константам.

ВНИМАНИЕ! Системой контролируется уникальность имен между параметрами сеанса и функциональными опциями.

5.5.10.6.4. Определение значения функциональной опции

Значение функциональной опции определяется объектом, который указан в свойстве **Хранение**. В случае константы используется ее значение. Для опции, связанной с реквизитом справочника или ресурсом регистра сведений, – значения, хранящиеся в этих объектах. Для того чтобы найти конкретный объект, который хранит значение функциональной опции, необходима дополнительная информация – набор значений параметров функциональных опций.

Если опция хранится в реквизите справочника, параметр должен содержать ссылку на конкретный элемент справочника. Если опция хранится в ресурсе регистра сведений, должны быть указаны значения всех измерений регистра. В этом случае каждое измерение должно характеризоваться своим параметром.

Если для функциональной опции, имеющей тип **Булево**, заданы не все параметры, то выполняется сложение «по ИЛИ» всех значений с незадавленными параметрами. Например, если функциональная опция хранится в регистре сведений с измерениями **Организация** и **Склад** и задано только измерение **Организация**, то значение функциональной опции будет равно **Истина**, если хотя бы у одного из складов, перечисленных в измерении **Склад**, значение функциональной опции будет равно значению **Истина**.

Для функциональной опции, имеющей тип, отличный от **Булево**, ситуация с не полностью заданными параметрами приводит к генерации исключения.

Методы встроенного языка позволяют получить значение опции, как в зависимости от переданных параметров, так и для параметров, установленных для командного интерфейса или конкретной формы.

Если функциональная опция привязана к ресурсу периодического регистра сведений, то система использует срез последних для получения значения опции. Если требуется получать значение опции на какую-либо другую дату, необходимо указать значение для параметра функциональных опций **Период (Period)**, имеющий тип **Дата**, который будет использоваться как дата получения среза. Этот параметр не нужно создавать в метаданных. Он предоставляется системой автоматически.

При использовании параметризованных функциональных опций следует учитывать следующие особенности поведения:

- В формах списков колонка реквизита, связанного с параметризованной функциональной опцией, будет отображаться, если в информационной базе хранится хотя бы одно включенное значение данной функциональной опции.
- Если необходимо, чтобы при открытии формы реквизиты, связанные с функциональными опциями, были отключены по умолчанию, то нужно установить значения этих параметров в значения, отсутствующие в информационной базе (для справочников – пустая ссылка, для регистров сведений – значения измерений, для которых нет записей). В этом случае функциональная опция будет иметь значение **Ложь**.
- В том случае, когда в качестве параметра указана ссылка на группу (если типа параметра функциональной опции допускает создание групп), а не ссылка на элемент, поведение системы будет следующим:
 - если реквизит, в котором хранится значение функциональной опции, используется как для элемента, так и для группы, то значение функциональной опции будет определяться значением этого реквизита.
 - если реквизит, в котором хранится значение функциональной опции, не используется для группы, то при получении значения функциональной опции с помощью методов `ПолучитьФункциональнуюОпцию()`, `ПолучитьФункциональнуюОпциюФормы()` и `ПолучитьФункциональнуюОпциюИнтерфейса()` будет возвращено значение **NULL**. Если, параметризованная таким значением, функциональная опция оказывает влияние на пользовательский интерфейс, то система будет воспринимать ее как выключенную (функциональная опция будет иметь значение **Ложь**).
- Для командообразующих объектов метаданных возможно установить привязку к параметризованной функциональной опции. В командном интерфейсе команды таких объектов будут отображаться только в том случае, если есть хотя бы одна комбинация параметров функциональных опций, при которых значение функциональной опции равно **Истина**. Однако с помощью метода `УстановитьПараметрыФункциональныхОпцийИнтерфейса()` можно задать конкретные значения параметров функциональных опций, и тогда видимость команд будет определяться именно заданными параметрами.
- Динамический список автоматически использует функциональные опции, используемые формой. Если реквизиты, которые используются в запросе динамического списка, будут отключены при заданной комбинации параметров функциональных опций, данные по ним не будут выбраны и отображены в динамическом списке, а реквизит будет удален из списков доступных реквизитов в диалоге настройки отображения данных динамического списка (в режиме 1С:Предприятие).

5.5.10.7. Работа с функциональными опциями во встроенном языке

Методы глобального контекста `ПолучитьФункциональнуюОпцию()` и `ПолучитьФункциональнуюОпциюИнтерфейса()` возвращают значение функциональной опции. Разница между ними заключается в том, что первый метод позволяет указать набор параметров функциональных опций, а второй – возвращает значение функциональной опции исходя из параметров, заданных для командного интерфейса. В форме есть свой метод, который возвращает значение опции для параметров, указанных в рамках формы, – `ПолучитьФункциональнуюОпциюФормы()`.

Для обновления глобального командного интерфейса следует явным образом вызывать метод `УстановитьПараметрыФункциональныхОпцийИнтерфейса()`. Командный интерфейс будет обновлен с учетом нового состояния функциональных опций.

ПРИМЕЧАНИЕ. Если значение функциональной опции изменяется в базе данных, то автоматического обновления глобального командного интерфейса и открытых в это время форм не происходит. Для этого следует использовать метод `ОбновитьИнтерфейс()` после записи значений функциональных опций в базу данных.

Следует помнить, что установка параметров функциональных опций (и выполнение метода `ОбновитьИнтерфейс()`) приводит к следующим последствиям:

- для каждой формы вызывается закрытие всех вспомогательных форм (с вызовом соответствующих обработчиков);
- формы, отказавшиеся от закрытия, не закрываются;
- происходит обновление состава элементов основной формы;
- если на момент обновления интерфейса активной формой была основная, происходит отображение основной формы в соответствии с новым составом элементов;
- если на момент обновления интерфейса активной формой была вспомогательная форма, то:
 - будет выполнена команда открытия вспомогательной формы, если после обновления интерфейса она является доступной;
 - в противном случае обновляется состав элементов основной формы и выполняется ее отображение;
- если на момент обновления интерфейса активной формой была вспомогательная форма, открытая с помощью команды, не относящейся к панели навигации формы, то вместо этой формы будет обновлен состав элементов основной формы и выполнено ее отображение.

Для того чтобы обновить конкретную форму, следует либо заново открыть ее, либо вызвать метод `УстановитьПараметрыФункциональныхОпцийФормы()`, при этом вышеописанная последовательность действий отрабатывает только для той формы, в контексте которой вызвана установка параметров функциональных опций формы.

Параметры не обязательно указывать все сразу, можно изменить значение конкретного параметра или набора параметров выборочно. Но эффективнее осуществляется именно групповая установка значений одним вызовом.

Для получения значений параметров необходимо вызвать соответствующую функцию (`ПолучитьПараметрыФункциональныхОпцийИнтерфейса()` или `ПолучитьПараметрыФункциональныхОпцийФормы()`), которая вернет установленные параметры в виде структуры, где ключом будет выступать имя параметра.

При открытии форма автоматически использует параметры функциональных опций, установленных для командного интерфейса.

5.5.11. Определяемые типы

Определяемый тип – это специальный объект конфигурации, который предназначен для упрощения и повышения удобства внедрения подсистем при разработке прикладного решения. В качестве примера можно рассмотреть универсальную подсистему, предназначенную для хранения контактной информации. В состав этой подсистемы входит регистр сведений, который хранит эту информацию. В этом регистре есть измерение, которое указывает, для какого объекта хранится контактная информация. В качестве значения измерения может выступать контрагент, собственная организация, физическое лицо и т. д. Причем тип (с именем `СправочникиОрганизацийИФизическихЛиц`), описывающий этот объект хранения, может встречаться не только в качестве типа измерения регистра. В процессе внедрения прикладного решения состав такого типа может изменяться – в него могут добавляться новые справочники или из его состава могут исключаться «лишние» типы. Если вместо типа `СправочникиОрганизацийИФизическихЛиц` использовать составной тип в каждом месте, где используется такая сущность, то придется везде изменять состав типов. Это сложно и существует возможность забыть о каком-либо реквизите с нужным составным типом. Если использовать определяемый тип, то в процессе доработки прикладного решения будет необходимо изменить только состав определяемого типа (и, естественно, все фрагменты прикладного кода, где этот тип используется). Все реквизиты, для которых в качестве типа указан тип `СправочникиОрганизацийИФизическихЛиц`, будут изменены автоматически.

Определяемый тип обладает следующими особенностями:

- Может быть составным;
- Не может входить в состав типа значения плана видов характеристик;
- Не может входить в состав составного типа данных другого реквизита;
- Определяемый тип не может входить в состав другого определяемого типа;
- Определяемый тип не может входить в состав типа следующих объектов:
 - Признак учета плана счетов;

- Признак учета субконто плана счетов;
- Тип номера нумератора.

В состав определяемого типа могут входить следующие типы:

- **Ссылочные типы:** СправочникСсылка, ДокументСсылка, ПеречислениеСсылка, ПланВидовХарактеристикСсылка, ПланСчетовСсылка, ПланВидовРасчетаСсылка, БизнесПроцессСсылка, ТочкаМаршрутаБизнесПроцессаСсылка, ЗадачаСсылка, ПланОбменаСсылка, ВнешнийИсточникДанныхКубТаблицаИзмеренияСсылка;
- **Объектные типы:** СправочникОбъект, ДокументОбъект, ПланВидовХарактеристикОбъект, ПланСчетовОбъект, ПланВидовРасчетаОбъект, ОтчетОбъект, ОбработкаОбъект, БизнесПроцессОбъект, ЗадачаОбъект, ПланОбменаОбъект;
- **Наборы записей и связанные типы:** РегистрСведенийНаборЗаписей, РегистрНакопленияНаборЗаписей, РегистрБухгалтерииНаборЗаписей, РегистрРасчетаНаборЗаписей, ПерерасчетНаборЗаписей, ВнешнийИсточникДанныхКубНаборЗаписей, ПоследовательностьНаборЗаписей, ВнешнийИсточникДанныхКубМенеджерЗаписи, ВнешнийИсточникДанныхКубТаблицаИзмеренияОбъект;
- **Менеджеры объектов:** СправочникМенеджер, ДокументМенеджер, ПеречислениеМенеджер, ПланВидовХарактеристикМенеджер, ПланСчетовМенеджер, ПланВидовРасчетаМенеджер, ОтчетМенеджер, ОбработкаМенеджер, БизнесПроцессМенеджер, ЗадачаМенеджер, ПланОбменаМенеджер, ЖурналДокументовМенеджер, РегистрСведенийМенеджер, РегистрНакопленийМенеджер, РегистрБухгалтерииМенеджер, РегистрРасчетаМенеджер, КонстантаМенеджерЗначения, КритерийОтбораМенеджер, ВнешнийИсточникДанныхТаблицаМенеджер, ВнешнийИсточникДанныхКубМенеджер, ХранилищеНастроекМенеджер, ВнешнийИсточникДанныхКубТаблицаИзмеренийМенеджер, ПоследовательностьМенеджер;
- **Системные перечисления:** ВидДвиженияНакопления, ВидСчета, ВидДвиженияБухгалтерии, ГоризонтальноеПоложение, ВертикальноеПоложение, РежимИзмененияРазмера, ВидДанныхАнализа, МетодКластеризации, ТипЕдиницыИнтервалаВремениАнализаДанных, ТипЗаполненияТаблицыРезультатаАнализаДанных, ТипИспользованияЧисловыхЗначенийАнализаДанных, ТипИсточникаДанныхПоискаАссоциаций, ТипКолонкиАнализаДанныхДеревоРешений, ТипКолонкиАнализаДанныхКластеризация, ТипКолонкиАнализаДанныхОбщаяСтатистика, ТипКолонкиАнализаДанныхПоискАссоциации, ТипКолонкиАнализаДанныхПоискаПоследовательностей, ТипКолонкиМоделиПрогноза, ТипКерыРасстоянияАнализаДанных, ТипОтсеченияПравилАссоциации, ТипСтандартизацииАнализаДанных, ТипУпорядочиванияДереваРешений, ТипУпорядочиванияПравилАссоциацииАнализаДанных, ТипУпорядочиванияШаблоновПоследовательностейАнализаДанных;
- **Прочие типы:** Произвольный, Число, Строка, Дата, Булево, ХранилищеЗначения, УникальныйИдентификатор, ДинамическийСписок, ТаблицаЗначений, ДеревоЗначений, СписокЗначений, ФорматированнаяСтрока, ДвоичныеДанные, ТабличныйДокумент, ФорматированныйДокумент, ТекстовыйДокумент, NULL, ФиксированныйМассив, ФиксированноеСоответствие, ФиксированнаяСтруктура, КомпонентНастроекКомпоновкиДанных, ГрафическаяСхема, ГеографическаяСхема, Диаграмма, ДиаграммаГанта, Дендрограмма, ОписаниеТипов, Картинка, Цвет, Шрифт, СтандартнаяДатаНачала, СтандартныйПериод;

При выполнении проверки конфигурации или при проверке перед обновлением конфигурации базы данных проверяется, что состав определяемого типа допустим в каждом конкретном месте использования. Так, один и тот же определяемый тип может быть использован в качестве типа одного объекта конфигурации и не может – для другого.

Для описания типа на встроенном языке следует использовать конструкцию вида `ОпределяемыйТип. <ИмяОпределяемогоТипа>`.

ПРИМЕЧАНИЕ. Если требуется использовать прикладное решение с младшими версиями системы «1С:Предприятие» (младше версии 8.3.3), то следует удалить из конфигурации все определяемые типы.

5.5.12. Хранилища настроек

Для сохранения информации о настройках пользователя, которые должны сохраняться между сеансами работы, в платформе реализованы хранилища настроек.

Существует два вида хранилищ настроек:

- **Стандартное хранилище** – хранилище, используемое системой по умолчанию и хранящее данные в системных таблицах информационной базы.
- **Хранилища настроек** – специальные объекты метаданных, которые описывают хранение данных в некотором объекте информационной базы. Например, в этом объекте может быть описана работа с настройками, которые хранятся в справочнике.

Платформа использует следующие хранилища:

- **Системное хранилище** – в данное хранилище система сохраняет все возможные настройки, которые нужны для работы платформы. К данным настройкам относятся настройки размеров форм, настройки печати табличного документа и т. п. Полный перечень настроек, сохраняемых в системном хранилище, см. [здесь](#). В качестве системного хранилища настроек всегда используется стандартное хранилище настроек. То есть данные системного хранилища всегда сохраняются в системной таблице информационной базы.
- **Хранилище общих настроек** – данное хранилище предназначено для хранения различных настроек прикладного решения. Платформа самостоятельно не записывает в данное хранилище никаких настроек. Данное хранилище должен использовать разработчик из встроенного языка, для того чтобы выполнять сохранение/восстановление прикладных настроек пользователя.
- **Хранилище пользовательских настроек отчетов** – в данное хранилище помещаются пользовательские настройки отчетов.
- **Хранилище вариантов отчетов** – в данное хранилище помещаются варианты отчетов.
- **Хранилище настроек данных форм** – в это хранилище сохраняются данные форм. Этим хранилищем можно пользоваться, например, для сохранения реквизитов обработок. При этом можно выбрать индивидуальное хранилище для каждого отчета и обработки.
- **Хранилище пользовательских настроек динамического списка** – в это хранилище сохраняются пользовательские настройки динамического списка.

При разработке конфигурации имеется возможность определить собственные хранилища настроек для всех хранилищ, кроме системного хранилища. Для этого необходимо создать объект хранилища настроек в соответствующей ветке дерева метаданных и затем указать его в нужном свойстве конфигурации. Свойства объекта [Конфигурация](#) имеют те же имена, что и вышеперечисленные хранилища.

Данные хранилищ могут храниться как в системной таблице информационной базы, так и в некотором специальном объекте информационной базы, например, в справочнике или регистре сведений. Например, можно создать в конфигурации объект хранилища настроек и указать в свойстве конфигурации, что данное хранилище следует использовать для хранения настроек отчетов. Таким образом, настройки отчетов будут сохраняться не в системной таблице, а в некотором объекте, например, в справочнике, что дает возможность организовать работу с едиными настройками отчетов, реализовать систему прав, обмен настройками и т. п.

Создавать собственное хранилище имеет смысл в тех случаях, когда необходима особая структура хранения настроек, необходимы специальные механизмы управления настройками, требуется обмен настройками в рамках распределенной базы данных (см. [здесь](#)) и других аналогичных случаях.

ПРИМЕЧАНИЕ. Для хранения настроек рекомендуется выбирать такие объекты, для которых системой поддерживается способ идентификации, при котором идентифицирующий реквизит можно преобразовать в строку и обратно без потери данных. В качестве примера можно привести справочник и стандартный реквизит `Код`, уникальный во всем справочнике.

5.5.12.1. Общие принципы работы хранилища настроек

Объект метаданных [ХранилищеНастроек](#) предназначен для обеспечения хранения прикладных настроек конфигурации. За счет реализации обработчиков событий и создания форм объекта выполняется модификация механизма работы с настройками таким образом, что изменяется место хранения настроек (вместо системных таблиц используются объекты конфигурации, которые создал разработчик) и визуальные механизмы работы с настройками.

В конфигурации может быть определено произвольное количество хранилищ настроек.

Хранилище настроек может использоваться как только для программной работы, так и для программной и интерактивной работы. В первом случае для обеспечения необходимой функциональности требуется обязательная реализация обработчиком модуля объекта [ХранилищеНастроек](#):

- **ОбработкаСохранения** – содержит реализацию метода `Сохранить()`. В данном обработчике необходимо выполнить сохранение настройки в некоторый объект. Например, в

- **ОбработкаЗагрузки** – содержит реализацию метода `Загрузить()`. В данном обработчике необходимо получить настройки из некоторого объекта. Например, из элемента справочника.

ВНИМАНИЕ! Если не реализовать тот или иной обработчик, то будет недоступно выполнение действия, которое обработчик реализует. Например, если не реализовать обработчик `ОбработкаСохранения`, будет недоступно сохранение настроек.

В процессе разработки хранилища разработчик самостоятельно определяет, каким образом будет идентифицироваться объект хранилища, тем самым определяя тип параметра. Например, если настройки сохраняются в справочнике, то в качестве ключа настройки можно использовать поле `Код` или значение `Ссылка` (элемента справочника).

Если для работы с настройками требуется интерактивность (формы сохранения и восстановления настроек), то необходимо реализовать формы сохранения и восстановления настроек и заполнить соответствующие свойства объекта `ХранилищеНастроек` (`Форма сохранения` и `Форма загрузки`).

ВНИМАНИЕ! Реализация форм сохранения и восстановления настроек является обязательной для выполнения интерактивных операций. Программное сохранение и восстановление настроек возможно и без реализации этих форм.

Когда пользователь применяет команды сохранения или загрузки настроек, система получает соответствующую форму объекта хранилища настроек и отображает ее на экране. Например, при сохранении настроек отчета система будет использовать форму сохранения того объекта метаданных, который указан в качестве хранилища настроек отчетов (непосредственно у самого отчета или у всей конфигурации). При этом форме будут переданы параметры, более подробную информацию о которых можно получить в описании объекта `ОписаниеНастроек` в синтаксис-помощнике.

При создании формы с помощью конструктора необходимые параметры будут автоматически добавлены в список параметров формы.

В формах следует использовать переданные параметры и соответствующим образом фильтровать список настроек. Так, следует отображать только настройки для указанного в параметре `КлючОбъекта` объекта настройки (например, отчета).

Если пользователь выбрал настройку, то результатом работы формы должно быть значение типа `ВыборНастроек`. В этом значении, в свойстве `КлючНастроек`, должен находиться ключ выбранной настройки (например, код элемента справочника или другой, идентифицирующий настройку параметр), а в свойстве `ДополнительныеСвойства` – дополнительная информация, которую пользователь мог указать в форме:

`Закреть(Новый ВыборНастроек(КлючСохраняемойНастройки));`

[Копировать в буфер обмена](#)

ВНИМАНИЕ! Не поддерживается сохранение настройки в стандартном хранилище настроек, если длина ключа объекта превышает 256 символов; длина ключа настроек превышает 128 символов; длина имени пользователя превышает 64 символа.

5.5.12.2. Создание объекта метаданных

Для создания `Хранилища настроек` следует создать одноименный объект конфигурации. Это можно сделать в ветке `Общие`, пункт `Хранилища настроек`.

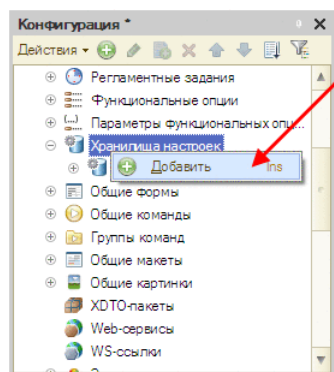


Рис. 95. Создание хранилища настроек

5.5.12.3. Стандартное хранилище настроек

Стандартное хранилище настроек представляется во встроенном языке объектом `СтандартноеХранилищеНастроекМенеджер`. Данный объект реализует тот же набор методов, что и объект `ХранилищеНастроекМенеджер`, и дополнительно реализует методы:

- `ПолучитьСписок()` – метод получения списка настроек для выбранного объекта настройки;
- `Удалить()` – удаление определенной настройки выбранного объекта настройки.

Стандартное хранилище сохраняет настройки в системных таблицах информационной базы.

В качестве ключа объекта настройки и в качестве ключа настройки системное хранилище настроек принимает строку.

В качестве настроек системное хранилище принимает любое значение, которое может быть помещено в хранилище значений.

ПРИМЕЧАНИЕ. При использовании методов `Сохранить()`, `Удалить()` и `УстановитьОписание()` объекта `СтандартноеХранилищеНастроекМенеджер` следует учитывать, что если объект, к которому относятся настройки (например, форма), в текущем сеансе уже использовался, то изменения произойдут только в следующем сеансе.

СОВЕТ. При существенном программном изменении настроек (например, при копировании настроек от одного пользователя другому) рекомендуется предложить пользователю выполнить перезапуск клиентского приложения.

5.5.12.4. Сохранение настроек форм

Разработчик может управлять сохранением данных форм в настройках. Для этого при разработке формы ему необходимо воспользоваться следующими свойствами формы:

- **Сохранение данных формы в настройках** – при помощи данного свойства разработчик формы может включить возможность сохранения данных формы (с возможностью выбора настройки, в которую будут сохраняться данные). Если у формы включена необходимость сохранения, то форма предоставляет команды сохранения/загрузки настроек.
- **Автоматическое сохранение данных в настройках** – указывает необходимость автоматического сохранения настроек при закрытии формы и восстановлении при открытии формы. При этом не важно, используется или нет список настроек.

Если для формы установлена возможность сохранения данных, то необходимо указать, какие реквизиты формы должны сохраняться (колонок `Сохранение` на закладке `Реквизиты` редактора формы).

При сохранении настроек в качестве ключа объекта используется полное имя формы. В настройках сохраняется объект типа `Соответствие`, в котором в качестве ключей находятся пути к сохраняемым реквизитам, а в качестве значений – их (реквизитов) значения.

5.5.12.5. Сохранение настроек отчетов

У объектов [Отчет](#) и [Внешний отчет](#) имеются свойства метаданных [Хранилище вариантов](#) и [Хранилище настроек](#). В данных свойствах указывается, в какие хранилища нужно сохранять варианты и настройки отчета соответственно. Если хранилища не указаны, то используются хранилища, указанные в свойствах конфигурации. Если в свойствах конфигурации также не указаны конкретные объекты [ХранилищеНастроек](#), то используется системное хранилище.

Форма отчета предоставляет команды сохранения и загрузки вариантов и настроек отчетов.

При необходимости сохранять в настройках компоновки данных или пользовательских настройках компоновки данных некоторую дополнительную информацию можно воспользоваться свойствами [ДополнительныеСвойства](#) объектов [НастройкиКомпоновкиДанных](#) и [ПользовательскиеНастройкиКомпоновкиДанных](#). Свойство [ДополнительныеСвойства](#) представляет собой объект типа [Структура](#).

5.5.12.6. Сохранение настроек динамических списков

Пользовательские настройки динамических списков сохраняются в хранилище, которое указано в свойствах конфигурации ([ХранилищеПользовательскихНастроекДинамическихСписков](#)). Если в свойствах конфигурации не указано хранилище настроек, то используется системное хранилище.

Возможностью автоматического сохранения пользовательских настроек динамического списка управляет свойство [Автоматическое сохранение пользовательских настроек](#) реквизита формы типа [ДинамическийСписок](#). Когда свойство установлено в значение [Истина](#), пользовательские настройки автоматически сохраняются при закрытии формы и загружаются при ее открытии.

Форма динамического списка предоставляет команды сохранения и загрузки пользовательских настроек. Также динамический список предоставляет команду для установки стандартных настроек. В случае выполнения данной команды в динамический список попадут настройки, которые находятся в свойстве [Список.КомпоновщикНастроек.Настройки](#).

Для динамического списка возможен вызов события [ПриОбновленииСоставаПользовательскихНастроекНаСервере](#) в следующих случаях:

- При открытии формы динамического списка;
- После завершения редактирования пользовательских настроек, если изменился состав настроек.

5.5.12.7. Порядок разработки хранилища настроек

Далее приводится рекомендованный порядок разработки хранилищ настроек:

1. Определяется, какое хранилище (см. начало раздела) будет использоваться. Например, будет реализовано хранилище настроек данных форм конфигурации.
2. Определяется перечень объектов метаданных, которые будут использовать это хранилище, и определяется перечень хранимой в хранилище информации, ее структура и типы. Эта информация поможет нам правильно выбрать объект метаданных, в данных которого будут храниться наши настройки.
3. На основании информации из п. 2 создается объект (и его структура), который будет хранить настройки. Пусть наши настройки хранятся в элементах справочника. Так как в нашем случае структура сохраняемых данных крайне разнородна, то нет смысла реализовывать отдельный набор реквизитов для хранения настроек каждой формы. Поэтому настройки будут храниться в реквизите справочника типа [ХранилищеЗначения](#).
4. Создается объект типа [ХранилищеНастроек](#), и для него выполняется реализация форм сохранения и загрузки. Тем самым обеспечивается интерактивность при сохранении и восстановлении настроек.
5. Для созданного объекта [ХранилищеНастроек](#) выполняется реализация обработчиков событий, связанных с сохранением и восстановлением настроек. Если не выполнить эту операцию, то не будут происходить собственно операции чтения/записи настроек. Для решения этой задачи нужно реализовать обработчики событий [ОбработкаСохранения](#) и [ОбработкаЗагрузки](#) в модуле созданного объекта типа [ХранилищеНастроек](#).
6. В объектах, выделенных в п. 2 (или в свойствах конфигурации), заполняются соответствующие свойства, тем самым указывая объектам, в каких хранилищах будут храниться настройки. В нашем примере необходимо заполнить свойство конфигурации [Хранилище настроек данных форм](#) ссылкой на объект, созданный на шаге 4.
7. При необходимости выполняется реализация обработчиков событий, связанных с сохранением и восстановлением настроек в тех прикладных объектах, где это необходимо.

5.5.12.8. Работа с хранилищем настроек из встроенного языка

В данном примере рассматривается копирование настроек текущего пользователя другим пользователям системы. В качестве источника настроек используется [ХранилищеСистемныхНастроек](#).

ПРИМЕЧАНИЕ. Пример, приведенный ниже, не является законченным. Он предназначен для демонстрации работы с стандартным хранилищем настроек.

Список имен пользователей передается параметром процедуры [СкопироватьНастройки\(\)](#) в виде массива.

[Копировать в буфер обмена](#)

```
Процедура СкопироватьНастройки(СписокПользователей)
ОписаниеНастроек = Новый ОписаниеНастроек;
ВыборкаНастроек = ХранилищеСистемныхНастроек.Выбрать();
Пока ВыборкаНастроек.Следующий() Цикл
    Для каждого Пользователь Из СписокПользователей Цикл
        ОписаниеНастроек.Представление = ВыборкаНастроек.Представление;
        ХранилищеНастроек.Сохранить(ВыборкаНастроек.КлючОбъекта, ВыборкаНастроек.КлючНастроек, ВыборкаНастроек.Настройки, ОписаниеНастроек, Пользователь);
    КонечЦикла;
КонечЦикла;
КонечПроцедуры
```

5.5.13. Общие формы

Механизм общих форм позволяет использовать формы, доступные из любого модуля текущей конфигурации. Подробно о порядке работы с редактором форм см. [здесь](#).

Если необходимо разместить команду открытия общей формы в командном интерфейсе, то сделать это можно с помощью свойства [Использовать стандартные команды](#). Команда открытия общей формы будет размещена в командном интерфейсе тех подсистем, которым принадлежит общая форма. Если общая форма входит в состав функциональной опции и функциональная опция отключена, то стандартная команда общей формы не отображается в командном интерфейсе.

Если создается общая форма, которая будет использоваться в качестве формы отчета, настроек отчета или варианта, то не рекомендуется для такой формы устанавливать свойство [Использовать стандартные команды](#).

5.5.14. Общие команды

В данной ветке разработчик может создавать команды, не имеющие объектной специфики или служащие для выполнения действий с объектами, которые не используют стандартные команды.

Описание командного интерфейса см. [здесь](#). Описание команд см. [здесь](#).

5.5.15. Группы команд

В данной ветке разработчик может создавать собственные группы команд. Созданная группа будет размещаться в той части командного интерфейса, который определен свойством команды [Категория](#). Группа команд может быть размещена:

- в панели навигации,
- панели навигации формы,

- панели действий,
- командной панели формы.

Описание командного интерфейса см. [здесь](#). Описание команд см. [здесь](#).

5.5.16. Общие макеты

Механизм общих макетов (печатных форм, форм отчетов, справочных данных и т. д.) позволяет создавать шаблоны печатных форм, доступные из любого модуля текущей конфигурации.

Подробнее о порядке работы с редактором табличных документов см. [здесь](#).

5.5.17. Общие картинки

Конфигуратор позволяет включать в конфигурацию графические изображения – картинки. Картинки можно размещать в некоторых элементах управления, в формах, в макетах, а также обращаться к ним при помощи встроенного языка системы «1С:Предприятие».

Если картинку планируется использовать в качестве пиктограммы в меню, панели инструментов, табличном документе и т. п., важно задать ей правильный размер, чтобы она отображалась без искажений.

Рекомендуемый размер картинок:

- для пиктограмм – не более **16x16** точек;
- для использования в табличном поле – не более **14x14** точек;
- в качестве картинки кнопки выбора поля редактирования – не более **9x9** точек;
- для картинки, которая будет использована в качестве представления подсистемы – не более **48x48** точек.

СОВЕТ. Если картинку предполагается использовать в нескольких местах, ее размер разумно ограничить меньшим значением.

Для работы с картинками предназначено окно **Библиотека картинок**. Для его вызова в окне **Конфигурация** нужно указать ветвь **Общие картинки** и в контекстном меню выбрать пункт **Все картинки**.

На экран выводится окно ведения списка картинок (см. [рис. 96](#)).

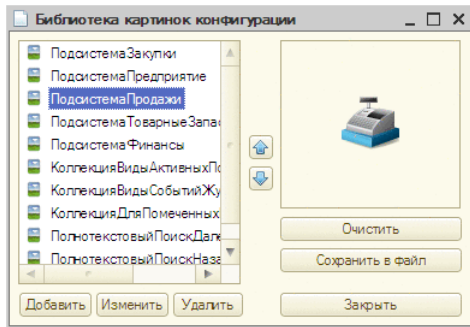


Рис. 96. Библиотека картинок

Для добавления новой картинки следует нажать кнопку **Добавить**. На экран выводится окно, в котором можно выбрать картинку из файла или открыть редактор картинок и создать новую картинку, а также выбрать или изменить для картинки прозрачный фон. Для выбора готовой картинки нужно нажать кнопку **Выбрать из файла** и выбрать файл, в котором расположена подготовленная заранее картинка или их набор (см. [здесь](#)). Система «1С:Предприятие» позволяет использовать картинки форматов **bmp**, **gif**, **jpg**, **png**, **tiff**, **ico**, метафайлы (**wmf**, **emf**) или наборы картинок в zip-архиве (см. [здесь](#)). Для картинок формата **gif** поддерживаются расширения **NETSCAPE2.0** и **ANIMEXTS1.0**. Другие расширения не поддерживаются и поведение системы в этом случае не определено.

СОВЕТ. Для картинок, которые будут использоваться в качестве пиктограмм в интерфейсе (см. рекомендованные размеры картинок в начале раздела), рекомендуется использовать форматы, поддерживающие сжатие без потерь (**png** и **gif**), для того чтобы избежать искажения и минимизировать трафик между клиентом и сервером.

Нужно задать имя, по которому данная картинка будет выбираться средствами встроенного языка.

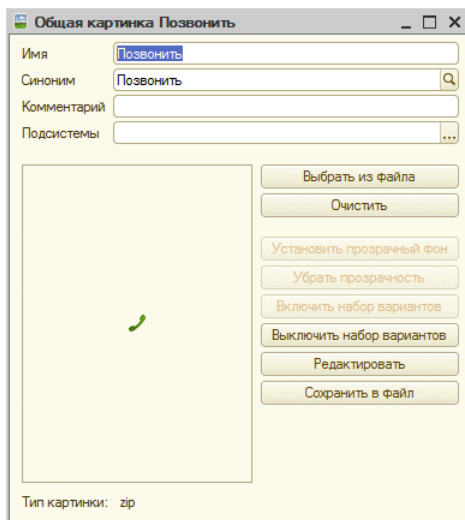


Рис. 97. Свойства картинки

Чтобы картинка при вставке хорошо вписывалась в элемент управления или форму, желательно установить ей прозрачный фон. Для этого в качестве цвета фона при редактировании картинки достаточно выбрать любой не используемый в картинке цвет, сформировать изображение и сохранить картинку. Для существующей картинки можно указать любой цвет. Благодаря установленной прозрачности данного цвета сквозь него становятся видны детали той части формы, которую закрывает область картинки.

Для установки прозрачного фона нужно нажать кнопку **Установить прозрачный фон**. Указатель мыши изменит вид. Подвести курсор к той части картинки, цвет которой нужно сделать прозрачным, и щелкнуть левой кнопкой мыши. Выбранный цвет становится прозрачным.

Для снятия прозрачности следует нажать кнопку [Убрать прозрачность](#).

Кнопки [Установить прозрачный фон](#) и [Убрать прозрачность](#) доступны только для картинок форматов **bmp**, **jpg** и **tiff**. Для картинок в остальных форматах данные кнопки недоступны.

Подобные действия можно выполнить с помощью ссылки [Открыть](#) в свойстве [Картинка](#). На экран выводится окно выбора картинки.

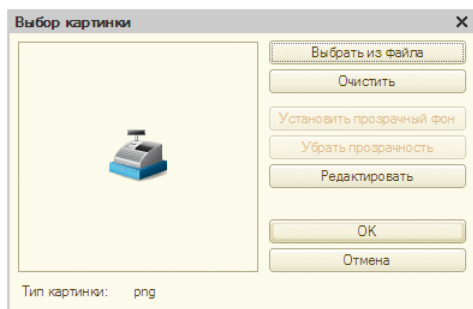


Рис. 98. Выбор картинки

Для редактирования картинки нужно нажать кнопку [Редактировать](#). Запускается редактор картинок (подробнее см. [здесь](#)).

Кнопки [Включить набор вариантов](#) и [Выключить набор вариантов](#) позволяют работать с картинками, предназначенными (в основном) для мобильных конфигураций и состоящие из набора картинок. Подробнее про такие картинки см. [здесь](#). Нажатие кнопки [Выключить набор вариантов](#) приводит к тому, что из набора картинок удаляются все картинки, кроме базовой картинки (картинки для разрешения ldpi). Если в наборе нет картинки для разрешения ldpi, то набор становится пустым. Если в наборе есть картинка для разрешения ldpi, и она установлена как контурная, то признак шаблонности сбрасывается. Нажатие кнопки [Включить набор вариантов](#) приводит к тому, что текущая картинка становится базовой картинкой (для разрешения ldpi). При необходимости отредактировать одну из картинок в наборе, следует нажать кнопку [Редактировать](#). В этом случае будет открыт диалог, в котором надо выбрать одну из картинок набора, которую необходимо редактировать.

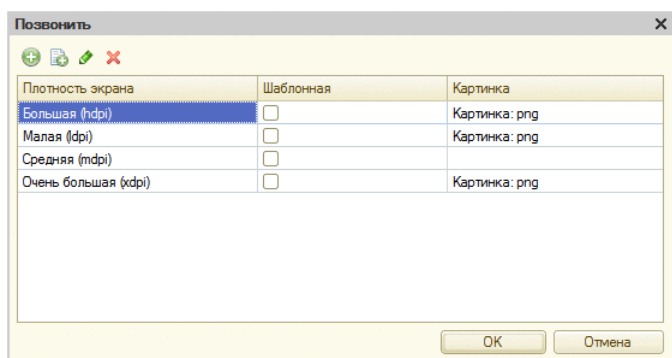


Рис. 99. Редактирование набора картинок

Если картинка для какого-либо разрешения не задана, поле ввода, соответствующее картинке, будет отображать пустым (как для картинки [Средняя \(mdpi\)](#) на [рис. 99](#)). Нажатие кнопки [...](#) в соответствующем поле диалога редактирования набора картинок, приведет к открытию диалога выбора картинки ([рис. 98](#)). Нажатие кнопки [X](#) приведет к очистке соответствующей картинке диалога. Редактирование отдельной картинки набора не отличается от редактирования отдельной картинки.

5.5.18. XDTO-пакеты

Механизм XDTO является универсальным способом представления данных для взаимодействия с различными внешними источниками данных и программными системами. Подробнее об использовании механизма XDTO см. [здесь](#).

5.5.18.1. Импорт схемы XML в глобальную фабрику XDTO

Для того чтобы импортировать схему XML из файла **.xsd** в глобальную фабрику XDTO, следует выделить в дереве конфигурации ветку XDTO и выполнить команду контекстного меню [Импорт XML-схемы...](#)

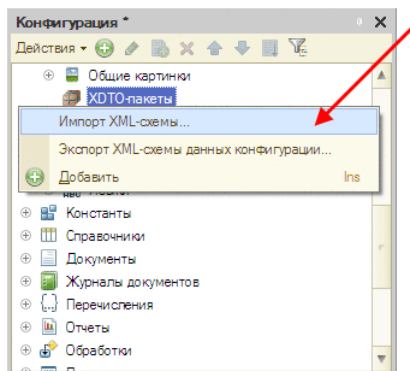


Рис. 100. Импорт XML-схемы

После указания требуемого файла **.xsd** будет выполнена проверка существования в дереве конфигурации пакетов XDTO, пространства имен которых совпадают с импортируемыми из файла. Если такие пакеты существуют, то будет отображен список этих пакетов и предложено указать те пакеты, которые должны быть обновлены (по умолчанию существующие пакеты не обновляются).

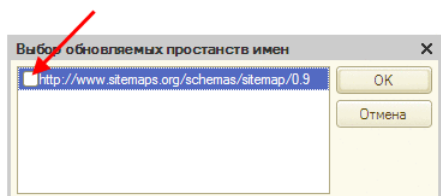


Рис. 101. Выбор пространств имен

После этого будет выполнен импорт, в результате которого новые пакеты XDTO будут добавлены в дерево конфигурации, а пакеты, отмеченные для обновления, обновлены.

5.5.18.2. Экспорт схемы XML-данных конфигурации

Для того чтобы экспортировать схему XML, соответствующую типам данных конфигурации (без учета пакетов XDTO, созданных в дереве конфигурации), в файл `.xsd`, следует выделить в дереве конфигурации ветку XDTO и выполнить команду контекстного меню [Экспорт XML-схемы данных конфигурации...](#)

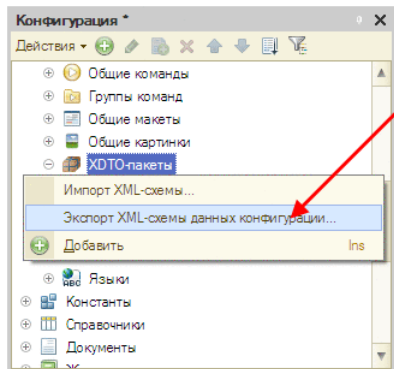


Рис. 102. Экспорт схемы конфигурации

После выбора каталога и указания имени файла будет выполнен экспорт схемы XML в указанный файл.

5.5.18.3. Экспорт схемы XML-пакета XDTO

Для того чтобы экспортировать схему XML, соответствующую существующему пакету XDTO, в файл `.xsd`, следует выделить в дереве конфигурации требуемый пакет XDTO и выполнить команду контекстного меню [Экспорт XML-схемы...](#)

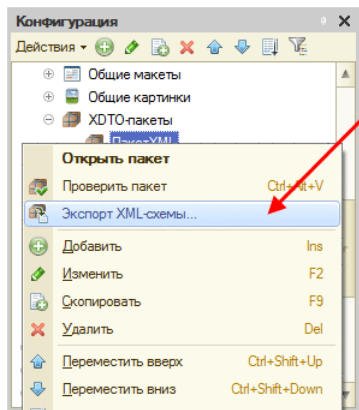


Рис. 103. Экспорт XML-схемы пакета XDTO

После этого будет выполнена проверка выгружаемого пакета XDTO. Если будут обнаружены ошибки, соответствующие сообщения будут выведены в окно сообщений, а процедура экспорта будет прервана.

В случае успешной проверки будет предложено выбрать каталог и имя файла `.xsd`, после чего схема XML будет экспортирована в указанный файл.

5.5.18.4. Проверка пакета XDTO

Для того чтобы проверить пакет XDTO, следует выделить в дереве конфигурации требуемый пакет XDTO и выполнить команду контекстного меню [Проверить пакет](#).

В результате будет выполнена проверка модели пакета XDTO (описание правил проверки см. [здесь](#)).

Если будут обнаружены ошибки, соответствующие сообщения будут выведены в окно сообщений.

5.5.18.5. Окно редактирования пакета XDTO

Редактирование пакета XDTO выполняется в окне редактирования пакета XDTO.

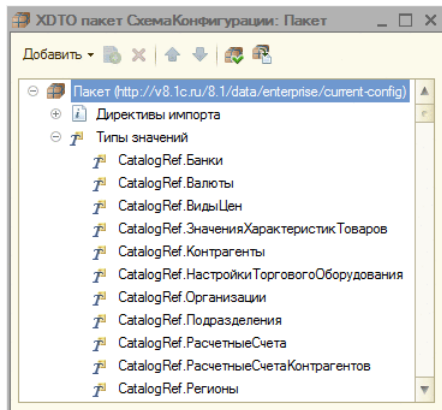


Рис. 104. Окно редактирования пакета XDTO

При добавлении нового пакета XDTO в дерево конфигурации окно редактирования пакета XDTO открывается автоматически.

Для того чтобы открыть окно редактирования для существующего пакета XDTO, следует выделить в дереве конфигурации требуемый пакет XDTO и выполнить команду контекстного меню [Открыть пакет](#).

5.5.18.5.1. Иерархическая структура пакета XDTO

Окно редактирования пакета XDTO содержит иерархическую структуру пакета XDTO, отображенную в виде дерева.

В корне дерева расположен идентификатор пакета XDTO, содержащий URI пространства имен данного пакета.

На первом уровне иерархии могут располагаться следующие элементы пакета:

- **Директивы импорта** – перечень директив импорта. Каждая директива импорта представляет собой ссылку на другой пакет, содержащий типы, на которые так или иначе ссылается данный пакет. При работе с данным пакетом XDTO средствами встроенного языка данный перечень директив импорта будет доступен в виде объекта [КоллекцияПакетовXDTO](#), содержащегося в свойстве [Зависимости](#) пакета XDTO.
- **Типы значений** – перечень типов значений XDTO, которые содержит пакет XDTO.
- **Типы объектов** – перечень типов объектов XDTO, которые содержит пакет XDTO.
- **Свойства** – перечень свойств пакета XDTO. Представляет собой объявления объектов/значений, которые могут являться корневыми элементами документов XML, принадлежащих URI пространству имен данного пакета XDTO.
- Каждый тип значения XDTO описывается иерархической структурой и может содержать в своем составе следующие элементы:
 - **Образец** – описывает один фасет XDTO типа [Образец](#).
 - **Перечисление** – описывает один фасет XDTO типа [Перечисление](#).

Каждый тип объекта XDTO описывается иерархической структурой, которая может содержать в своем составе набор свойств объекта.

5.5.18.5.2. Свойства пакета XDTO

Редактирование свойств пакета XDTO выполняется в палитре свойств.

Если палитра свойств открыта для пакета XDTO, выделенного в дереве конфигурации, то в ней будут содержаться следующие свойства: [Имя](#), [Синоним](#), [Комментарий](#), [Подсистемы](#) и [URI пространства имен](#). Кроме этого, палитра свойств будет содержать ссылку [Пакет](#), по которой можно перейти в окно редактирования пакета XDTO.

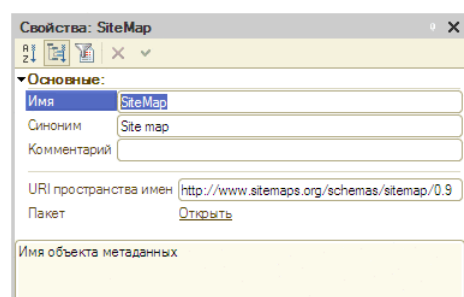


Рис. 105. Свойства пакета XDTO

Если палитра свойств открыта для пакета XDTO, выделенного в окне редактирования пакета XDTO (корневой элемент), то она содержит единственное свойство – [URI пространства имен](#). Это свойство задает URI пространства имен пакета XDTO, к которому принадлежат все определенные в этом пакете типы.

5.5.18.5.3. Свойства директивы импорта

Редактирование свойств директивы импорта выполняется в палитре свойств. Для директивы импорта палитра свойств содержит единственное свойство – [Пространство имен](#). Это свойство задает URI пространства имен импортируемого пакета.

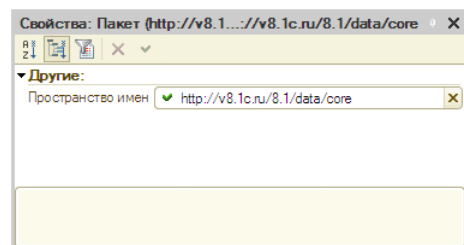


Рис. 106. Свойства директивы импорта

5.5.18.5.4. Свойства типа значения XDTO

Редактирование свойств типа значения XDTO выполняется в палитре свойств.

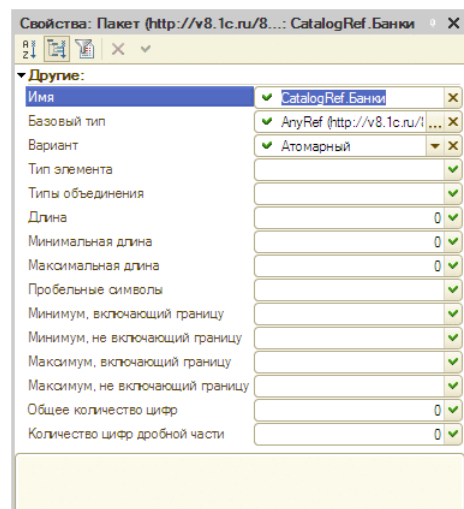


Рис. 107. Свойства значения XDTO

Для типа значения XDTO палитра свойств содержит следующие свойства:

- **Имя** – имя типа значения XDTO.
- **Базовый тип** – базовый тип для данного типа значения XDTO.
- **Вариант** – вариант простого типа (атомарный тип, список, объединение). Если значение установлено, то должно не противоречить значениям **Тип элемента** и **Типы объединения**.
- **Тип элемента** – тип элемента списка в случае, когда тип значения XDTO определяется списком. При этом все фасеты и свойство **Типы подчиненных** должны быть пустыми.
- **Типы объединения** – список типов, образующих объединение в случае, когда тип значения XDTO определяется объединением. Объединяться могут только типы значений XDTO. При этом все фасеты и свойство **Тип элемента** должны быть пустыми.
- **Длина** – фасет длины.
- **Минимальная длина** – фасет минимальной длины.
- **Максимальная длина** – фасет максимальной длины.
- **Пробельные символы** – фасет пробельного символа.
- **Минимум, включающий границу** – фасет минимума, включающего границу.
- **Минимум, не включающий границу** – фасет минимума, не включающего границу.
- **Максимум, включающий границу** – фасет максимума, включающего границу.
- **Максимум, не включающий границу** – фасет максимума, не включающего границу.
- **Общее количество цифр** – фасет общего количества цифр.
- **Количество цифр дробной части** – фасет количества цифр дробной части.

5.5.18.5.5. Свойства типа объекта XDTO

Редактирование свойств типа объекта XDTO выполняется в палитре свойств.

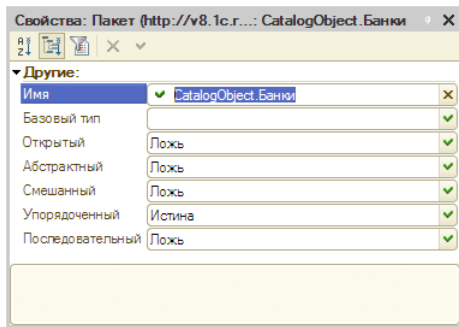


Рис. 108. Свойства объекта XDTO

Для типа объекта XDTO палитра свойств содержит следующие свойства:

- **Имя** – имя типа объекта XDTO.
- **Базовый тип** – базовый тип для данного типа объекта XDTO. Это может быть только тип объекта XDTO.
- **Открытый** – признак, является ли тип объекта XDTO открытым. Данное свойство показывает, может ли экземпляр объекта XDTO содержать дополнительные свойства, не определенные в его типе.
- **Абстрактный** – признак, является ли тип объекта XDTO абстрактным.
- **Смешанный** – свойство показывает, имеет ли соответствующий объект XDTO смешанное содержание. Если значение свойства **Смешанный** равно **Истина**, то значение **Последовательный** обязательно равно **Истина**, так как смешанное содержание невозможно смоделировать без применения последовательности XDTO.
- **Упорядоченный** – признак, является ли порядок следования элементов, представляющих значения свойств, строго соответствующим порядку следования свойств в типе объекта XDTO. Если свойство **Упорядоченный** имеет значение **Ложь**, то на входе порядок следования элементов XML не контролируется, а на выходе определяется порядком следования свойств, если только свойство **Последовательный** не имеет значение **Истина**.
- **Последовательный** – это свойство показывает, содержит ли экземпляр соответствующего объекта XDTO последовательность XDTO. Данный признак равен значению **Истина** в тех случаях, когда порядок следования вложенных элементов XML не может однозначно определяться порядком следования свойств в типе или соответствующий объект XDTO имеет смешанное содержание. Последовательность XDTO позволяет задать в явном виде порядок следования элементов, как они будут представлены в документе XML. Для объектов типов, у которых свойство **Последовательный** установлено в значение **Ложь**, порядок следования вложенных элементов соответствует порядку следования свойств.

5.5.18.5.6. Свойства свойств типа объекта XDTO

Редактирование свойства свойств типа объекта XDTO выполняется в палитре свойств.

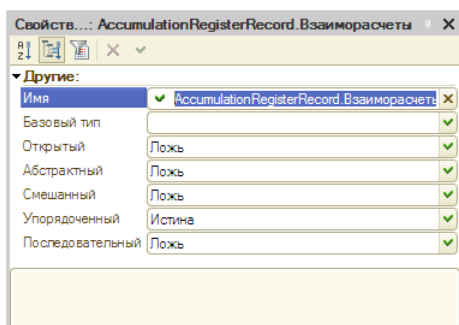


Рис. 109. Палитра свойств типа XDTO

Для типа объекта XDTO палитра свойств содержит следующие свойства:

- **Имя** – имя свойства. В пределах одного типа объекта XDTO имена свойств должны быть уникальными.
- **Тип** – тип свойства. Может быть как типом значения XDTO, так и типом объекта XDTO.
- **Минимальное количество** – минимальное количество значений свойства. Минимальное количество значений свойства может принимать значения ≥ 0 . Естественно, значение **Минимальное количество** должно быть меньше или равно значению **Максимальное количество** (если, конечно, **Максимальное количество** не равно -1);
- **Максимальное количество** – свойство типа объекта XDTO может быть определено как содержащее одно или множество значений. Свойство считается содержащим одно значение, если свойство **Максимальное количество** равно 1. Если же свойство **Максимальное количество** больше 1, то считается, что свойство может содержать множество значений. Такое свойство в структуре объекта моделируется как список. Свойство **Максимальное количество** показывает максимальное количество значений свойства. Максимальное количество > 1 может быть задано только для свойств, представляемых в виде элемента XML.
- **Возможно пустое** – показывает, может ли свойство принимать неопределенное значение. Свойство **Возможно пустое**, равное *Истина*, может быть определено только для свойств с формой представления **Элемент**. Если **Максимальное количество** > 1 , неопределенное значение является допустимым для элемента списка значений свойства.
- **Фиксированное** – указывает, является ли значение свойства фиксированным. Если установлено в значение *Истина*, то само фиксированное значение можно получить через свойство **По умолчанию**.
- **По умолчанию** – значение свойства по умолчанию. Тип значения по умолчанию может быть только типом значения XDTO. При этом данное значение должно быть совместимо с типом свойства (быть того же типа, что и тип свойства или же унаследованного типа). При создании объекта XDTO свойство, если оно допускает единственное значение, принимает значение по умолчанию. Для свойств с множеством значений список значений изначально пуст, независимо от того, определено или нет значение по умолчанию.
- **Форма** – форма представления свойства в XML. Это может быть **Текст**, **Элемент** или **Атрибут**. Если формой представления является **Атрибут** или **Текст**, то значение свойства **Максимальное количество** не может быть больше 1. Если свойство принимает значение **Текст**, то значение свойства **Минимальное количество** также должно быть равным 1. У одного типа только одно свойство может иметь форму представления **Текст**, при этом остальные свойства должны иметь форму представления **Атрибут**.
- **Локальное имя** – локальное имя, используемое для представления свойства. Для свойств с формой представления **Текст** – пустая строка.

5.5.18.5.7. Глобальное свойство

Редактирование свойств глобального свойства выполняется в палитре свойств.

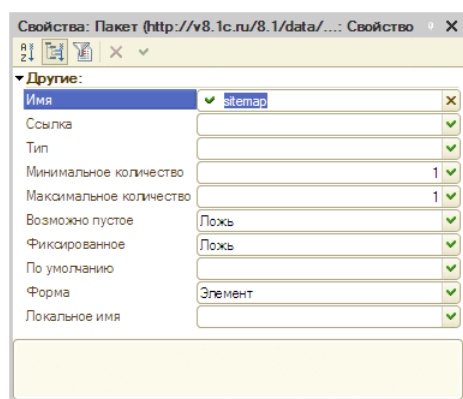


Рис. 110. Палитра свойств глобального свойства

Для глобального свойства палитра свойств содержит следующие свойства:

- **Имя** – имя глобального свойства. В пределах одного типа объекта XDTO имена глобальных свойств должны быть уникальными.
- **Ссылка** – ссылка на корневое определение свойства пакета.
- **Тип** – тип глобального свойства.
- **Минимальное количество** – минимальное количество значений свойства. Если **Минимальное количество** = 0, то значение свойства может быть не установлено.
- **Максимальное количество** – максимальное количество значений свойства. Если **Максимальное количество** равно -1, то количество значений свойства неограниченно.
- **Возможно пустое** – показывает, может ли свойство принимать неопределенное значение.
- **Фиксированное** – указывает, является ли значение свойства фиксированным.
- **По умолчанию** – значение свойства по умолчанию. Лексическое представление значения свойства должно соответствовать правилам проверки типа данного свойства.
- **Форма** – форма представления свойства в XML. Это может быть **Текст**, **Элемент** или **Атрибут**.
- **Локальное имя** – локальное имя, используемое для представления свойства.

5.5.19. Web-сервисы

Механизм Web-сервисов позволяет использовать «1С:Предприятие» как набор сервисов в сложных распределенных и гетерогенных системах, а также позволяет интегрировать «1С:Предприятие» с другими промышленными системами использованием сервисно-ориентированной архитектуры.

Подробнее об использовании механизма Web-сервисов см. [здесь](#).

5.5.19.1. Добавление Web-сервиса

Для того чтобы добавить Web-сервис в дерево конфигурации, следует выделить ветку **Общие – Web-сервисы** и выполнить команду контекстного меню **Добавить**.

В результате выполнения команды будет открыто окно редактирования Web-сервиса (см. [здесь](#)).

На закладке **Прочее** окна редактирования Web-сервиса следует установить следующие параметры:

- **URI пространства имен** – содержит URI пространства имен Web-сервиса. Каждый Web-сервис может быть однозначно идентифицирован по своему имени и URI пространству имен, которому он принадлежит.
- **Пакеты XDTO** – перечень пакетов XDTO, типы которых могут использоваться в качестве типов возвращаемого значения операций и типов параметров операций Web-сервиса.
- **Имя файла публикаций** – имя файла описания Web-сервиса, который расположен на веб-сервере (о публикации Web-сервисов см. книгу «1С:Предприятие 8.3. Руководство администратора»).

Для получения доступа к Web-сервису необходимо использовать адрес, который формируется следующим образом: <Имя хоста веб-сервера>/<Имя виртуального каталога>/ws/<Имя Web-сервиса> или <Имя хоста веб-сервера>/<Имя виртуального каталога>/ws/<Адрес Web-сервиса>.

Так, если виртуальный каталог имеет имя DemoWS, имя Web-сервиса в конфигураторе указано как ДемонстрацияРаботыWS, а в качестве адреса указано DemoWorkWS, то обращение к Web-сервису можно выполнять одновременно по двум адресам (для получения доступа с локальной машины): <http://localhost/DemoWS/ws/ДемонстрацияРаботыWS> или <http://localhost/DemoWS/ws/DemoWorkWS>.

Кроме этого, на закладке содержится кнопка **Модуль**, которая позволяет открыть для редактирования модуль Web-сервиса.

5.5.19.2. Иерархическая структура Web-сервиса

Каждый Web-сервис, описываемый в дереве конфигурации, может содержать набор операций. Каждой операции должна соответствовать экспортируемая процедура, описанная в модуле Web-сервиса.

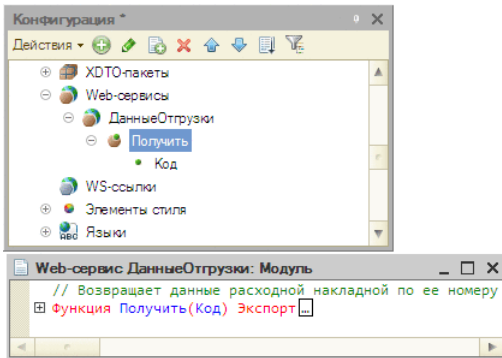


Рис. 111. Описание Web-сервиса

В свою очередь, каждая операция может содержать набор параметров, имена которых должны соответствовать именам параметров процедуры, описывающей данную операцию.

5.5.19.3. Операции Web-сервиса

На закладке **Операции** выполняется добавление операции Web-сервиса. Редактирование свойств операции выполняется в палитре свойств.

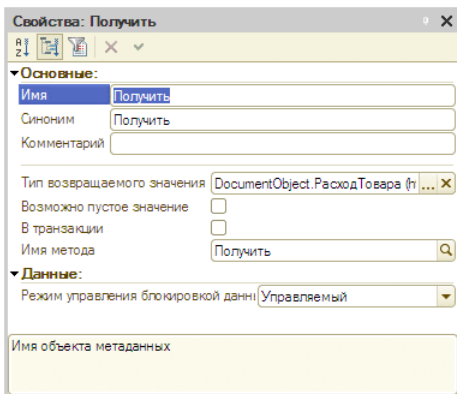


Рис. 112. Свойство операции Web-сервиса

Тип возвращаемого значения – тип значения, которое возвращает операция Web-сервиса. Может являться типом значения XDTO или типом объекта XDTO.

Возможно пустое значение – показывает, может ли возвращаемое значение принимать неопределенное значение.

В транзакции – показывает, будет ли выполняться код модуля Web-сервиса в транзакции или нет. Если свойство установлено, то при вызове Web-сервиса автоматически будет начата транзакция, а при завершении транзакция будет либо зафиксирована, либо произойдет откат транзакции (в зависимости от результатов выполнения). Если свойство не установлено, при начале исполнения модуля Web-сервиса транзакция начата не будет.

Имя процедуры – имя процедуры модуля Web-сервиса, которая будет выполнена при вызове данного свойства.

5.5.19.4. Параметры операции

На закладке **Операции** для указанной операции нужно осуществить задание параметров операции Web-сервиса. Редактирование свойств параметра выполняется в палитре свойств.

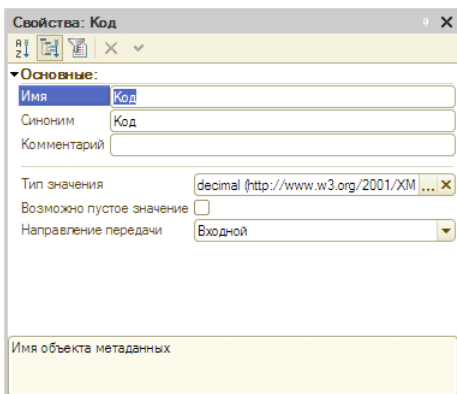


Рис. 113. Свойства параметра операции

Тип значения – тип значения параметра операции Web-сервиса. Может являться типом значения XDTO или типом объекта XDTO.

Возможно пустое значение – показывает, может ли значение параметра операции принимать неопределенное значение.

Направление передачи – определяет направление передачи данных с помощью данного параметра. Возможные значения:

- **Входной** – означает, что параметр может использоваться только для передачи данных Web-сервису.
- **Выходной** – означает, что параметр может использоваться как для передачи данных, так и для их получения от Web-сервиса.

5.5.19.5. Указание типов, определяемых системой

Чтобы в Web-сервисе воспользоваться типами, определяемыми системой «1С:Предприятие» (например, в параметрах и возвращаемом значении операций), нужно в конфигурации определить пакеты XDTO и для каждого пакета указать в его списке импортируемых пакетов (свойство **Директивы импорта**) набор пакетов платформы, в которые эти типы входят. URI пространства имен для указания типа содержится в статье синтакс-помощника по объекту данного типа.

5.5.19.6. Публикация Web-сервисов

Публикация Web-сервисов описывается в книге «1С:Предприятие 8.3. Руководство администратора».

5.5.20. HTTP-сервисы

Механизм HTTP-сервисов позволяет использовать «1С:Предприятие» как набор сервисов в сложных распределенных и гетерогенных системах, а также позволяет интегрировать «1С:Предприятие» с другими промышленными системами.

Подробнее об использовании механизма HTTP-сервисов см. [здесь](#).

Описание редактирования HTTP-сервисов см. [здесь](#).

5.5.21. WS-ссылки

Система «1С:Предприятие» может использовать веб-сервисы, предоставляемые другими поставщиками, с помощью статических ссылок, создаваемых в дереве конфигурации.

5.5.21.1. Добавление WS-ссылки

Для того чтобы добавить статическую ссылку на внешний веб-сервис в дерево конфигурации, следует выделить ветку WS-ссылки и выполнить команду контекстного меню **Добавить** или соответствующую команду **Действия**.

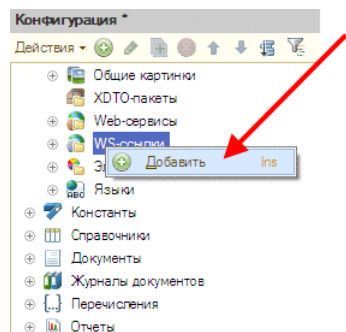


Рис. 114. Добавление WS-ссылки

В открывшемся окне следует ввести URL описания добавляемого веб-сервиса, например: <http://users.v8.1c.ru/ws/products.1cws?wsdl>.

ПРИМЕЧАНИЕ. При добавлении WS-ссылки следует помнить, что «1С:Предприятие» удаляет из введенного URL завершающий символ «/», в силу чего URL <http://localhost/ws/ws-service/?wsdl> и URL <http://localhost/ws/ws-service?wsdl> с точки зрения «1С:Предприятия» будут считаться идентичными.

5.5.21.2. Иерархическая структура WS-ссылки

Просмотр иерархической структуры WS-ссылки выполняется в окне просмотра WS-ссылки. Значения свойств элементов ссылки можно просмотреть в палитре свойств.

Для того чтобы открыть окно просмотра WS-ссылки, следует выделить в дереве конфигурации требуемую WS-ссылку и выполнить команду контекстного меню **Свойства**. После этого в палитре свойств воспользоваться ссылкой WS-ссылка, которая открывает окно просмотра WS-ссылки.

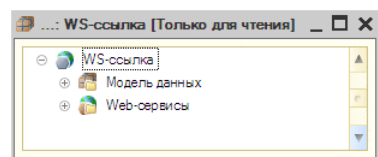


Рис. 115. WS-ссылка

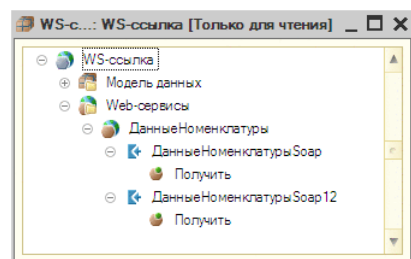
Окно просмотра содержит иерархическую структуру WS-ссылки, отображенную в виде дерева.

На первом уровне иерархии могут располагаться:

- **Модель данных** – содержит перечень пакетов XDTO, описывающих структуру типов, используемую веб-сервисами, на которые ссылается данная WS-ссылка.
- **Web-сервисы** – перечень Web-сервисов, на которые ссылается данная ссылка.

Просмотр структуры и свойств модели данных выполняется аналогично работе с пакетами XDTO (см. [здесь](#)), за исключением того, что редактирование свойств пакетов, отображаемых в окне просмотра WS-ссылки, невозможно.

Просмотр структуры WS-ссылки выполняется аналогично работе с Web-сервисами, за исключением того, что для каждого Web-сервиса отображаются поддерживаемые точки подключения Web-сервиса, для которых, в свою очередь, отображается список операций и параметров каждой операции.



Различные точки подключения Web-сервиса позволяют выполнять операции, используя различные протоколы.

5.5.22. Элементы стиля

Объекты конфигурации **Элементы стиля** предназначены для единообразного оформления различных элементов формы в тех случаях, когда недостаточно того оформления, которое автоматически предлагает «1С:Предприятие». Например, требуется цвет каких-либо надписей в формах конфигурации сделать одинаковым. В этом случае логично создать элемент стиля, задать ему цвет и использовать созданный элемент для установки цвета текста элемента формы.

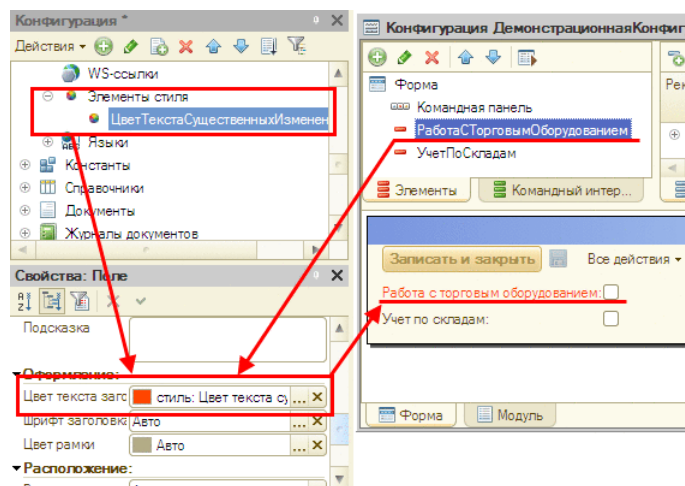


Рис. 117. Использование элемента стиля

Элементы **стиля** могут быть трех разных видов:

- Цвет,
- Шрифт,
- Рамка.

ПРИМЕЧАНИЕ. Системные элементы стиля недоступны для выбора в качестве значений пользовательских элементов стиля.

В диалоге выбора шрифта в тонком клиенте в списке шрифтов присутствуют шрифты, установленные на текущем компьютере, и перечень специальных шрифтов, заключенных в угловые скобки (<>). Если выбран шрифт <Шрифт текста>, то будет использоваться шрифт интерфейса «1С:Предприятия», остальные шрифты соответствуют соответствующим шрифтам операционной системы. В начале списка размеров шрифта <Шрифт текста> присутствует элемент <>. Выбор этого размера шрифта (его значение равно 0) означает, что будет использован размер шрифта интерфейса «1С:Предприятия». Начертание шрифта (жирный, наклонный и т. д.) по умолчанию берется из стиля, но может быть изменено пользователем без каких-либо ограничений. При выборе другого шрифта внесенные пользователем изменения размера или начертания не сохраняются и устанавливаются в значения по умолчанию.

Имеется возможность программного доступа к значению элемента стиля с помощью свойства **Значение**.

Пример:

Метаданные: ЭлементыСтиля.ЦветОтрицательного.Значение

[Копировать в буфер обмена](#)

5.5.23. Языки

Объекты конфигурации **Языки** предназначены для создания интерфейса программы на различных языках. Для каждого объекта конфигурации типа **Языки** резервируется отдельная строка для тех реквизитов метаданных, для которых допускается представление на разных языках.

Рассмотрим пример формы для элемента справочника **Номенклатура**. Для языка **Русский** надписи формы выглядят так:

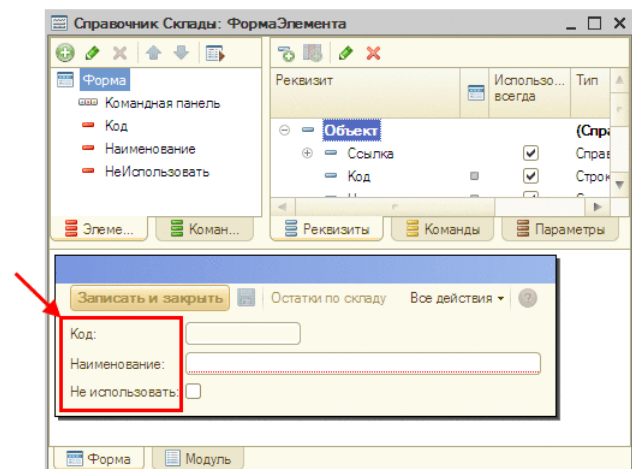


Рис. 118. Форма на русском языке

Если в ветви **Языки** создано несколько объектов (например, **Русский** и **Английский**), то для смены языка нужно выбрать пункт **Конфигурация – Язык редактирования конфигурации**. В открывшемся окне выбора языка выбрать язык просмотра конфигурации.

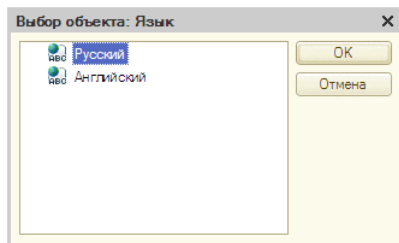


Рис. 119. Выбор языка редактирования конфигурации

Аналогичный результат можно получить с помощью кнопки выбора языка, расположенной в панели состояния справа от кнопок CAP и NUM (правый нижний угол основного окна конфигуратора).

Конфигуратор заменит текст надписей на варианты надписей выбранного языка.

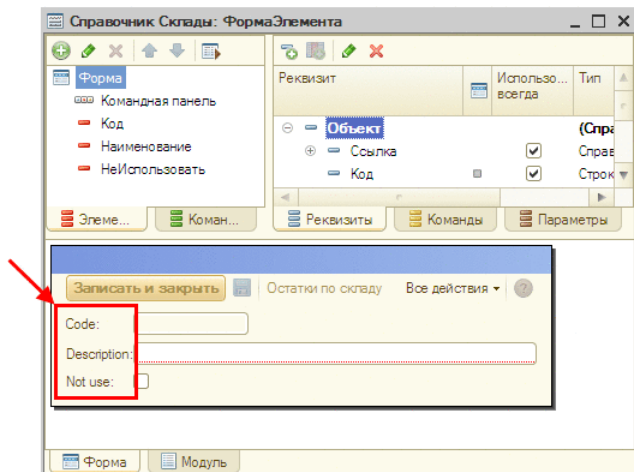


Рис. 120. Форма на английском языке

Разумеется, текст надписей должен быть заранее введен для каждого элемента управления. Для ввода текста в палитре свойств элемента управления **Надпись** в свойстве **Заголовок** (**Текст** или **Синоним**, в зависимости от типа элемента управления) нужно нажать кнопку «лупа» (см. [рис. 121](#)). На экран выводится окно **Строки на разных языках**.

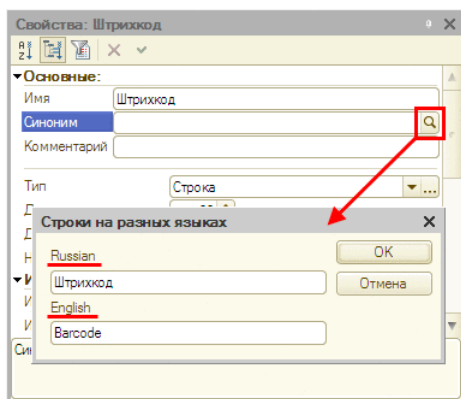


Рис. 121. Строки на разных языках

СОВЕТ. В конфигурации нет ограничений на число языков. Однако не следует создавать объекты типа **Языки** «с запасом», т. е. создать объект **Язык** можно в любой момент.

В свойстве **Код языка** указывается код языка, например, EN для языка **Английский**.

Если в конфигурации определено два и более объекта типа **Языки**, то для свойства **Синоним** и **Заголовок** элемента управления появляется кнопка редактирования текста на разных языках (в виде лупы).

Первый объект языка программа создает в соответствии с выбором языка (страны) при создании новой информационной базы.

Чтобы сформировать текстовое представление реквизита для отображения в форме, используется следующее правило:

- Выполняется получение заголовка отображаемого объекта на **языке конфигурации** текущего пользователя. Если заголовок задан, используется именно он.
- Выполняется попытка получения синонима отображаемого объекта на **языке конфигурации** текущего пользователя. Если синоним задан, используется именно он.
- Дальнейшее зависит от вида отображаемого объекта:
 - для стандартных реквизитов получается представление на языке **локализации платформы**;
 - для объектов, созданных прикладным разработчиком, используется имя объекта так (и на **том языке**), как оно задано в конфигурации.

ВНИМАНИЕ! Если тексты в свойствах **Синоним** или **Заголовок** введены, то изменение кода языка в свойстве **Код языка** приведет к «потере» введенных текстов (тексты остаются для прежнего значения кода). Тексты «восстанавливаются» при указании прежнего значения кода языка.

Для редактирования текстов и выполнения задач локализации (создания интерфейса на другом языке) следует использовать режим **Редактирование текстов интерфейса** (см. [здесь](#)).

В том случае, если предполагается использовать прикладное решение на мобильных устройствах, рекомендуется указывать коды языков в соответствии со стандартом ISO 639 (http://www.iso.org/iso/ru/language_codes). В противном случае при проверке конфигурации будет выдаваться предупреждение.

5.6. Общие свойства объектов конфигурации

5.6.1. Основные свойства

Практически все объекты конфигурации имеют следующие свойства, располагающиеся в категории свойств **Основные**:

- **Имя** – имя объекта конфигурации. Имя должно состоять из одного слова, начинаться с буквы и не содержать специальных символов, кроме «_». По имени объекта производится доступ и управление объектом конфигурации средствами встроенного языка. Имена объектов конфигурации не должны совпадать с зарезервированными словами языка запросов (список зарезервированных слов см. [здесь](#)).
- **Синоним** – помимо имени можно указать также его синоним. Если конфигурация создается для использования на разных языках, то следует указать синонимы на используемых языках. При работе с системой «1С:Предприятие» синоним будет выдаваться в различных списках выбора, заголовках окон, текстах надписей, при формировании интерфейсов с учетом текущего языка. Синоним не имеет ограничений на использование символов. Если синоним не задан, выбирается имя.

ПРИМЕЧАНИЕ. Имя или синоним, выдаваемые пользователю, также называются **представлением объекта конфигурации**.

- **Комментарий** – произвольная строка символов. Как правило, расшифровывает и поясняет имя объекта.

5.6.2. Представление объектов метаданных

Для многих объектов метаданных реализована возможность задавать различные представления объекта. Это сделано для того, чтобы разработчик имел возможность задать представление стандартных команд, их подсказок, а также заголовков форм.

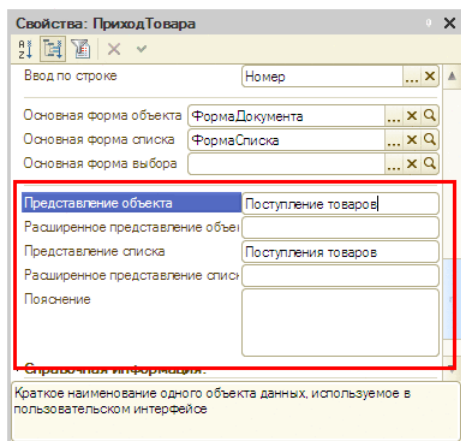


Рис. 122. Представление объектов

- **Представление объекта** (для регистра – записи):
 - название одного объекта (например, [Расчетный счет](#));
 - используется в представлении стандартной команды (создание объекта).
- **Расширенное представление объекта** (для регистра – записи) используется для формирования заголовка формы объекта. Например, [Расчетный счет организации](#).
- **Представление списка**:
 - название списка объектов (например, [Расчетные счета](#));
 - используется в представлении стандартной команды (команда открытия списка объектов).
- **Расширенное представление списка** – используется для формирования заголовка формы списка (например, [Расчетные счета организации](#)).
- **Расширенное представление** – заголовок формы отчета или обработки (например, [Отчет о взаиморасчетах по расчетным счетам](#)).
- **Пояснение** – используется для формирования подсказки к стандартным командам (например, [Расчетные счета наших организаций](#)).
- **Картинка** – картинка для представления подсистемы в панели разделов.

Следует учитывать, что заполнение свойств, связанных с представлением объектов и списков, необходимо только в тех случаях, когда требуется несколько уточнить информацию, отображаемую для объекта по умолчанию.

Например, существует справочник **Товары** (Имя объекта метаданных – **Товары**, Синоним объекта метаданных – **Товары**), элементами которого могут быть товары и услуги. Но в текстах команд вы хотите использовать единственное число для команды (создать товар) и не хотите отражать информацию об услугах, т. к. она «удлиняет» текст команды. В то же время в форме объекта вы хотите указать пользователю, что в данной форме можно редактировать и товары, и услуги. В этом случае имеет смысл свойство **Представление объекта** заполнить текстом **Товар**, а свойство **Расширенное представление** объекта заполнить текстом **Товар (услуга)**. Тогда команда создания элемента справочника товары будет выглядеть как **Товар: создать**, а заголовок формы – **Товар (услуга)**.

Подробную информацию о том, какие из вышеперечисленных свойств каким объектам метаданных соответствуют см. [здесь](#). Там же можно получить информацию о правилах формирования текстов стандартных команд, подсказок команд и заголовков форм.

5.6.3. Произвольное представление данных

Стандартное представление данных, которое формирует система, может не устраивать пользователя. Например, если с системой работают пользователи, говорящие на разных языках и объекты системы содержат всю необходимую информацию также на разных языках. Например, для товаров существует два названия: на русском и на английском языке. А выводить нужный текст необходимо в зависимости от кода локализации сеанса. Также нужно, чтобы представление на нужном языке генерировалось везде, где формируется представление объекта: динамический список, отчет и т. д.

Для реализации этой задачи существует специальный механизм, позволяющий разработчику определить реквизиты, которые участвуют в формировании представления, а также описать алгоритм, по которому формируется представление объекта.

Данная возможность доступна для следующих объектов конфигурации:

- Планы обмена,
- Справочники,
- Документы,
- Планы видов характеристик,
- Планы счетов,

- Планы видов расчета,
- Бизнес-процессы,
- Задачи,
- Таблицы внешних источников данных.

Механизм формирования представления состоит из двух этапов: определение реквизитов, участвующих в формировании представления и собственно формирование представления.

Для определения списка реквизитов используется обработчик события `ОбработкаПолученияПолейПредставления` менеджера соответствующего объекта. Обработчик вызывается при первой попытке получения представления выбранного объекта, и результат работы сохраняется на время сеанса. В данном обработчике необходимо заполнить массив `Поля` именами реквизитов, которые будут участвовать в формировании представления. Если параметр `СтандартнаяОбработка` в обработчике будет установлено в значение `Истина`, то после завершения работы обработчика, массив `Поля` будет очищен и заполнен полями, используемыми для формирования стандартного представления данного объекта. В противном случае система будет использовать только те значения, которые были помещены в массив `Поля`.

ПРИМЕЧАНИЕ. Примеры, приведенные ниже, не являются законченным. Они предназначены для демонстрации механизмов работы с произвольным представлением.

Допустим, что в справочнике `Товары` есть реквизиты `РусскоеНаименование`, `АнглийскоеНаименование` и `Артикул`, которые должны участвовать в формировании представления. В этом случае обработчик получения полей представления (в модуле менеджера справочника `Товары`) будет выглядеть следующим образом:

[Копировать в буфер обмена](#)

Процедура `ОбработкаПолученияПолейПредставления(Поля, СтандартнаяОбработка)`

```
Поля.Добавить("РусскоеНаименование");
Поля.Добавить("АнглийскоеНаименование");
Поля.Добавить("Артикул");
СтандартнаяОбработка = Ложь;
КонецПроцедуры
```

При необходимости получения представления система вызывает обработчик события `ОбработкаПолученияПредставления` в менеджере соответствующего объекта. В этот обработчик передаются значения реквизитов, участвующих в формировании представления. Программный код, расположенный в обработчике, выполняет формирование текста представления.

ВНИМАНИЕ! Обработчик события `ОбработкаПолученияПредставления` вызывается при любой необходимости получения представления какого-либо объекта информационной базы. Избыточные данные, или неправильный выбор данных для формирования представления, может привести к существенному замедлению работы системы.

Обработчик `ОбработкаПолученияПредставления` формирует представление и возвращает его с помощью параметра `Представление` обработчика. Данные, необходимые для формирования представления, передаются с помощью параметра `Данные`. Данные представляют из себя структуру, где ключом является имя реквизита, а значением – значение реквизита для текущего объекта. Если параметр `СтандартнаяОбработка` установить в значение `Истина`, то система попытается сформировать стандартное представление для текущего объекта, исходя из переданных данных. Если в параметре `Данные` отсутствуют реквизиты, необходимые для формирования стандартного представления, то представлением будет пустая строка.

Рассмотрим пример формирования представления для двуязычного справочника `Товары`:

[Копировать в буфер обмена](#)

Процедура `ОбработкаПолученияПредставления(Данные, Представление, СтандартнаяОбработка)`

```
СтандартнаяОбработка = Ложь;
КодЛокализацииСеанса = ВРег(ТекущийКодЛокализации());
Если Найти(КодЛокализацииСеанса, "RU") <> 0 Тогда
    Текст = Данные.РусскоеНаименование;
ИначеЕсли Найти(КодЛокализацииСеанса, "EN") <> 0 Тогда
    Текст = Данные.АнглийскоеНаименование;
Иначе
    Текст = Данные.РусскоеНаименование;
КонецЕсли;
Представление = Текст + "(" + Данные.Артикул + ")";
КонецПроцедуры
```

Следует помнить, что если в обработчике `ОбработкаПолученияПредставления` будет совершена ошибка встроенного языка, то возможна ситуация, когда работа приложения будет аварийно завершена.

5.6.4. Стандартные реквизиты

Для того чтобы на уровне конфигурации переопределять некоторые интерфейсные свойства (такие как синоним, проверка заполнения и т. д.) стандартных реквизитов (например, `Код`, `Наименование`, `Родитель`) и стандартных табличных частей (например, `ВидыСубконто`, `БазовыеВидыРасчета`) прикладных объектов, существует возможность настройки этих свойств.

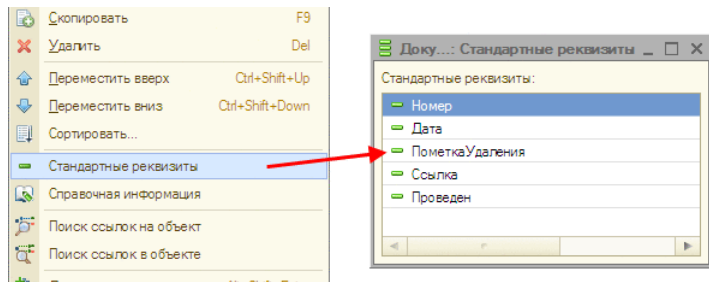


Рис. 123. Стандартные реквизиты

Для этого в палитре свойств ряда объектов существуют команды, открывающие списки стандартных реквизитов и стандартных табличных частей (см. [рис. 123](#)). Такие команды доступны для тех объектов, у которых имеются стандартные реквизиты и стандартные табличные части.

С помощью палитры свойств можно переопределять некоторые свойства стандартных реквизитов таким образом, чтобы они более полно соответствовали требованиям решаемой прикладной задачи. Например, для свойства `Владелец` справочника `РасчетныеСчета` можно задать синоним `Контрагент`. Тогда во всех формах представление этого реквизита (`Владелец`) по умолчанию будет выглядеть как «**Контрагент**».

В том случае если свойства стандартного реквизита (или стандартной табличной части) не заданы, будут использованы свойства стандартных реквизитов по умолчанию.

По своему набору свойств стандартный реквизит практически ничем не отличается от любого другого реквизита, за исключением того, что нельзя изменить:

- имя стандартного реквизита,
- тип стандартного реквизита,
- имя стандартной табличной части,
- индексирование стандартного реквизита.

ВНИМАНИЕ! Наличие описания стандартного реквизита не меняет его имени, то есть все обращения к реквизиту во встроенном языке и в языке запросов остаются неизменными.

5.6.5. Предопределенные данные

5.6.5.1. Общая информация

Предопределенные данные – это элементы прикладных объектов, которые создаются в конфигураторе и обращение к ним возможно по имени, без предварительного поиска. Предопределенные данные имеет смысл создавать в том случае, если какой-либо элемент данных будет нужен постоянно и требуется упростить обращение к этому элементу из программного кода. Например, в справочнике **Товары** можно завести предопределенный элемент **Услуга**, обращение к которому будет выглядеть следующим образом: **Справочники.Товары.Услуга**. Предопределенными могут быть как элементы, так и группы элементов. Группы можно создавать в тех случаях, когда предопределенные элементы создаются в иерархических объектах, например в иерархическом справочнике. Предопределенные данные могут быть созданы:

- для справочников (см. [здесь](#)),
- планов счетов (см. [здесь](#)),
- планов видов характеристик (см. [здесь](#)),
- планов видов расчета (см. [здесь](#)).

После создания предопределенного элемента в конфигураторе, возможность создания или обновления связанного с ним элемента данных определяется несколькими параметрами:

- Значение свойства объекта в информационной базе. Получить и установить значение этого свойства можно в режиме «1С:Предприятие» с помощью методов `ПолучитьОбновлениеПредопределенныхДанных()` и `УстановитьОбновлениеПредопределенныхДанных()`;
- Значение свойства объекта конфигурации `ОбновлениеПредопределенныхДанных`. Устанавливается в конфигураторе;
- Значение, установленное для всей информационной базы. Получить и установить значение этого свойства можно в режиме «1С:Предприятие» с помощью методов `ПолучитьОбновлениеПредопределенныхДанныхИнформационнойБазы()` и `УстановитьОбновлениеПредопределенныхДанныхИнформационнойБазы()`;
- Вид узла распределенной информационной базы (см. [здесь](#)). Значение этого параметра определяется автоматически:
 - Для корневого узла и не распределенной информационной базы принимает значение **Обновлять автоматически**;
 - Для подчиненного узла распределенной информационной базы – **Не обновлять автоматически**.

Каждый вышеописанный параметр может принимать одно из следующих значений (кроме параметра, описывающего вид узла распределенной информационной базы):

- **Не обновлять автоматически** – в этом случае система не выполняет создание или обновление элемента данных при создании или изменении предопределенных данных. Следует понимать, что если для объекта конфигурации установлено такое свойство, то при обращении к предопределенным данным может генерироваться исключение, т. к. отсутствует элемент данных, связанный с предопределенным элементом;
- **Обновлять автоматически** – в этом случае система автоматически создаст (или обновит существующие) элементы данных для новых (или измененных) предопределенных данных;
- **Авто** – система автоматически определяет необходимость выполнения обновления.

Выполняемое действие (обновлять или не обновлять) определяется последовательным анализом вышеперечисленных параметров (в указанном порядке) до тех пор, пока не будет обнаружено значение, отличное от значения **Авто**. Это значение и будет определять возможность создания или обновления предопределенных элементов.

Связь элемента данных и элемента предопределенных данных осуществляется с помощью свойства `ИмяПредопределенныхДанных`. С помощью этого свойства имеется возможность выполнять следующие действия:

- Связывать элемент предопределенных данных с элементами данных. Для этого следует присвоить свойству `ИмяПредопределенныхДанных` имя предопределенного элемента данных, который надо связать с данными:

[Копировать в буфер обмена](#)

```
ЭлементСсылка = Справочник.Товары.НайтиПоНаименованию("Услуга доставки ");
Объект = ЭлементСсылка.ПолучитьОбъект();
Объект.ИмяПредопределенныхДанных = "Услуга";
Объект.Записать();
```

В результате этой операции, предопределенному элементу **Услуга** справочника **Товары** будет сопоставлен элемент данных с именем **Услуга доставки**.

При записи объекта проверяется, что в информационной базе не существует другого элемента данных с именем предопределенных данных, установленного в значение, с которым записывается элемент данных.

- «Отключить» элемент данных от элемента предопределенных данных. Для этого необходимо присвоить свойству `ИмяПредопределенныхДанных` пустую строку и записать элемент:

[Копировать в буфер обмена](#)

```
Объект = Справочник.Товары.Услуга.ПолучитьОбъект();
Объект.ИмяПредопределенныхДанных = "";
Объект.Записать();
```

Если попробовать обратиться к предопределенному элементу **Справочник.Товары.Услуга** после выполнения вышеприведенного примера, то будет сгенерировано исключение.

Таким образом, процесс смены элемента данных, связанного с предопределенными данными, выполняется в два этапа:

- Выполняется «отключение» существующего элемента данных от предопределенного элемента;
- Выполняется связывание нового элемента данных с предопределенным элементом.

Создание элементов данных, связанных с предопределенными данными, может быть выполнено одним из трех способов:

- Автоматически, во время реструктуризации информационной базы. Это происходит в следующем случае:
 - Разрешено автоматическое создание и обновление предопределенных данных;
 - Предопределенные данные ранее создавались в данной области данных или информационной базе.
- Автоматически, при первом обращении к таблице, хранящей данных объекта конфигурации. Создание предопределенных элементов в этом случае будет выполняться при условии, что создание предопределенных элементов не запрещено.
- С помощью встроенного языка, указав свойство `ИмяПредопределенныхДанных` при создании элемента данных. Этот способ можно использовать, если автоматическое создание предопределенных элементов запрещено в свойствах объектов конфигурации.

Обновление данных, связанных с предопределенными, будет выполняться в том случае, если разрешено автоматическое обновление предопределенных данных, предопределенные данные связаны с реальными данными и предопределенные данные изменены в конфигураторе. Во всех остальных случаях информация, введенная в конфигураторе, не будет перенесена в информационную базу.

Изменение порядка следования предопределенных элементов в предопределенных табличных частях плана счетов и плана видов расчета обрабатывается в связанных данных, т. е. порядок элементов в данных будет таким же, каким он задан в метаданных. Данные, введенные пользователем (не предопределенные) будут располагаться после предопределенных.

Если в конфигураторе выполнено удаление элемента предопределенных данных, то будет выполнено следующее:

- Элемент данных, связанный с элементом предопределенных данных, будет помечен на удаление.
- В списке субконто элемента плана счетов будут удалены соответствующие записи.
- В списках ведущих, базовых и вытесняющих видов расчета будут удалены соответствующие записи.

Для принудительного создания элементов предопределенных данных (для справочников, планов видов характеристик, планов видов расчетов и планов счетов) следует воспользоваться методом [ИнициализироватьПредопределенныеДанные\(\)](#). Вызов этого метода выполнит принудительное создание элементов предопределенных данных при выполнении следующих условий:

- Предопределенные элементы ранее не создавались в текущей области данных.
- Обновление предопределенных данных настроено таким образом, что создание и обновление предопределенных данных должно выполняться.

Если указанные условия не выполняются – создание предопределенных элементов не выполняется.

Если в прикладном решении отключено автоматическое обновление предопределенных данных, то удаление предопределенного элемента приводит к изменению в связанном объекте данных (если таковой есть). Значение свойства [ИмяПредопределенныхДанных](#) заменяется значением [#xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx](#), где [xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx](#) – это уникальный идентификатор удаленного элемента метаданных.

Если требуется получить список предопределенных элементов для объекта конфигурации, то сделать это можно с помощью метода [ПолучитьИменаПредопределенных\(\)](#).

Пример получения списка предопределенных:

```
Список = Метаданные.Справочники.Товары.ПолучитьИменаПредопределенных();
```

[Копировать в буфер обмена](#)

Если метод вернул пустой массив, значит, у объекта конфигурации отсутствуют предопределенные значения. В противном случае можно, например, для каждого предопределенного элемента выполнить проверку наличия связанного элемента в данных.

5.6.5.2. Работа в распределенной информационной базе

При работе в режиме распределенной информационной базы вместе с конфигурацией передаются предопределенные элементы. Собственно элементы данных, связанные с предопределенными данными, передаются обычным образом, с помощью плана обмена. При этом автоматическое создание предопределенных данных в периферийной информационной базе не произойдет, подробное описание причины см. [здесь](#).

ВНИМАНИЕ! Если данные, связанные с предопределенными элементами, оказались получены в подчиненном узле раньше, чем конфигурация со связанными предопределенными элементами, то автоматического «связывания» данных и предопределенных данных не произойдет. Для этого потребуется повторно выгрузить из центральной в периферийную информационную базу нужные элементы данных.

Если начальный образ периферийной информационной базы создается до того, как с главной информационной базой начали работать, можно воспользоваться специальным методом [ИнициализацияПредопределенныхДанных\(\)](#) для создания предопределенных элементов, чтобы они автоматически попали в периферийную информационную базу. Метод надо вызывать до того, как выполнено создание начального образа.

При отмене назначения главного узла (например, для выполнения каких-либо специальных действий), следует учитывать следующий момент: при определении необходимости создания или обновления предопределенных элементов такая информационной базы не будет считаться периферийной. В связи с этим, при первом запуске такой информационной базы в режиме «1С:Предприятие», режим создания или обновления предопределенных элементов может быть определен как [Обновлять автоматически](#). Чтобы избежать этого, можно воспользоваться параметром командной строки [/SetPredefinedDataUpdate](#) (или использовать метод глобального контекста [УстановитьОбновлениеПредопределенныхДанныхИнформационнойБазы\(\)](#)).

Этим же способом можно воспользоваться в том случае, если используется универсальный механизм обмена данными между информационными базами с подобными, но не идентичными типовыми решениями, и необходимо сделать одну информационную базу ведущей (в части предопределенных данных). Для информационной базы, где необходимо создавать предопределенные элементы, следует указать значение [Создавать автоматически](#), а для другой информационной базы – [Не создавать автоматически](#). При этом связь элемента конфигурации и объекта данных будет выполняться по имени предопределенного элемента.

Также при использовании универсального механизма обмена данными следует учитывать следующие особенности:

- При записи объекта данных, который в базе-источнике является предопределенным, система проверяет наличие такого же имени предопределенных данных в типобразующем объекте конфигурации. Если такое имя есть – объект будет загружен без изменений. Если такого имени нет – в объекте при загрузке будет очищено свойство [ИмяПредопределенныхДанных](#).

Например, выполняется универсальный обмен данными между одинаковыми информационными базами. В каждой информационной базе существует справочник [Номенклатура](#). В каждой информационной базе в этом справочнике содержится предопределенный элемент [Услуга](#). Ссылки на эти объекты данных разные.

Тогда при загрузке произойдет следующее: система обнаружит наличие в базе-приемнике предопределенный элемент [Услуга](#) и выполнит загрузку с сохранением свойства [ИмяПредопределенногоЭлемента](#). Это будет возможно в том случае, если перед записью элемента свойство [ОбменДанными.Загрузка](#) установлено в значение [Истина](#). В противном случае будет вызвано исключение.

Однако при попытке обратиться в базе-приемнике к предопределенному элементу [Услуга](#), произойдет ошибка, т. к. в системе не может быть двух объектов, связанных с одинаковым предопределенным элементом.

Не следует допускать возникновения такой ситуации. Прикладной разработчик должен самостоятельно обрабатывать попытку загрузки объекта данных с дублирующим именем предопределенных данных.

- При записи объекта данных, у которого имя предопределенных данных ссылается на удаленный элемент (имеет вид [#xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx](#)), то такой элемент будет записан в любом случае. Однако система не позволит записать два объекта, ссылающиеся на один удаленный предопределенный элемент (с одинаковым значением имени предопределенных данных), кроме как в случае, когда перед записью элемента свойство [ОбменДанными.Загрузка](#) установлено в значение [Истина](#).

При необходимости отключать информационной базы от главного узла, при использовании подчиненных данных, рекомендуется следовать рекомендациям, описание которых см. [здесь](#).

5.6.5.3. Работа в режиме разделения

Описание особенностей работы в разделенной информационной базе см. [здесь](#).

5.6.6. Нумерация объектов

5.6.6.1. Общая информация

В системе «1С:Предприятие» существуют объекты, которые необходимо нумеровать в автоматическом или ручном режиме. Нумерация доступна для следующих объектов:

- Справочники;
- Документы;
- Планы видов характеристик;

- Бизнес-процессы;

- Задачи;

- Планы обмена.

В зависимости от объекта, доступны различные виды и режимы нумерации. Нумерация может выполняться для номера (документы, бизнес-процессы, задачи) и кода (все остальные объекты). Рассмотрим особенности нумерации объектов более подробно.

Тип кода/номера

Код/номер объекта конфигурации может быть числовым или строковым.

Если код/номер задается строковым, то появляется несколько различных дополнительных возможностей:

- Код/номер может состоять не только из цифр, а из любых доступных символов.
- **Допустимая длина кода** – определяет, какую длину будет иметь код/номер. Если это свойство имеет значение **Фиксированная**, то длина кода/номера всегда будет равна значению, указанному в качестве свойства **Длина кода** или **Длина номера**. При этом строка будет дополняться пробелами. Если для свойства выбрано значение **Переменная**, то длина кода/номера не может быть больше значения, указанного в свойствах **Длина кода** или **Длина номера**, но фактическая длина будет определяться количеством символов, участвующих в формировании номера.
- Для кода/номера имеется возможность задавать различные префиксы (см. [здесь](#)).

При использовании строкового кода/номера необходимо понимать, как работает нумерация. Рассмотрим пример нумерации. Тип кода справочника – **Строка**. Длина кода – 3 символа. Допустим, оператор ввел номер **01** и записал элемент справочника. В результате этого действия максимальное значение номера, который может быть создан системой автоматической нумерации, будет равен **99**. Это произошло потому, что при ручном вводе нового номера произошло явное ограничение длины номера двумя разрядами. Система автоматической нумерации «не умеет» автоматически расширять создаваемый номер, если длина номера явно ограничена.

Таким образом, для того, чтобы автоматическая нумерация строковых кодов/номеров работала хорошо, рекомендуется или не указывать номера вручную или задавать лидирующие нули так, чтобы любой код/номер всегда был максимально возможной длины (с учетом ведущих нулей).

Также не стоит забывать, что указание ведущих нулей обеспечит корректную сортировку кодов/номеров в случае, если тип кода/номера является строкой.

Признак автонумерации

Данное свойство определяет, будет система автоматически назначать код/номер для нового объекта при записи объекта или нет. В случае включенной автоматической нумерации система обеспечивает следующее поведение:

- Выдается код/номер, не совпадающий ни с одним номером в информационной базе;
- Выдаваемый код/номер должен быть на 1 больше максимального из значения кода/номера среди объектов информационной базы (по которым обеспечивается нумерация) и кодов/номеров, которые выданы для объектов, еще не записанных в информационную базу.
- При удалении или ручном уменьшении максимального кода/номера, нумерация (при возможности) продолжается с максимального оставшегося номера/кода. Продолжение нумерации с минимально возможного кода/номера в общем случае не гарантируется.

Контроль уникальности

Данное свойство определяет, будет система контролировать уникальность кода/номера для объекта или нет. Контроль будет выполняться как для автоматически назначенного кода/номера, так и для кода/номера, который указан вручную.

Нумератор

Для документов имеется возможность указывать нумератор, что, например, позволяет обеспечить общее пространство номеров для нескольких разных видов документов. Подробнее о нумераторах см. [здесь](#).

5.6.6.2. Серии кодов/номеров

Для справочников и планов видов характеристик имеется возможность указать, каким образом будет формироваться новый номер. Это можно сделать с помощью свойства **Серии кодов**:

- **Во всем справочнике/Во всем плане видов характеристик** – в процессе формирования нового кода для элемента справочника/плана видов характеристик будет сформирован код, уникальный во всем справочнике/плане видов характеристик;
- **В пределах подчинения** – в процессе формирования нового кода для элемента справочника/плана видов характеристик будет сформирован код, уникальный в пределах иерархии элемента (элементы, имеющие одного и того же родителя будут иметь различные коды, элементы, имеющие разных родителей могут иметь одинаковые коды);
- **В пределах подчинения владельцу** – в процессе формирования нового кода для элемента справочника будет сформирован код, уникальный в пределах подчинения (элементы, имеющие одного и того же владельца будут иметь различные коды; элементы, имеющие различных владельцев могут иметь одинаковые коды). Данный вариант применим только для справочников.

5.6.6.3. Периодичность кодов/номеров

Для документов и бизнес-процессов имеется возможность обеспечивать периодичность номеров. Это означает, что нумерация начнется с начала, каждый раз, когда начинается новый период. Периодичность настраивается с помощью свойства **Периодичность**. Данное свойство может принимать следующие значения:

- **Непериодический** – нумерация не зависит от временных периодов;
- **В пределах года** – нумерация начинается сначала при наступлении нового года;
- **В пределах квартала** – нумерация начинается сначала при наступлении нового квартала;
- **В пределах месяца** – нумерация начинается сначала при наступлении нового месяца;
- **В пределах дня** – нумерация начинается сначала при наступлении нового дня.

Контроль уникальности при периодической нумерации будет выполняться в рамках установленного периода (за год, квартал, месяц или день). Если задана непериодическая нумерация, то уникальность будет контролироваться во всех существующих объектах данного вида.

Принадлежность к тому или иному периоду определяется по дате документа или бизнес-процесса, а не по текущей дате. В связи с этим изменение даты документа или бизнес-процесса может привести к изменению номера в том случае, когда новая дата относится к другой периодичности (если таковая установлена).

5.6.6.4. Префиксы нумерации

При автоматизации различных задач может возникать задача обеспечения отдельной нумерации объектов одного вида в зависимости от некоторых условий. Например, в информационной базе ведется учет от лица нескольких фирм и для каждой фирмы нужна собственная нумерация документов. Данная задача решается с помощью **префиксов** кодов/номеров. Префикс кода/номера – это последовательность символов, предвещающих собственно код/номер, в рамках которых происходит нумерация объектов. Использование префиксов нумерации имеет несколько особенностей:

1. Префиксы используются только со строковым типом кода/номера.
2. Длина кода/номера, при использовании префикса, может оказаться больше, чем без использования префикса (на размер префикса и разделителя). Так, если изначально длина кода была задана в 6 символов, и необходимо начать использование префикса длиной 3 символа, то для сохранения пространства нумерации в 6 разрядов,

необходимо длину кода/номера увеличить до 9 символов (6 символов – собственно код и 3 символа – префикс).

3. Для автоматического изменения префиксов в зависимости от значений реквизитов объектов необходимо выполнять дополнительные действия на встроенном языке.

Для установки нового номера, с учетом префикса, следует использовать обработчик `ПриУстановкеНовогоКода()`/`ПриУстановкеНовогоНомера()` или методы `УстановитьНовыйКод()`/`УстановитьНовыйНомер()` с указанием нужного префикса. Если префикс не указан или указан не полностью, то среди существующих в информационной базе префиксов будет определен максимальный префикс (в алфавитном порядке) и в рамках этого префикса будет получен новый номер.

Рассмотрим примеры поведения системы нумерации.

Есть два объекта, у которых указаны номера **A-151** и **B-152**. При установке нового кода для префикса **A-** будет получен номер **A-152**, а для префикса **B-** – номер **B-153**. Если в системе есть объекты со следующими номерами: **AA151**, **AB150** и **BB114**, то, в зависимости от заданного префикса, возможны несколько вариантов полученных номеров (каждое описание действий префикса отталкивается от стартовых значений номеров и не учитывает действия других примеров):

- Указан префикс **AA**. В этом случае будет получен номер **AA152**.
- Указан префикс **AB**. В этом случае будет получен номер **AB151**.
- Указан префикс **A**. В этом случае будет получен номер **AB151**. Такой номер будет получен в результате того, что префикс указан не полностью и система доопределит максимальный префикс (в алфавитном порядке). Этим префиксом будет префикс **AB** и в рамках этого префикса будет получен новый номер: **AB152**.

Рассмотрим пример установки префикса в зависимости от значения реквизита объекта.

Допустим, что у нас существует документ, который содержит реквизит `Организация` типа `СправочникСсылка`. Реквизит `Префикс` (типа `Строка`) справочника `Организации` содержит префикс номера документа. Таким образом, для каждой организации будет поддерживаться собственная уникальная нумерация. Для назначения префикса следует в модуле объекта используемого документа создать обработчик `ПриУстановкеНовогоНомера()`, в котором будет устанавливаться префикс номера документа в зависимости от организации, выбранной в конкретном экземпляре документа.

Пример:

```
Процедура ПриУстановкеНовогоНомера(СтандартнаяОбработка, Префикс)
    Префикс = Организация.Префикс+"-";
КонецПроцедуры
```

[Копировать в буфер обмена](#)

5.6.6.5. Особенности работы механизма автонумерации

Контроль уникальности кода/номера объекта, если установлено свойство `Контроль уникальности`, выполняется при его записи, вне зависимости от режима автонумерации объектов и того, каким образом объекту присваивается новый код/номер – вручную или автоматически. Если объект записывается в «режиме загрузки» (при записи объекта свойство `ОбменДанными.Загрузка` установлено в значение `Истина`) – контроль уникальности кода/номера не производится.

5.6.7. Ввод по строке

5.6.7.1. Общая информация

Для прикладных объектов (справочники, документы, планы видов характеристик, планы счетов, планы видов расчета, регистры, бизнес-процессы и задачи, а также планы обмена) в свойстве `Ввод по строке` можно указать те реквизиты, по которым выполняется поиск. Если ввод по строке разрешен (заданы соответствующие реквизиты), то допустимо в поле ввода соответствующего типа не выполнять выбор из формы объекта, а ввести информацию, которая содержится в заданных реквизитах объекта.

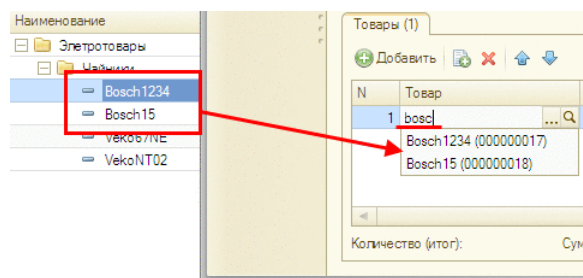


Рис. 124. Ввод по строке

Например, у нас есть справочник `Номенклатура`, в котором находится несколько товаров, начинающихся со слова `Bosch`. Тогда, введя в поле ввода номенклатуры слово `bosch`, мы получим список товаров, которые начинаются с этого слова (см. [рис. 124](#)).

В качестве реквизитов, по которым будет выполняться поиск, могут выступать:

- Для справочников, планов видов характеристик, планов счетов, планов видов расчета, планов обмена – `Код` и `Наименование`.
- Для документов – `Номер`.
- Для бизнес-процессов и задач – `Номер` и `Наименование`.
- Реквизиты, имеющие тип `Число` или `Строка`, для которых свойство `Индексировать` имеет значение `Индексировать` или `Индексировать с доп. упорядочиванием`. Примером последних может служить поле для ввода артикула, штрихкода, ИНН.

Для формирования списка реквизитов нужно нажать кнопку выбора и в открывшемся диалоге перенести в левый список те поля, по которым может производиться ввод по строке.

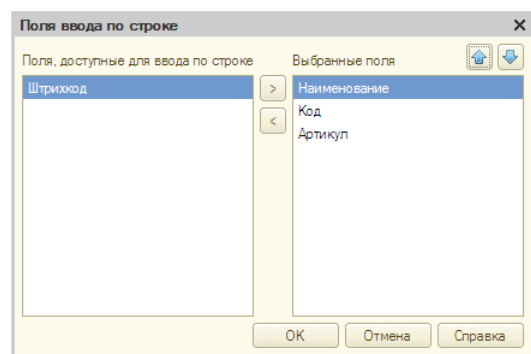


Рис. 125. Настройка ввода по строке

Если полей несколько, то установить порядок их следования. При поиске по строке поиск выполняется в полях в той последовательности, в которой они приведены в данном

диалоге. Например, для элемента справочника [Номенклатура](#) ввод может осуществляться как по коду, так и по артикулу. Если значение кода одного элемента совпадает со значением артикула другого, то в список найденные значения попадут в том порядке, в котором указаны при настройке.

ВНИМАНИЕ! При выполнении поиска данных (во время ввода по строке) действуют ограничения доступа к данным (см. [здесь](#)).

Кроме указания того, по каким полям будет выполняться поиск, система позволяет дополнительно настроить некоторые параметры поиска. Свойство [Способ поиска строки](#) предназначено для указания способа поиска строки: только в начале поля, выбранного для ввода по строке (значение [Начало](#)), или в любой части поля (значение [Любая часть](#)). В том случае, если способ поиска строки указан как [Начало](#), то поисковая строка не разбивается на слова, а введенный текст ищется целиком. Например, если поиск выполняется в следующих данных: [Иванов Максим](#) и [Максимов](#), то при вводе в поле ввода слова [Максим](#) будут найдены данные [Максимов](#), а при вводе в поле ввода [Иван Макс](#) не будет найдено ничего. Если строка поиска состоит из нескольких слов и поиск выполняется в любой части, то для поиска используется запрос, аналогичный следующему (для двух полей и двух слов):

[Копировать в буфер обмена](#)

(Поле1 ПОДОБНО "%Слово1%" ИЛИ Поле2 ПОДОБНО "%Слово1%") И (Поле1 ПОДОБНО "%Слово2%" ИЛИ Поле2 ПОДОБНО "%Слово2%")

Свойство [Полнотекстовый поиск](#) позволяет задействовать полнотекстовый поиск при организации ввода по строке. Полнотекстовый поиск нельзя использовать для таблиц внешних источников данных. Использование полнотекстового поиска обладает несколькими особенностями:

1. Полнотекстовый поиск всегда выполняется по началу слов.
2. Полнотекстовый поиск используется для первоначального отбора объектов, в которых будет выполняться дальнейший поиск (уменьшается количество объектов, в которых будет выполняться поиск).
3. В списке отобранных объектов будет выполняться поиск по полям, которые отобраны для использования в механизме ввода по строке способом, который задан свойством [Способ поиска строки](#).
4. Для каждого слова из поискового запроса в конце добавляется символ "*".

Указанные особенности могут привести к тому, что:

- Полнотекстовый поиск может чрезмерно сузить набор данных для поиска, если используется поиск по любой части вместе с полнотекстовым поиском. В результате данные найдены не будут. При этом отключение полнотекстового поиска приведет к нахождению нужных данных. Это произойдет потому, что полнотекстовый поиск ищет только по началу слова. Если искомый текст не находится в начале слова – ввод по строке не найдет такие данные.
- Неактуальный индекс полнотекстового поиска может негативно сказаться на результатах поиска.
- Если полнотекстовый поиск завершился с ошибкой, то будет применяться обычный поиск, в соответствии со свойством [Способ поиска строки](#). Это может привести к ситуации, когда найденный объект превратится в ненайденный (для способа поиска по любой части слова). Например: в поле ввода введен символ «а», который есть не только в начале, а и в середине массива слов, в котором выполняется поиск. Для большого набора данных полнотекстовый поиск завершится с ошибкой (слишком большой результат поиска) и с помощью обычного поиска будут найдены данные, которые внутри себя содержат букву «а». Затем пользователь введет еще одну букву (например, «ц») и поисковая строка станет «ац». В этом случае полнотекстовый поиск отберет только те данные, которые начинаются на «ац» и проигнорирует те данные, которые содержат «ац» внутри себя.

Если выпадающий список формируется системой (а не с помощью программного кода прикладного решения), то представления найденных объектов формируются по специальному алгоритму. Представление найденного объекта состоит из двух частей: представление реквизита, содержащего строку поиска (обязательная часть) и дополнительное представление (не обязательная часть).

Собственно представление формируется следующим образом:

- Определяется поле, содержащее поисковую строку:
 - В случае использования полнотекстового поиска или при обычном поиске по нескольким словам таким полем будет первое поле, из состава полей, указанных в свойстве [Ввод по строке](#), в котором найдено первое слово из поисковой строки.
 - При использовании обычного поиска по одному слову таким полем будет реквизит, в котором выполнялся поиск.
- Представление значения, расположенного в определенном поле будет являться обязательной частью представления найденного объекта для выпадающего списка.
- Определяется, необходимо или нет формировать дополнительное представление. Для этого определяется состав полей, который будет использоваться для формирования дополнительного представления:
 - Если у найденного объекта есть только одно поле, используемое для формирования представления, и поисковая строка обнаружена в этом реквизите – используются поля альтернативного представления.
 - В противном случае используются поля основного представления. Список полей основного и альтернативного представления приведен ниже.
- Если список полей для формирования дополнительного представления является пустым – дополнительное представление не формируется. В противном случае дополнительное представление формируется следующим образом:
 - В круглые скобки заключается текстовое представление всех полей, которые выбраны для формирования дополнительного представления, через ", ";
 - Сформированное дополнительное представление добавляется справа, через пробел, к обязательной части представления найденного объекта.
- Для формирования основного и альтернативного представления используются следующие поля:
 - Справочник, бизнес-процесс, план видов характеристик, план счетов, план видов расчета:
 - Основное представление: [Наименование](#), [Код](#).
 - Альтернативное представление: в зависимости от того, какое выбрано основное представление, для альтернативного представления используется оставшееся поле.
 - Документ:
 - Основное представление: [Номер](#), [Дата](#).
 - Альтернативное представление: отсутствует.
 - Задача:
 - Основное представление: [Наименование](#), [Номер](#), [Дата](#).
 - Альтернативное представление: отсутствует.
 - Внешний источник данных:
 - Основное представление: поле указанное в свойстве [Поле представления](#) таблицы внешнего источника данных (см. [здесь](#)).
 - Альтернативное представление: отсутствует.

Свойство [Режим получения данных выбора](#) управляет тем, как платформа осуществляет поиск данных при вводе по строке. Если свойство установлено в значение [Фоновый](#), то для получения данных используется фоновое задание, а в выпадающем списке отображается анимированная картинка, аналогичная фоновому выполнению отчета (см. [здесь](#)). Фоновое задание используется в том случае, если за 200 мс требуемые данные не получены для отображения пользователю. Для файлового варианта информационной базы возможно получение данных в фоновом режиме в том случае, если в момент получения данных не выполняется другое фоновое или регламентное задание. Во время фонового получения результатов выбора пользователь может продолжать редактировать поле ввода. Указанием значения [Непосредственно](#) в качестве

значения свойства приводит к тому, что поиск выполняется без использования фоновых заданий.

При настройке ввода по строке рекомендуется следовать следующим рекомендациям:

- Свойство **Режим получения данных выбора** устанавливать в значение **Фоновый** рекомендуется для объектов, в которых планируется большое количество данных.
- Полнотекстовый поиск рекомендуется включать для объектов, в которых ожидается большое количество данных.
- Поиск строки в любом месте рекомендуется включать совместно с фоновым поиском, либо в таблицах, в которых не ожидается большого количества данных. Желательно, так же при большом количестве данных, если включается поиск в любом месте, то включать и полнотекстовый поиск (чтобы большая часть поисков выполнялась бы через полнотекстовый поиск).

5.6.7.2. Особенности поведения поля ввода

При вводе данных в поле ввода следует учитывать некоторые особенности работы со списком выбора. Если в результате набора текста система однозначно идентифицирует объект, который пользователь хочет ввести, то происходит автоматическая подстановка найденного объекта в поле ввода.

Если в результате набора обнаружено несколько объектов, начинающихся с введенного текста, то список этих объектов будет расположен в выпадающем списке. Во время ввода пользователь может с помощью кнопок **Стрелка вверх** и **Стрелка вниз** передвигаться по списку, при этом продолжая набор текста в самом поле. В этом случае для осуществления выбора требуется выбрать необходимый элемент и подтвердить выбор нажатием кнопки **Enter** или **Tab**. Для формирования списка выбора для поля ввода используется следующий алгоритм:

1. Выполняется получение данных при выполнении поиска в первом поле (из списка выбранных полей) по началу строки и по подстроке. Количество записей, полученных в результате поиска по началу строки, будем называть **PC**, а количество записей, полученных в результате поиска по подстроке, будем называть **PP**.
2. Если **PC+PP<=10**, то в выпадающий список попадут все результаты каждого поиска.
3. Если **PP** и **PC** больше 5 (каждый), то в список выбора попадет по 5 записей из каждого списка.
4. Если любое из значений **PP** или **PC** меньше 5, то в выпадающий список попадут все записи из меньшего списка и столько записей из большего списка, чтобы дополнить выпадающий список до 10.
5. Если общее количество записей в выпадающем списке оказалось меньше 10, то выполняем алгоритм с шага 1 для следующего поля из списка выбранных полей, используя разницу между 10 и количеством полученных результатов как число необходимых к получению данных на следующем шаге.

5.6.7.3. Программное формирование списка выбора

Если разработчик не устраивает, каким образом формируется список выбора, он может переопределить его самостоятельно.

Сделать это можно двумя способами:

- Непосредственно в форме – в этом случае особое формирование списка выбора будет работать только для этого, единственного поля. Для этого необходимо определить обработчики событий **НачалоВыбора**, **АвтоПодбор** и **ОкончаниеВводаТекста**.
- В модуле менеджера соответствующего объекта – в этом случае особое формирование списка выбора будет выполняться для всех полей ввода, в которых вводятся значения используемого объекта.

Если поле связано с данными, имеющими тип ссылки на прикладной объект (например, **СправочникСсылка**, **ПеречислениеСсылка** и т. д.), то список выбора можно сформировать или в модуле менеджера соответствующего объекта, или с помощью обработчиков модуля формы. В остальных случаях список выбора можно сформировать только с помощью обработчиков модуля формы.

Подробнее остановимся на втором способе.

Для программного формирования списка выбора необходимо переопределить обработчик события **ОбработкаПолученияДанныхВыбора** менеджера объекта. Обработчик получает набор параметров, определяющих условия формирования списка выбора. Набор параметров представляет собой структуру, которая содержит:

- Строку поиска – текстовую строку, содержащую текст, который пользователь ввел в поле ввода. Данное свойство присутствует всегда (свойство **СтрокаПоиска**).
- Отбор – структура, которая описывает отбор так, как его описывает параметр формы **Отбор** для расширения формы для динамического списка. Данное свойство присутствует всегда.
- Отбор, указанный в данном свойстве, будет конвертироваться (по возможности) в запрос полнотекстового поиска, уточняя этот запрос для повышения скорости поиска.
- Свойство, которое указывает режим выбора групп и элементов (передается только для иерархических списков) – свойство **ВыборГруппИЭлементов**.
- Свойство, которое определяет способ поиска строки: по началу слова или по любой части (свойство **СпособПоискаСтроки**).
- Необходимость использования полнотекстового поиска (свойство **ПолнотекстовыйПоиск**).
- Режим получения данных выбора (свойство **РежимПолученияДанныхВыбора**).
- Кроме этого, в структуре передаются элементы, которые заданы в свойствах элемента формы – **Связи параметров выбора** (см. [здесь](#)) и **Параметры выбора** (см. [здесь](#)).

Кроме того, в обработчик передается переменная, в которую следует вернуть сформированный список выбора, и параметр **СтандартнаяОбработка**, который определяет поведение системы после того, как произойдет выход из обработчика.

Если разработчик устанавливает параметр **СтандартнаяОбработка** в значение **Ложь**, то ему необходимо самому полностью формировать список выбора.

Если флажок стандартной обработки установить в значение **Истина**, то можно предоставить системе возможность сформировать список выбора, но при этом можно модифицировать параметры выбора (добавить дополнительные значения отбора, изменить режим выбора групп и элементов и т. д.).

ПРИМЕЧАНИЕ 1. Если система сама формирует список выбора, при формировании списка учитываются ограничения доступа к данным (см. [здесь](#)).

ПРИМЕЧАНИЕ 2. Примеры, приведенные ниже, не являются законченными. Они предназначены для демонстрации различных механизмов получения списков выбора.

Так, следующий код в случае любого текста, вводимого пользователем, будет предоставлять выбор из трех товаров, с кодами **00000002**, **00000003** и **00000004**.

Пример:

```
Процедура ОбработкаПолученияДанныхВыбора (ДанныеВыбора, Параметры, СтандартнаяОбработка)
    ДанныеВыбора = Новый СписокЗначений;
    ДанныеВыбора.Добавить (Справочники.Товары.НайтиПоКоду ("00000002"));
    ДанныеВыбора.Добавить (Справочники.Товары.НайтиПоКоду ("00000003"));
    ДанныеВыбора.Добавить (Справочники.Товары.НайтиПоКоду ("00000004"));
    СтандартнаяОбработка = Ложь;
КонецПроцедуры
```

[Копировать в буфер обмена](#)

В следующем примере отбор, который сформировало поле ввода, будет расширен установкой дополнительного отбора так, чтобы в список выбора не попали услуги. При этом все формирование списка выбора берет на себя система.

Пример:

[Копировать в буфер обмена](#)

Процедура [ОбработкаПолученияДанныхВыбора](#) ([ДанныеВыбора](#), [Параметры](#), [СтандартнаяОбработка](#))
 Параметры.Отбор.Вставить ("Вид", [Перечисления.ВидыТоваров.Товар](#));
 КонецПроцедуры

В качестве последнего примера рассмотрим упрощенный случай реализации отбора полностью средствами встроенного языка. В этом примере будет рассмотрен отбор товаров, название которых начинается со строки, введенной пользователем в поле ввода.

Пример:

[Копировать в буфер обмена](#)

```
Процедура ОбработкаПолученияДанныхВыбора (ДанныеВыбора, Параметры, СтандартнаяОбработка)
  Запрос = Новый Запрос;
  Запрос.Текст = "ВЫБРАТЬ РАЗРЕШЕННЫЕ
  | Товары.Ссылка как Товар
  | ИЗ
  | Справочник.Товары КАК Товары
  | ГДЕ
  | Товары.Наименование ПОДОБНО &Наименование";
  Запрос.УстановитьПараметр ("Наименование", Параметры.СтрокаПоиска + "%");
  Результат = Запрос.Выполнить();
  ТаблицаРезультатов = Результат.Выгрузить();
  МассивТоваров = ТаблицаРезультатов.ВыгрузитьКолонку ("Товар");
  ДанныеВыбора = Новый СписокЗначений;
  ДанныеВыбора.ЗагрузитьЗначения (МассивТоваров);
  СтандартнаяОбработка = Ложь;
КонецПроцедуры
```

Также следует обратить внимание на еще один способ формирования списка выбора: передачу в качестве значения элемента списка значений не ссылки на искомый объект (как в примерах выше), а структуру особого содержания.

Эта структура состоит из следующих элементов:

- **Значение** – собственно значение выбираемого элемента. Элемент структуры с таким именем должен быть обязательно.
- **ПометкаУдаления** – признак, что выбираемое значение помечено на удаление в информационной базе. Элемент структуры с таким именем не является обязательным.
- **Предупреждение** – строка с текстом предупреждения, которое отобразит «1С:Предприятие» при выборе такого элемента из списка значений. Элемент структуры с таким именем не является обязательным.

Если в структуре свойство **ПометкаУдаления** равно значению **Истина** и не указано свойство **Предупреждение**, то система автоматически сформирует текст предупреждения. Если свойство **Предупреждение** указано, то отображается именно оно. Следует помнить, что текст **Предупреждение** завершается вопросом **Продолжить?** и отображается в качестве вопроса с вариантами ответа **Да** и **Нет**.

Далее будет показана модификация предыдущего примера, в котором для складов с установленным в значение **Истина** реквизитом **НеИспользовать** будет сформировано предупреждение **Этот склад не должен использоваться**.

Также можно совмещать в одном списке выбора и значения типа **Структура**, и простые значения.

Пример:

[Копировать в буфер обмена](#)

```
Процедура ОбработкаПолученияДанныхВыбора (ДанныеВыбора, Параметры, СтандартнаяОбработка)
  СтандартнаяОбработка = Ложь;
  ДанныеВыбора = Новый СписокЗначений;
  // Сформируем список с предупреждениями
  Запрос = Новый Запрос;
  Запрос.Текст = "ВЫБРАТЬ
  | Склады.Ссылка,
  | Склады.Наименование,
  | Склады.НеИспользовать
  | ИЗ
  | Справочник.Склады КАК Склады";
  Результат = Запрос.Выполнить();
  ВыборкаДетальныеЗаписи = Результат.Выбрать();
  Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    Структура = Новый Структура ("Значение", ВыборкаДетальныеЗаписи.Ссылка);
    // Заполним предупреждение
    Если ВыборкаДетальныеЗаписи.НеИспользовать Тогда
      Структура.Вставить ("Предупреждение", "Этот склад не должен использоваться!");
    КонецЕсли;
    Элемент = ДанныеВыбора.Добавить();
    Элемент.Значение = Структура;
    Элемент.Представление = ВыборкаДетальныеЗаписи.Наименование;
  КонецЦикла;
КонецПроцедуры
```

ПРИМЕЧАНИЕ. Если представления элементов (в том числе и элементов структуры с именем **Значение**) не указаны в явном виде, то они будут получены автоматически.

5.6.8. Формы

Форма – это объект, созданный для ввода или просмотра какой-либо информации, а также для управления различными процессами. С помощью форм программа запрашивает у пользователя ту информацию, которая необходима ей для дальнейшей работы, либо выдает какую-либо информацию пользователю для просмотра и редактирования.

В тех случаях, когда необходимо обеспечить функционирование конфигурации одновременно в обычном и управляемом режимах, можно использовать дополнительные формы объектов метаданных. При этом «1С:Предприятие» будет автоматически выбирать для использования ту форму, которая соответствует текущему режиму работы.

Рис. 126. Основные и дополнительные формы

Тонкий клиент и веб-клиент оперируют только управляемыми формами. Значит:

- если обе назначенные формы – управляемые, то будет открыта та форма, которая назначена как основная;
- если среди назначенных форм есть только одна управляемая, будет открыта именно она;
- если управляемая форма не назначена, она будет сгенерирована автоматически.

Толстый клиент при выборе формы старается выбрать форму, которая максимально соответствует текущему режиму запуска:

- если никакие формы не назначены, то будет сгенерирована:
 - в обычном режиме – обычная форма;
 - в управляемом режиме – управляемая форма;
- если назначена только одна форма, то именно она и будет открыта;
- если назначены две формы, обычная и управляемая, то будет открыта:
 - в обычном режиме – обычная форма;
 - в управляемом режиме – управляемая форма.
- если назначены две обычные или две управляемые формы, то будет открыта та форма, которая назначена основной.

Двойной комплект форм можно использовать в тех случаях, когда вы переводите вашу конфигурацию из неуправляемого в управляемый режим, либо когда необходимо, например, часть возможностей конфигурации сделать доступными в веб-клиенте (или тонком клиенте). В этом случае можно реализовать необходимые возможности в управляемых формах и указать их в качестве дополнительных форм. Тогда при работе в режиме веб-клиента (или тонкого клиента) будут использоваться нужные формы (управляемые).

Однако следует учитывать, что на получение форм по умолчанию в толстом клиенте оказывают влияние свойства конфигурации [Использовать управляемые формы в обычном приложении](#) и [Использовать обычные формы в управляемом приложении](#):

- Если свойство конфигурации [Использовать управляемые формы в обычном приложении](#) имеет значение *Ложь*, то в толстом клиенте, в обычном режиме, при получении формы по умолчанию обязательно должна быть получена обычная форма. Если ни основная, ни дополнительная форма не является обычной, то генерируется обычная форма. Однако если задана основная форма констант (свойство [Основная форма констант](#)), то будет использоваться именно она, вне зависимости от состояния флажка [Использовать управляемые формы в обычном приложении](#).
- Если свойство конфигурации [Использовать обычные формы в управляемом приложении](#) имеет значение *Ложь*, то в толстом клиенте, в управляемом режиме, при получении формы по умолчанию обязательно должна быть получена управляемая форма. Если ни основная, ни дополнительная форма не является обычной, то генерируется управляемая форма.

Если в результате открывается форма, которая автоматически сгенерирована системой и режим запуска клиентского приложения отличается от режима запуска, установленного в свойстве конфигурации [Основной режим запуска](#), то формы объектов и формы наборов записей открываются в режиме [Только просмотр](#). Также в формах списков и формах объектов будут отсутствовать команды [Создать на основании](#), а в формах списков – команда [Создать](#), если для выполнения этих команд системе необходимо автоматически сгенерировать форму, подходящую для текущего режима запуска и режим запуска клиентского приложения отличается от режима запуска, установленного в свойстве конфигурации [Основной режим запуска](#).

Если в качестве основного режима запуска прикладного решения выбрано управляемое приложение, то для каждой управляемой формы прикладного решения можно выбрать назначение формы. Свойство формы [Назначения использования](#) эквивалентно (по составу и применению) аналогичному свойству конфигурации (см. [здесь](#)). Поведение редактора формы зависит от пересечения множеств назначений конфигурации и конкретной формы.

Если в пересечении отсутствует значение [Персональный компьютер](#), то:

- В палитре свойств недоступны свойства реквизитов, параметров, команд, элементов, которые не поддерживаются мобильной платформой;
- Недоступно редактирование контекстного меню элемента формы;
- Недоступно редактирование расширенной подсказки для элемента формы;
- Недоступны те виды полей формы, которые не поддерживаются мобильной платформой (см. [здесь](#));
- Редактор настроек динамического списка не позволяет разворачивать доступные поля (связано с тем, что на мобильной платформе динамический список работает строго с одной таблицей).

Если в назначении формы отсутствует значение [Мобильное приложение](#), то она не включается в конфигурацию мобильного приложения.

5.6.9. Команды

Для выполнения операций, связанных с конкретным объектом метаданных, существуют команды этого объекта. При этом непараметризованные команды объекта будут доступны в командном интерфейсе тех подсистем, в состав которых входит объект метаданных. Если команда является параметризованной, то она будет доступна в тех формах, которые содержат реквизиты формы (включая подчиненные реквизиты первого уровня основного реквизита формы) того же типа, что и тип параметра команды.

Для команд требуется написать процедуру выполнения команды. Для этого служит модуль команды, в котором необходимо реализовать обработчик [ОбработкаКоманды\(\)](#). Данная процедура должна предваряться директивой [&НаКлиенте](#), так как выполнение команды происходит в клиентском приложении. Однако другие процедуры и функции, расположенные в модуле команды, могут предваряться другими директивами, если это необходимо для выполнения команды. Подробнее про директивы компиляции см. [здесь](#).

Модуль команды может содержать, например, открытие формы отчета с предварительно установленным ему параметром для вывода на печать карточки определенного бухгалтерского счета или открытие формы списка товаров с установленным отбором по виду товара. Если команда, принадлежащая группе [Панель навигации](#), должна открывать одну и ту же форму в разных закладках, следует при получении (или открытии) формы указывать ключ уникальности или ключевой параметр формы.

ПРИМЕЧАНИЕ. Если команда расположена в панели навигации основного окна приложения (см. [здесь](#)), то повторный выбор этой команды не приведет к ее (команды) повторному выполнению. Это относится и к стандартным командам, и к командам, созданным в конфигурации.

5.6.10. Механизм заполнения реквизитов новых объектов

Существует возможность заполнения реквизитов новых объектов при интерактивном создании (кроме копирования), при вводе на основании, при использовании методов [ОткрытьФорму\(\)](#)/[ПолучитьФорму\(\)](#) или при явном вызове метода [Заполнить\(\)](#). Это заполнение может выполняться:

- значениями отбора при вводе из списка;
- конкретными значениями, указанными в конфигурации в свойствах реквизитов (значения заполнения);
- значениями, которые разработчик указал в обработчике события [ОбработкаЗаполнения\(\)](#).

Механизм обработки заполнения реализован для следующих объектов:

- планы обмена,

- справочники,
- документы,
- планы видов характеристик,
- планы счетов,
- планы видов расчета,
- наборы записей регистров сведений,
- бизнес-процессы,
- задачи.

Для получения данных заполнения в обработчике `ОбработкаЗаполнения()` существует параметр `ДанныеЗаполнения`. В зависимости от того, каким образом вызван обработчик, значение параметра `ДанныеЗаполнения` может принимать разные значения:

- **Ввод на основании** – в качестве значения параметра передается ссылка на объект-основание. При этом значение параметра `ДанныеЗаполнения` будет являться ссылкой на объект-основание базы данных.
- **Ввод из списка с установленным отбором** – в качестве значения параметра передается структура, элементами которой становятся используемые элементы отбора с установленным видом сравнения `Равно` или `В списке` (при этом в списке находится единственное значение). При создании нового документа из формы журнала документов элементы отбора по графам журнала предварительно преобразуются таким образом, чтобы в качестве имени элемента структуры `ДанныеЗаполнения` выступало имя реквизита документа, а не имя графы журнала.
- **Ввод нового объекта или записи без использования отбора** – значение параметра равно `Неопределено`.
- **Ввод нового объекта или записи с помощью глобальной команды** – значение параметра равно `Неопределено`.
- **Программное выполнение методов `ОткрытьФорму()`/`ПолучитьФорму()`** – в случае, если исполнение методов `ОткрытьФорму()`/`ПолучитьФорму()` приводит к созданию формы нового объекта, в качестве значения параметра обработчика `ДанныеЗаполнения` будет передано значение (типа `Структура`) параметра формы `ЗначенияЗаполнения`. Если параметр `ЗначенияЗаполнения` не задан в параметрах формы, то значение параметра обработчика будет равно `Неопределено`.
- **Программный вызов метода объекта `Заполнить()`** – в качестве значения параметра передается та информация, которая передана в качестве параметра метода `Заполнить()`.

ПРИМЕЧАНИЕ. Операция интерактивного создания нового объекта с помощью копирования обрабатывается в обработчике `ПриКопировании()` соответствующего объекта.

На заполнение реквизитов из данных заполнения оказывает влияние свойство `Заполнять из данных заполнения` реквизита объекта метаданных. Если это свойство имеет значение `Истина`, то реквизиты будут заполняться системой из данных заполнения автоматически. Если свойство имеет значение `Ложь` или в данных заполнения нет необходимого значения, то система будет пытаться заполнить реквизит из свойства `Значение заполнения`.

ВНИМАНИЕ! Разработчик может управлять заполнением стандартных реквизитов объектов наравне с самостоятельно созданными реквизитами. Например, можно запретить заполнение стандартного реквизита `Родитель`, и тогда он не будет автоматически заполняться текущей группой.

Если после исполнения обработчика `ОбработкаЗаполнения()` параметр `СтандартнаяОбработка` равен значению `Истина`, то система автоматически заполнит те реквизиты (включая стандартные), для которых в метаданных установлены свойства `Заполнять из данных заполнения` или `Значение заполнения` и которые не заполнены в обработчике (значение равно значению по умолчанию для типа реквизита). Свойство `Заполнять из данных заполнения` автоматически устанавливается системой для некоторых стандартных реквизитов ряда объектов:

- для справочников – реквизиты `Родитель` и `Владелец`;
- для планов счетов, планов видов характеристик и планов видов расчета – реквизит `Родитель`;
- для регистра сведений – ведущие измерения;
- для стандартных реквизитов остальных объектов данное свойство автоматически не устанавливается.

При этом данные для стандартного заполнения берутся системой из одноименных реквизитов данных, передающихся в параметре `ДанныеЗаполнения`.

Данные заполнения передаются в создаваемую форму объекта в качестве стандартного параметра формы `ЗначенияЗаполнения` и передаются из этого параметра расширением формы для заполнения объекту. Также имеется возможность программно задать параметр формы `ЗначенияЗаполнения` нового объекта, при этом будут выполнены все действия, которые исполняются при интерактивном создании объекта.

ПРИМЕЧАНИЕ. При программном создании нового объекта обработка заполнения системой автоматически не вызывается. Для вызова обработчика заполнения существует метод `Заполнить()`.

Значение заполнения – это свойство реквизита объекта метаданных, позволяющее задать значение по умолчанию, которое может принимать реквизит при интерактивном создании объекта.

ВНИМАНИЕ! Заполнение реквизитов из свойства `Значение заполнения` происходит после вызова обработчика `ОбработкаЗаполнения()`. Реквизит будет заполнен в том случае, если его значение не заполнено ранее (в обработчике `ОбработкаЗаполнения()` или механизмом стандартного заполнения).

Тип значения заполнения совпадает с типом реквизита. При этом в качестве значения этого свойства можно указывать значения примитивных типов или predefined данные.

5.6.11. Проверка заполнения реквизитов

5.6.11.1. Общая информация

В информационной системе данные могут вноситься множеством различных способов, и часто они могут быть некорректны. Поэтому при разработке решения бывает необходимо прилагать много усилий на реализацию проверки правильности вводимых данных и уведомления пользователя о некорректности введенной информации.

Механизм проверки заполнения позволяет существенно упростить процесс разработки конфигураций.

Платформа поддерживает автоматическую проверку указанных реквизитов прикладных объектов и форм, а также позволяет выполнить процесс проверки в модуле.

Платформа выполняет автоматическую проверку заполнения:

- констант;
- справочников, документов, отчетов, обработок, планов видов характеристик, планов счетов, планов видов расчета, бизнес-процессов, задач:
 - реквизитов и стандартных реквизитов,
 - табличных частей,
 - реквизитов и стандартных реквизитов табличных частей;

- наборов записей регистров бухгалтерии, регистра сведений, регистра накопления, регистра расчета, перерасчетов, последовательностей:
 - измерений,
 - ресурсов,
 - реквизитов и стандартных реквизитов;
- реквизитов форм;
- реквизитов форм отчетов;
- реквизитов форм обработок.

ПРИМЕЧАНИЕ. Проверка заполнения реквизитов выполняется аналогично функции `ЗначениеЗаполнено()`. Проверка заполнения табличных частей подразумевает, что табличная часть считается заполненной, когда в ней присутствует хотя бы одна строка.

Проверка заполнения может быть вызвана двумя способами:

- вызовом метода `ПроверитьЗаполнения()` (у объекта или формы);
- автоматически.

ВНИМАНИЕ! Если свойство конфигурации `Режим совместимости` установлено в значение `Версия 8.1`, то автоматическая проверка заполнения не работает.

5.6.11.2. Установки по умолчанию

По умолчанию свойство устанавливается в значение `Выдавать ошибку` для следующих стандартных реквизитов:

- `ПланОбмена` – `Наименование`;
- `Справочник` – `Владелец`, `Наименование`;
- `Документ` – `Дата`;
- `ПланВидовХарактеристик` – `Наименование`;
- `ПланСчетов` – `Код`, `Наименование`;
- `ПланСчетов.ВидыСубконто` – `ВидСубконто`;
- `ПланВидовРасчета` – `Наименование`;
- `ПланВидовРасчета.ВедущиеВидыРасчета` – `ВидРасчета`;
- `ПланВидовРасчета.БазовыеВидыРасчета` – `ВидРасчета`;
- `ПланВидовРасчета.ВытесняющиеВидыРасчета` – `ВидРасчета`;
- `РегистрСведений` – `Период`;
- `РегистрНакопления` – `Период`;
- `РегистрБухгалтерии` – `Период`;
- `РегистрРасчета` – `ПериодРегистрации`, `ВидРасчета`, `ПериодДействияНачало`, `ПериодДействияКонец`;
- `БизнесПроцесс` – `Дата`;
- `Задача` – `Наименование`.

5.6.11.3. Порядок работы

Автоматическая проверка заполнения вызывается расширением формы перед интерактивной записью всех объектов, кроме документов, бизнес-процессов, отчетов и обработок. Для этих объектов определено следующее поведение:

- для документов:
 - проверка заполнения вызывается расширением формы перед проведением, если свойство `Проведение` установлено в значение `Разрешить`;
 - проверка заполнения вызывается расширением формы перед записью, если свойство `Проведение` установлено в значение `Запретить`;
- для бизнес-процессов проверка заполнения вызывается расширением формы перед стартом;
- для отчетов проверка заполнения вызывается расширением формы при нажатии кнопки `Сформировать`;
- для обработок проверка заполнения вызывается расширением формы в случаях, если нажаты кнопки, связанные со стандартными командами формы `ОК`, `Да`, `Повторить`, `Пропустить`.

Если реквизит является основным реквизитом формы следующих типов: `СправочникОбъект`, `ДокументОбъект`, `ОтчетОбъект`, `ПланВидовХарактеристикОбъект`, `ПланСчетовОбъект`, `ПланВидовРасчетовОбъект`, `БизнесПроцессОбъект`, `ЗадачаОбъект`, то для значения такого реквизита также будет вызвана проверка заполнения.

Для вызова проверки заполнения системой необходимо, чтобы у формы (с которой происходит работа) было установлено свойство `Проверять заполнение автоматически`. В этом случае вначале будет вызван обработчик `ОбработкаПроверкиЗаполненияНаСервере()` формы, а затем обработчик `ОбработкаПроверкиЗаполнения()` модуля объекта.

ВНИМАНИЕ! Если у формы свойство `Проверять заполнение автоматически` установлено в значение `Истина` при выполнении стандартных команд `Записать` (`Провести для документов` и т. д.), а также стандартных команд формы `ОК`, `Да`, `Повторить`, `Пропустить`, будет вызван метод `ПроверитьЗаполнение()`. В противном случае проверка заполнения не вызывается ни для формы, ни для объекта.

Процесс проверки заполнения происходит следующим образом:

- Формируется список имен реквизитов формы, для которых возможна проверка заполнения и для которых свойство `ПроверкаЗаполнения` установлено в значение `ВыдаватьОшибку`. В этот список не будут включены имена реквизитов, тип которых не поддерживает проверку заполнения (например, `СправочникОбъект`), но будет включено имя основного реквизита формы.
- В автоматически формируемый список реквизитов не будут включены реквизиты формы, отключенные с помощью функциональных опций.
- Вызывается обработчик события формы `ОбработкаПроверкиЗаполненияНаСервере`, в котором разработчик может описать свой алгоритм проверки заполнения или изменить состав проверяемых реквизитов. В обработчик будет передан сформированный список имен реквизитов. Если в обработчике необходимо добавить к списку какие-либо реквизиты, это можно сделать только для реквизитов вышеперечисленных типов (для которых возможна проверка заполнения в форме) и основного реквизита.

Добавление в список имен реквизитов объектного типа (например, [СправочникОбъект](#)) вызовет исключение при дальнейшей автоматической проверке. Добавление в список имени несуществующего реквизита вызовет исключение при дальнейшей автоматической проверке.

- После завершения работы обработчика события механизм проверки заполнения получает обратно список имен, проверяемых реквизитом (который, возможно, был изменен в обработчике). Система анализирует список реквизитов и проверяет заполненность каждого реквизита. Если реквизит является основным реквизитом объектного типа (например, [Объект](#) типа [СправочникОбъект](#)), будет вызвана проверка заполнения самого объекта. Если реквизит является реквизитом объектного типа, но не основным, будет вызвано исключение.

Разработчик имеет возможность влиять на процесс проверки путем определения в модуле объекта, набора записей и в модуле формы обработчика события [ОбработкаПроверкиЗаполнения](#).

Определив обработчик события, разработчик получает полный контроль над проверкой заполнения. В параметре [ПроверяемыеРеквизиты](#) обработчик получает массив реквизитов, для которых в режиме Конфигуратор указано, что они должны проверяться. Разработчик может произвольно модифицировать этот массив:

- удалять те реквизиты, проверку заполнения которых он реализует сам или считает, что их не нужно проверять в данный момент;
- добавлять необходимые реквизиты, для которых должна выполняться проверка заполнения.

В случае если разработчик не устраивает стандартная процедура проверки, он может написать алгоритм проверки сам и, используя объект [СообщениеПользователю](#), сообщить об имеющихся проблемах пользователю.

Если в процессе проверки разработчик вывел пользователю сообщения об ошибках, необходимо установить параметр [Отказ](#) в значение [Истина](#), чтобы уведомить платформу о том, что текущее действие не может быть завершено.

Параметр события [ПроверяемыеРеквизиты](#) содержит имена атрибутов в следующем формате:

- для реквизитов и констант – [ИмяРеквизита](#), например, [Поставщик](#);
- для табличных частей – [ИмяТабличнойЧасти](#), например, [Товары](#);
- для реквизитов табличных частей – [ИмяТабличнойЧасти.ИмяРеквизита](#), например, [Товары.Номенклатура](#);
- для реквизитов форм – [ИмяРеквизита](#), например, [ДокументОбъект](#).

Для реквизитов, входящих в состав функциональных опций (см. [здесь](#)) без параметров, значение опции учитывается при проверке заполнения. Если функциональная опция включена, то реквизит будет включен в список проверяемых реквизитов; если опция выключена, реквизит не будет включен в список проверяемых реквизитов. Это значит, что отключенное поле не будет передано через параметры [ПроверяемыеРеквизиты](#) в [ОбработкаПроверкиЗаполнения](#).

Реквизиты, входящие в состав функциональной опции с параметрами (см. [здесь](#)), всегда включаются в список проверяемых реквизитов (параметр [ПроверяемыеРеквизиты](#)). Удаление реквизита из списка проверяемых реквизитов в таких случаях необходимо выполнять в обработчике [ОбработкаПроверкиЗаполнения](#). Для этого следует получить значение функциональной опции, указав в качестве параметров необходимые данные объекта.

5.6.11.4. Правила отображения отметки незаполненного

На отображение отметки незаполненного влияют следующие свойства элементов конфигурации:

- свойство [Проверка заполнения](#) реквизита или стандартного реквизита объекта метаданных;
- свойство [Проверка заполнения](#) реквизита формы;
- свойство [Автоотметка незаполненного](#) элемента формы;
- свойство [ОтметкаНезаполненного](#) элемента формы (доступно только для программного изменения).

Свойство элемента формы [Автоотметка незаполненного](#) может быть установлено в значение:

- [Авто](#) – отметка будет отображаться, если у реквизита формы или реквизита (стандартного реквизита) объекта метаданных свойство [Проверка заполнения](#) установлено в значение [Выдавать ошибку](#) и реквизит, связанный с элементом формы, содержит пустое значение.
- [Да](#) – отметка будет отображаться для незаполненного элемента вне зависимости от того, в каком состоянии находится свойство [Проверка заполнения](#).
- [Нет](#) – отметка не будет отображаться для незаполненного элемента вне зависимости от того, в каком состоянии находится свойство [Проверка заполнения](#).

При начале интерактивного изменения поля ввода отметка незаполненного перестает отображаться, при этом значение свойства [ОтметкаНезаполненного](#) не изменяется. Отображение отметки восстанавливается (если это необходимо) после того, как в поле ввода завершено интерактивное редактирование или установлено значение. При этом свойство [ОтметкаНезаполненного](#) может быть изменено системой в том случае, если свойство [АвтоОтметкаНезаполненного](#) установлено в значение [Истина](#):

- если значение в поле не заполнено, то свойство [ОтметкаНезаполненного](#) будет установлено в значение [Истина](#);
- если в поле значение заполнено, то свойство [ОтметкаНезаполненного](#) будет установлено в значение [Ложь](#).

ПРИМЕЧАНИЕ. Если свойство [ОтметкаНезаполненного](#) установлено у таблицы, то для таблицы, в которой нет строк, будет подсвечена первая строка, а для заполненной таблицы – все строки.

Кроме того, отметка незаполненного также будет отображена, если в окне сообщений есть сообщение (см. [здесь](#)), связанное с конкретным полем формы, вне зависимости от установленных свойств элемента формы и связанных с ним реквизитов. После очистки окна сообщений отметка незаполненного снимается у тех элементов формы, значение свойства [ОтметкаНезаполненного](#) которых равно значению [Ложь](#).

5.6.12. Индексирование реквизитов объектов

Большинство прикладных объектов конфигурации имеют в составе подчиненных объектов группу [Реквизиты](#). В этой группе указываются дополнительные характеристики объектов.

В режиме 1С:Предприятие часто требуется осуществлять отбор данных по значению какого-либо реквизита или сортировать списки данных по реквизитам. Средства «1С:Предприятия» позволяют выполнить подобную задачу, однако если данных достаточно много, такая задача может выполняться долго.

Чтобы ускорить эту работу, следует реквизитам, по которым будет выполняться отбор или сортировка, устанавливать свойство [Индексировать](#). Если свойство установлено (выбрано значение [Индексировать](#) или [Индексировать с доп. упорядочиванием](#)), то подобные задачи будут выполняться эффективнее. Для примитивных типов реквизитов указание индексирования предоставляет пользователям средство сортировки списка по щелчку мыши в области заголовка колонки.

Наряду с сортировкой по реквизиту или отбором данных по значению какого-либо реквизита часто требуется, чтобы в результирующем списке данные были дополнительно отсортированы по основному представлению (наименованию или коду), т. е. в пределах одного значения реквизита записи были отсортированы по представлению. Добиться правильного результата можно, если выбрано значение индексирования [Индексировать](#), а в условиях сортировки списка указаны реквизит и представление.

Если возникает необходимость минимизации времени таких отборов или сортировки, то для реквизита следует выбрать значение индексирования [Индексировать с доп. упорядочиванием](#) (если выбор такого значения возможен).

ВНИМАНИЕ! Механизм дополнительного упорядочивания будет эффективно использоваться только в том случае, когда в режиме 1С:Предприятие в условиях сортировки списка указан определенный порядок сортировки: сначала по реквизиту, а затем по представлению.

Если такие условия сортировки не задаются, то использование значения [Индексировать с доп. упорядочиванием](#) не имеет смысла, т. к. оно будет эквивалентно обычному индексированию, однако размер индекса при этом будет больше.

Индексирование с дополнительным упорядочиванием используется для реквизитов справочников, документов, планов видов характеристик, планов счетов, планов видов расчетов.

Для реквизитов регистров допускается использование обычного индексирования.

5.6.13. Права

Права доступа к объектам конфигурации можно редактировать как с помощью редактора прав доступа ролей (см. [здесь](#)), так и с помощью окна редактирования объекта (см. [здесь](#)).

В окне редактирования объекта выполняется настройка прав доступа для данного объекта во всех ролях, которые существуют в системе.

Имеется возможность задавать права доступа:

- для собственно объектов метаданных,
- табличных частей,
- реквизитов объектов.

5.6.14. Быстрый выбор

За способ выбора объектов при заполнении полей ввода отвечают свойства [Способ выбора](#) и [Быстрый выбор](#).

Свойство [Быстрый выбор](#) отвечает за режим выбора по умолчанию. Свойство доступно только в том случае, если свойство [Способ выбора](#) имеет значение [Обоими способами](#). Покажем работу свойств [Быстрый выбор](#) на примере.

Допустим, у нас есть справочник [Склады](#). У него свойство [Способ выбора](#) равно [Обоими способами](#), а также установлено свойство [Быстрый выбор](#). В этом случае по умолчанию выбор из значений справочника [Склады](#) во всей конфигурации будет осуществляться в режиме быстрого выбора. Если отключить свойство [Быстрый выбор](#), то выбор по умолчанию станет [Из формы](#).

Также имеется возможность явного переопределения режима выбора для реквизита объекта метаданных и/или элемента формы. Для этого достаточно изменить значение [Авто](#) свойства [Быстрый выбор](#) на значение [Использовать](#) – для разрешения быстрого выбора или [Не использовать](#) – для запрещения быстрого выбора. По умолчанию свойство [Быстрый выбор](#) реквизита объекта метаданных и элемента формы установлено в значение [Авто](#).

Таким образом, получается трехуровневая схема управления режимом выбора:

1. Вначале платформа анализирует свойство [Быстрый выбор](#) элемента формы.
2. Если свойство равно [Авто](#), то оно анализируется у реквизита объекта.
3. В случае равенства [Авто](#) выполняется анализ свойств [Способ выбора](#) и [Быстрый выбор](#) прикладного объекта, соответствующего типу реквизита. Если на каком-то из первых двух уровней (элемент формы и реквизит объекта) значение свойства [Быстрый выбор](#) отлично от [Авто](#), анализ прекращается и выполняется выбор в определенном режиме.

ПРИМЕЧАНИЕ. Если свойство [Способ выбора](#) прикладного объекта равно [Из формы](#) или [Быстрый выбор](#) (но только не [Обоими способами](#)), то установка свойств [Быстрый выбор](#) реквизита и элемента формы никак не влияет на способ выбора.

5.6.15. История выбора при вводе

В том случае, если поле ввода формы связано с данными ссылочного типа (справочники, документы, перечисления, бизнес-процессы, задачи, планы видов характеристик, планы видов расчета, планы счетов и планы обмена), для него доступно управление отображением и формированием истории выбора при вводе. Управление основано на значении свойства [ИсторияВыбораПриВводе](#), которое присутствует в свойствах объекта конфигурации, реквизита объекта конфигурации и поля ввода управляемой формы.

Управление историей выбора реализовано с помощью трехуровневой схемы управления:

1. Вначале платформа анализирует свойство [ИсторияВыбораПриВводе](#) для поля ввода формы.
2. Если значение свойства равно [Авто](#), то выполняется анализ свойства [ИсторияВыбораПриВводе](#) у реквизита, отображаемого полем ввода.
3. Если значение свойства равно [Авто](#), выполняется анализ свойства [ИсторияВыбораПриВводе](#) у объекта конфигурации, соответствующего типу реквизита.

Значение свойства [ИсторияВыбораПриВводе](#) может принимать значение [Авто](#) и [Не использовать](#). В том случае, если свойство имеет значение [Авто](#) на всех «уровнях» принятия решения, то история выбора будет отображаться и сохраняться при условии выполнения всех следующих условий:

- Поле ввода отображает значение ссылочного типа;
- Не установлен режим выбора из списка;
- Не используется быстрый выбор (см. [здесь](#)).

Во всех остальных случаях история выбора не будет сохраняться и отображаться.

5.6.16. Создание нового элемента при вводе

В том случае, если поле ввода формы связано с данными ссылочного типа (справочники, документы, перечисления, бизнес-процессы, задачи, планы видов характеристик, планы видов расчета, планы счетов и планы обмена), для него доступно управление созданием нового элемента из поля ввода. Управление основано на значении свойства [СозданиеПриВводе](#), которое присутствует в свойствах объекта конфигурации, реквизита объекта конфигурации и поля ввода управляемой формы.

Управление возможностью создания элемента из выпадающего списка реализовано с помощью трехуровневой схемы управления:

1. Вначале платформа анализирует свойство [КнопкаСоздания](#) для поля ввода формы.
2. Если значение свойства равно [Авто](#), то выполняется анализ свойства [СозданиеПриВводе](#) у реквизита, отображаемого полем ввода.
3. Если значение свойства равно [Авто](#), выполняется анализ свойства [СозданиеПриВводе](#) у объекта конфигурации, соответствующего типу реквизита.

Анализ прекращается после того, как на каком-то из уровней обнаружено значение свойство, отличное от значения [Авто](#).

В том случае, если создание элемента из поля ввода необходимо выполнить особым образом, разработчик должен обработать событие [Создание](#). Событие возникает при нажатии кнопки создания в выпадающем списке.

Если в обработчике данного события параметр [СтандартнаяОбработка](#) установлен в значение [Ложь](#), то система не будет выполнять никаких действий, и ожидается, что открытие формы нового элемента разработчик выполнит самостоятельно. Если параметр [СтандартнаяОбработка](#) установлен в значение [Истина](#), то система выполнит открытие основной формы объекта, при этом в форму будут переданы следующие параметры:

- [РежимВыбора](#), тип [Булево](#), значение – [Истина](#). Признак того, что форма находится в режиме выбора.
- [ТекстЗаполнения](#), тип [Строка](#). Параметр содержит текст, введенный в поле ввода перед нажатием кнопки создания нового элемента.
- [ЗначенияЗаполнения](#), тип [Структура](#). Параметры заполнения для создаваемого объекта. Структура заполняется из параметров выбора и связей параметров выбора. В

структуру попадают только те значения, имена которых начинаются с текста **Отбор** . .

- **ПараметрыВыбора**, тип **Структура**. Значение параметра является полной копией параметра **ЗначенияЗаполнения**.
- **ДополнительныеПараметры**, тип **Структура**. Дополнительные параметры, которые прикладной разработчик считает нужным передать в форму объекта. Структура заполняется из параметров выбора и связей параметров выбора. В структуру попадают только те значения, имена которых начинаются с текста **Дополнительно** . .

Владельцем открываемой формы нового объекта будет выступать поле ввода, откуда вызвано создание объекта.

Параметры выбора (см. [здесь](#)) и связи параметров выбора (см. [здесь](#)) ограничивают список существующих элементов, которые могут быть выбраны в поле ввода. Для того чтобы создаваемый из поля ввода элемент также соответствовал наложенным ограничениям, существует специальный механизм:

- При открытии формы нового объекта, которая инициирована из поля ввода, значение параметра формы **ПараметрыВыбора** копируются в одноименное свойство управляемой формы.
- Если создание элемента выполняется в режиме выбора (свойство управляемой формы **РежимВыбора** установлено в значение **Истина**), то перед вызовом обработчика события управляемой формы **ПослеЗаписиНаСервере** выполняется проверка соответствия реквизитов созданного объекта требуемым параметрам (свойство формы **ПараметрыВыбора**). По результатам проверки устанавливается свойство управляемой формы **ВыборДоступен**.
- Перед закрытием формы (перед вызовом обработчика события **ПриЗакрытии**) вызывается обработчик события **ВыборЗначения**. По результатам вызова обработчика (значение параметра **СтандартнаяОбработка**) и свойства **ВыборДоступен**, принимается решение о дальнейших действиях:
 - Обработчик события **ВыборЗначения** не задан, или обработчик задан, а параметр **СтандартнаяОбработка** после работы обработчика установлен в значение **Истина**:
 - Значение свойства **ВыборДоступен** установлено в значение **Истина** – выполняется выбор созданного элемента;
 - Значение свойства **ВыборДоступен** установлено в значение **Ложь** – формируется сообщение о том, что созданный элемент не соответствует параметрам выбора, и выбор созданного элемента не выполняется.
 - Обработчик события **ВыборЗначения** задан и после работы обработчика параметр **СтандартнаяОбработка** установлен в значение **Ложь** – выбор созданного элемента не выполняется, сообщение не формируется. Разработчик должен самостоятельно выполнить необходимые действия: выдать диагностическое сообщение, установить в поле ввода значение по умолчанию и т. д.

Если выполняется выбор созданного элемента, то:

- Форма элемента закрывается;
- Для поля ввода, из выпадающего списка которого вызвано создание элемента, вызывается событие **ОбработкаВыбора**.

Если выбор созданного элемента не выполняется, то форма элемента закрывается, более никаких действий не выполняется.

5.6.17. Связи параметров выбора

Связи параметров выбора – данное свойство позволяет указать список реквизитов, которые будут поставлять значения, используемые при выборе значения реквизита, при открытии формы выбора, при отображении списка быстрого выбора и при выполнении ввода по строке.

В качестве примера можно привести выбор договора с контрагентом. Вначале осуществляется выбор контрагента, а затем выполняется выбор договора только из списка договоров выбранного контрагента. Причем отбор автоматически изменяется при смене контрагента.

Для того чтобы ограничить выбор, реквизит в свойстве **Связи параметров выбора** устанавливается соответствие имени реквизита, по которому будет выполняться фильтрация выбираемых значений, и реквизита, из которого будет браться значение фильтрации.

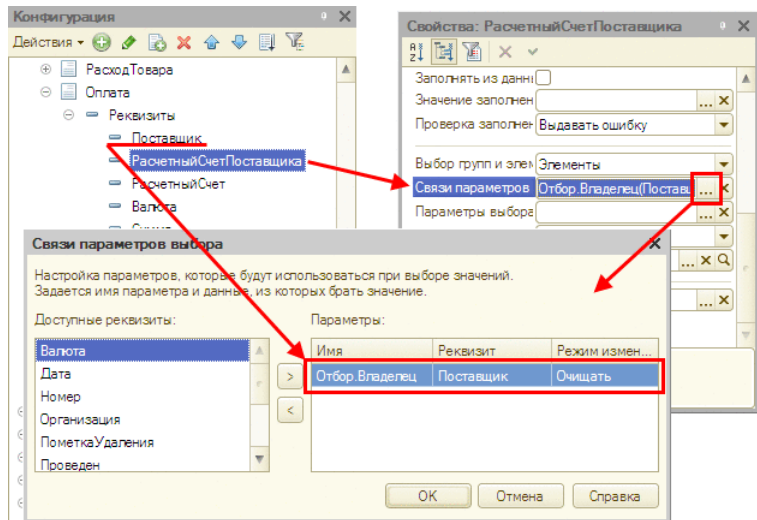


Рис. 127. Связи параметров выбора

Значения, указанные в данном свойстве, будут переданы в открываемую форму через структуру **Параметры** (см. [здесь](#)). При этом значение колонки **Имя** будет соответствовать ключу элемента структуры, а значение реквизита, указанное в колонке **Реквизит**, – значению элемента структуры. Если в колонке **Имя** указано значение вида **Отбор. Владелец**, то будет создан параметр формы **Отбор** (типа **Структура**). В этой структуре будет создан элемент с ключом **Код** и значением, полученным из реквизита, указанного в колонке **Реквизит** (в нашем примере – **Поставщик**).

Если имя у какого-либо элемента свойства **Связи параметров выбора** совпадает с именем какого-либо элемента свойства **Параметры выбора** (см. [здесь](#)), то приоритет отдается элементу из свойства **Связи параметров выбора** в том случае, если значение поля, указанного в элементе **Связи параметров выбора**, переданное в функцию **ЗначениеЗаполнено()**, возвращает **Истина**. При этом элемент из свойства **Параметры выбора** игнорируется.

Также в окне редактирования связей параметров выбора можно задать режим очистки поля при изменении полей связи. Если значение свойства **Режим изменения связанного значения** равно **Очищать**, то поле будет очищено при интерактивном изменении значения связи (изменением считается также повторный выбор значения, ранее находившегося в поле) до наступления события **ПриИзменении**. В противном случае (значение свойства равно **Не изменять**) поле не будет очищено. Очистка происходит независимо от реального изменения значения в элементе связи и выполняется до вызова обработчика события **ПриИзменении**.

Если поле отображает данные таблицы (колонка таблицы или отдельное поле, связанное с текущими данными), то очистка такого поля производится, если источником данных для таблицы формы является **ДанныеФормаКоллекция** или **ДанныеФормыСтруктураСКоллекцией**. Если реквизит, который нужно очистить, связан с табличными данными, а реквизит связи не является табличными данными, то очищаются значения во всех строках таблицы, где расположен реквизит, для которого выполнена настройка связей параметров выбора. Реквизит не может быть очищен, если он связан с колонкой реквизита типа **ДинамическийСписок**.

Для стандартного реквизита **Родитель** подчиненного справочника возможна ситуация, когда свойство **Связи параметров выбора** будет изменено системой автоматически. Это изменение происходит в следующих случаях:

- когда у справочника изменяется состояние подчиненности;

- когда у справочника изменяется значение свойства **Иерархический справочник**.

Это изменение происходит следующим образом:

- если справочник является подчиненным и иерархическим, то для реквизита **Родитель** в свойство **Связи параметров выбора** добавляется параметр **Имя: Отбор.Владелец – Реквизит: Владелец**;
- в противном случае параметр **Имя: Отбор.Владелец – Реквизит: Владелец** удаляется из списка параметров свойства **Связи параметров выбора** стандартного реквизита **Родитель**.

Рассмотрим пример: на форме есть поле **Поставщик** и поле **РасчетныйСчетПоставщика**. Для поля **РасчетныйСчетПоставщика** свойство **СвязиПараметровВыбора** установлено в значение **Объект.Поставщик**, которое будет устанавливаться в поле отбора **Владелец**.

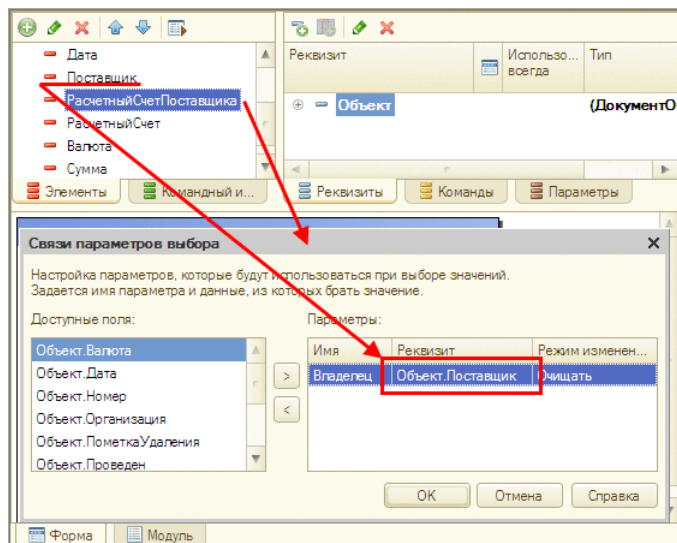


Рис. 128. Очистка связанных элементов формы

Тогда при интерактивном изменении значения поля **Поставщик** будет происходить автоматическая очистка значения поля **РасчетныйСчетПоставщика**.

5.6.18. Параметры выбора

Параметры выбора – данное свойство позволяет указать значения параметров, которые будут применяться при выборе значения реквизита. Параметры выбора будут выполняться при открытии формы выбора, при отображении списка быстрого выбора и при выполнении ввода по строке. Например, необходимо ограничить выбор только теми товарами, у которых реквизит **Вид** равен значению **Перечисление.ВидыТоваров.Услуга**. Для конкретного значения отбора можно указать список значений. Для этого при редактировании колонки **Значение** следует выбрать тип **Фиксированный массив**.

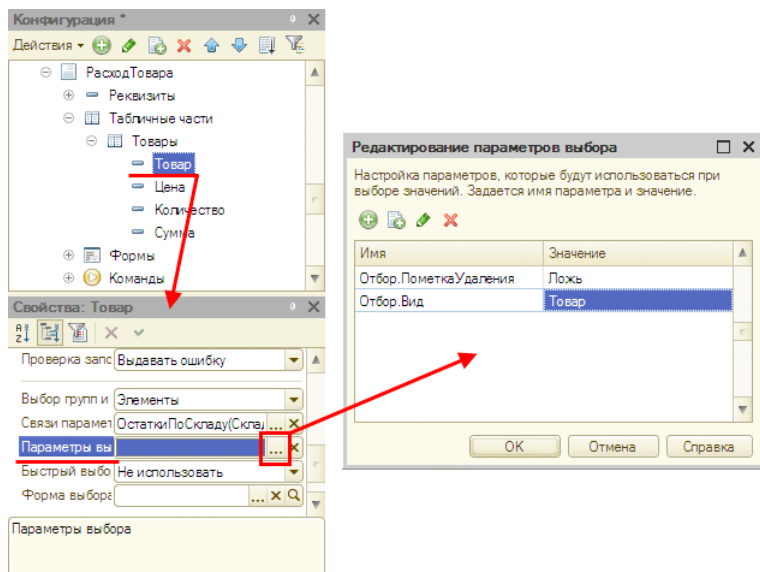


Рис. 129. Параметры выбора

Значения, указанные в данном свойстве, будут переданы в открываемую форму через структуру **Параметры**. При этом значение колонки **Имя** будет соответствовать ключу элемента структуры, а колонка **Значение** – значению элемента структуры. Если в колонке **Имя** указано значение вида **Отбор.Код**, то будет создан параметр формы **Отбор** (типа **Структура**). В этой структуре будет создан элемент с ключом **Код** и значением **Значение**.

Если имя у какого-либо параметра свойства **Связи параметров выбора** (см. [здесь](#)) совпадает с именем какого-либо параметра свойства **Параметры выбора**, то приоритет отдается элементу из свойства **Связи параметров выбора** (параметр из свойства **Параметры выбора** игнорируется). Приоритет будет отдан в том случае, если значение поля, указанного в параметре **Реквизит** свойства **Связи параметров выбора**, является заполненным (функция **ЗначениеЗаполнено()** для этого значения возвращает значение **Истина**).

ПРИМЕЧАНИЕ. Значения, указанные в свойстве реквизита **Параметры выбора**, также используются динамическим списком и системой компоновки данных.

При выборе значения субконто в ячейке таблицы, связанной с набором записей регистра бухгалтерии, автоматически устанавливается параметр выбора для поля **Владелец**, только если данный вид субконто имеет тип справочника дополнительных значений субконто. Значением параметра выбора является соответствующий вид субконто.

5.6.19. Прочее

Свойство **Использовать стандартные команды** определяет возможность включения стандартных команд (например, команды открытия списка справочника) этого объекта в командный интерфейс. Если свойство имеет значение **Ложь**, то стандартные команды не будут отображаться системой, и включить объект в интерфейс можно только с помощью команды, созданной в конфигураторе.

Свойство **Форма выбора** устанавливает форму, которая будет применяться для выбора значения реквизита. Используется для реквизитов, тип которых образован объектом

метаданных, имеющим понятие форма. Например, у справочника **Товары** есть несколько форм выбора, при этом для стандартного реквизита **Родитель** необходимо использовать особую форму. Для реализации такого поведения нужно указать эту особую форму в свойстве **Форма выбора** стандартного реквизита **Родитель** (подробнее про стандартные реквизиты см. [здесь](#)).

Формат – для типов **Число**, **Булево**, **Дата** позволяет задать формат, с помощью которого при выводе будут отображаться данные реквизита.

Формат редактирования – свойство позволяет задать формат редактирования данных типов **Число**, **Булево**, **Дата**.

Связь по типу – устанавливает связь с реквизитом, ограничивая тип вводимых значений для поля ввода. Настройка связи по типу имеет смысл для реквизитов с составным типом данных, логически связанных с другим реквизитом типа **ПланВидовХарактеристик.Ссылка**, в том числе для связи реквизита, содержащего субконто с реквизитом, который содержит значение типа **Ссылка на план счетов**. Элемент связи по типу содержит номер вида субконто для случая, если реквизит, с которым выбрана связь по типу, имеет значение типа **Ссылка на план счетов**.

Например, есть два реквизита – **ВидХарактеристики** и **Характеристика**. У реквизита **ВидХарактеристики** – тип **ПланВидовХарактеристикВидыХарактеристикТоваров.Ссылка**; у реквизита **Характеристика** – тип **Характеристика.ВидыХарактеристикТоваров**. В свойствах реквизита **Характеристика** можно установить свойство **Связь по типу** – **ВидХарактеристики**. Тогда тип выбираемого значения будет определяться типом, который задан для выбранного значения плана видов характеристик.

5.7. Константы

В системе «1С:Предприятие» константы предназначены для хранения постоянной или условно-постоянной информации. Такая информация либо совсем не изменяется в процессе деятельности предприятия, либо изменяется достаточно редко. Наиболее простой пример подобной информации – название организации, которое, как правило, не меняется.

Основная причина использования констант заключается в том, что в них один раз заносится какая-либо информация, которая затем может использоваться при формировании документов, в расчетах, при построении отчетных форм. Значение константы время от времени может редактироваться.

Рассмотрим такой пример. Наиболее часто на документах предприятия встречается подпись директора и главного бухгалтера. Естественно, что эти должностные лица должны самостоятельно ставить подписи. Но помимо самой подписи в документах требуется ее расшифровка – фамилия подписавшего документ. Конечно, можно вести фамилии директора и главного бухгалтера непосредственно в формы документов. Но если какая-либо из фамилий изменится, то придется вновь редактировать все бланки документов и исправлять фамилии. Поэтому гораздо удобнее создать в конфигурации две константы – для хранения фамилий директора и главного бухгалтера, ввести эти фамилии в константы один раз, а в многочисленных бланках документов использовать имена констант для получения их значений. При смене директора или главного бухгалтера достаточно внести изменения только в константы – ввести в них новые фамилии, и все изменения автоматически будут отражены в тех местах, где эти константы используются.

ВНИМАНИЕ! При изменении значения константы прежнее значение теряется. Для получения прежних значений данных вместо константы нужно использовать регистр сведений без измерений.

Конфигуратор системы «1С:Предприятие» позволяет создавать практически неограниченное количество констант для хранения любой нужной информации.

5.7.1. Свойства константы

Для работы с константами предназначена ветвь **Константы** дерева конфигурации.

Свойства константы редактируются в палитре свойств. Необходимо обратить внимание, что константа является типизированным объектом конфигурации.

Тип – указывается тип константы. Тип константы может быть любым из стандартных типов конфигурации (**Дата**, **Число**, **СправочникСсылка**, **ДокументСсылка** и т. д.) или составным (состоящим из набора различных типов).

В зависимости от выбранного типа палитра свойств может содержать дополнительные свойства, уточняющие выбранный тип.

Если форму редактирования константы необходимо показать в командном интерфейсе, то следует установить свойство **Использовать стандартные команды**. В этом случае команда открытия редактора константы будет отображена в тех подсистемах, к которым отнесена константа. Форма редактирования константы задается с помощью свойства **Основная форма** (подробнее см. [здесь](#)).

Формы для ввода констант создаются в ветви **Общие формы** (см. [здесь](#)).

5.8. Справочники

При заполнении бланка какого-либо документа часто требуется указывать информацию, выбирая значение из заранее заданного списка.

Возьмем в качестве примера анкету, которую требуется заполнять при поступлении на работу.

При заполнении графы **Место рождения** необходимо указать населенный пункт. Хотя общее количество населенных пунктов достаточно большое, список всех населенных пунктов все-таки ограничен. Фактически место рождения можно указать, выбрав из подобного списка нужный населенный пункт. Такой список и представляет собой справочник.

Таким образом, можно сказать, что справочник является списком возможных значений того или иного реквизита документа (в широком смысле слова «документ»).

Справочники используются в тех случаях, когда необходимо исключить неоднозначный ввод информации.

Например, для того, чтобы покупатель, продавец, кладовщик, директор понимали, о каком товаре идет речь, каждый должен называть его одинаково. И в этом случае необходим справочник. Обычно в торговом предприятии он имеет вид прайс-листа, а если такой справочник хранится в компьютере, то в него заносят всю возможную номенклатуру товаров, с которыми работает торговая фирма.

Система «1С:Предприятие» позволяет вести практически неограниченное количество необходимых справочников. Каждый справочник представляет собой список однородных экземпляров объектов: сотрудников, организаций, товаров и т. д. Каждый такой экземпляр объекта будем называть элементом справочника.

Следует иметь в виду, что в конфигурации создается структура справочника, а собственно его содержимое – элементы справочника – вводится пользователем при работе с программой. В процессе конфигурирования описывается структура информации, которая будет храниться в справочнике, разрабатывается экранное и, если необходимо, печатное представление справочника, задаются различные особенности его «поведения».

Как правило, справочники имеют стандартные реквизиты код и наименование, при этом код может иметь тип **Число** или **Строка**.

Система «1С:Предприятие» предоставляет широкие возможности по работе с кодами элементов справочника: автоматическое присвоение кодов, автоматический контроль уникальности кода и другие.

Справочник в системе «1С:Предприятие» может быть иерархическим. Существуют два вида иерархии: **иерархия групп и элементов** и **иерархия элементов**. В первом случае все данные справочника будут разделяться на два вида: «просто» элементы справочника и группы справочника. **Группы** – это логическое объединение элементов справочника. Примером иерархического справочника может служить справочник товаров, где группами являются виды товаров («Сантехника», «Бытовая химия» и т. д.), а элементами – конкретные товары («Смеситель», «Зеркало», «Стиральный порошок»).

Использование иерархических справочников позволяет организовать ввод информации в справочник с нужной степенью детализации. Элементы и группы элементов в иерархическом справочнике можно переносить из одной группы в другую.

Для справочников с видом иерархии **Иерархия элементов** группы как самостоятельный вид отсутствуют. Их роль выполняют сами элементы. Отличительной особенностью таких справочников является функциональность всех элементов. Примером справочника данного вида может служить справочник подразделений. Каждое подразделение описывается одинаковым набором реквизитов и при этом логически может содержать другое подразделение или входить в него.

Для иерархических справочников конфигуратор позволяет установить ограничение числа уровней справочников, или допускается неограниченное число уровней вложенности.

Помимо кода и наименования для справочника можно создать набор реквизитов, позволяющих хранить дополнительную информацию об элементе справочника.

Например, справочник **Контрагенты** может содержать такие сведения, как полное наименование контрагента, его ИНН, фамилии директора и главного бухгалтера и другую информацию.

Если объект предметной области, которой соответствует справочник, имеет не только такие «простые» свойства, например, полное наименование или ИНН, но и составные (списочные) свойства, справочнику может быть создан набор табличных частей.

Например, в справочнике **Контрагенты** может быть создана табличная часть для списка телефонных номеров контрагента.

Имена реквизитов справочника не должны совпадать ни с одним именем реквизита из какой-либо табличной части.

Для работы с информацией, хранящейся в справочнике, можно создать экранные формы. Могут быть созданы отдельные формы для просмотра списка элементов справочника, для редактирования элемента справочника, формы для выбора из справочника нужного элемента.

Конфигуратор позволяет создать несколько форм одного типа, например, формы для выбора из справочника нужного элемента, и использовать разные формы в разных случаях.

Рекомендуется создавать разные формы для отображения списка элементов и для выбора элементов справочника.

5.8.1. Свойства справочника

Для работы со справочниками предназначена ветвь **Справочники** дерева конфигурации.

При создании нового справочника открывается окно редактирования объекта (см. [здесь](#)).

Иерархический справочник – если свойство установлено, то справочник имеет иерархическую структуру и становится доступным свойство **Вид иерархии** и **Ограничение количества уровней иерархии**.

Вид иерархии – определяется, какой вид иерархии используется в данном справочнике. При выборе вида **Иерархия групп и элементов** для справочника определяются два вида элементов: группы и элементы. Группы предназначены только для объединения других групп и элементов справочника. Обычно для описания группы достаточно кода, наименования и родителя (ссылка на верхний уровень). Элемент справочника помимо этих реквизитов может содержать другие реквизиты, указанные на закладке **Данные**. Для справочников с этим видом иерархии можно создать формы группы и формы элемента. При выборе вида **Иерархия элементов** все элементы справочника равнозначны. Примером справочников такого вида могут служить справочники подразделений и статьи затрат.

Размещать группы сверху – свойство становится доступным, если выбрано значение **Иерархия групп и элементов**. Если свойство **Размещать группы сверху** установлено, то при отображении справочника в виде иерархического списка группы окажутся в верхних строчках списка, а элементы справочника будут располагаться ниже. Если это свойство не установлено, расположение групп и элементов будет подчиняться установленным правилам сортировки (по коду, наименованию и пр.). Например, при создании новой группы с кодом большим, чем у всех имеющихся групп и элементов (при сортировке по коду), в первом варианте эта подгруппа окажется нижней среди групп, но выше остальных элементов справочника; во втором варианте она займет самую нижнюю строчку.

Следует отметить, что установка или снятие свойства **Размещать группы сверху** не влияет на показ справочника в виде неиерархического списка.

Количество уровней иерархии – свойство становится доступным, если установлено свойство **Ограничение количества уровней иерархии**. Справочники в системе «1С:Предприятие» могут иметь более одного уровня вложенности. Если свойство **Ограничение количества уровней иерархии** не установлено, то максимальное количество уровней вложенности справочника неограниченно.

Владельцы – это свойство требует подробного объяснения.

Любой справочник может использоваться как сам по себе, так и быть подчиненным какому-либо другому справочнику или справочникам. Например, справочник договоров может использоваться отдельно, а может быть связан со справочником организаций.

Чтобы подчинить справочник другому, уже существующему в системе, в поле **Список владельцев** справочника следует нажать кнопку редактирования и в открывшемся окне выбора объекта отметить те справочники, которые являются владельцами данного справочника. Каждый такой справочник в системе «1С:Предприятие» называется владельцем, а сам справочник – подчиненным.

В отличие от многоуровневого справочника, в котором все элементы имеют одинаковую структуру, использование механизма подчиненных справочников позволяет связать элементы разной структуры. В этом случае каждый элемент подчиненного справочника будет логически связан с одним из элементов справочника-владельца.

Для справочников, которые имеют нескольких владельцев, у разных элементов могут быть владельцы разного типа, но у одного элемента может быть только один владелец.

Использование подчинения – позволяет управлять ограничением, накладываемым на владельцев. Могут использоваться только элементы, только группы или и группы, и элементы. Если у справочника несколько владельцев, то ограничение применяется ко всем владельцам.

Длина кода – свойство устанавливает максимальную длину кода элемента справочника.

Конфигуратор позволяет установить длину кода равной 0. Это может понадобиться в тех случаях, когда код элемента справочника не используется.

При назначении этого свойства желательно реально определить возможную длину кода. Однако следует иметь в виду, что в процессе эксплуатации конфигурации, если потребуются, длину кода можно увеличить.

ПРИМЕЧАНИЕ. Максимальная длина кода равна 50.

Длина наименования – в свойстве устанавливается максимально возможная длина наименования элемента справочника.

Конфигуратор позволяет установить длину наименования равной 0. Это означает, что у справочника не будет наименования.

ПРИМЕЧАНИЕ. Максимальная длина наименования равна 150.

Серии кодов – свойство позволяет установить диапазон проверки кода на уникальность и автоматическое присвоение кодов.

Если выбрана установка **Во всем справочнике**, то при автоматическом присвоении кода или при вводе кода пользователем вручную его уникальность будет проверяться среди всех элементов справочника.

Установка **В пределах подчинения** справедлива только для иерархических и подчиненных справочников. В этом случае уникальность кода будет проверяться системой только в пределах той группы и того элемента справочника-владельца, в которую вводится новый элемент справочника или редактируется уже существующий элемент.

При установке серии кодов **В пределах подчинения** нормальной является ситуация, когда находящиеся в разных группах элементы справочника имеют одинаковые коды. Это следует учитывать, если требуется переносить элементы многоуровневого справочника из одной группы в другую. При совпадении кода переносимого элемента с кодом уже существующего элемента в группе будет выдано предупреждение, и элемент не будет перенесен.

При установке серии кодов **В пределах подчинения** владельцу в иерархических подчиненных справочниках обеспечивается автоматическая нумерация и контроль уникальности кодов среди элементов с одинаковым владельцем, но различными родителями.

Тип кода – свойство позволяет выбрать тип значения для кода элемента справочника: **Число** или **Строка**. Выбор строкового типа кода бывает полезным, когда используется сложная система кодирования, и код может включать помимо цифр также буквы и символы-разделители. Наиболее характерный пример – использование в качестве кодов артикулов для швейных изделий.

Следует обратить внимание, что выбор строкового типа кода не исключает возможности автоматического присвоения таких кодов.

Для самого первого элемента система формирует код вида **001** (количество нулей зависит от установленной длины кода), то есть код представляет собой строку символов, но все символы в этой строке являются цифрами. При вводе других элементов в справочник система будет продолжать присваивать коды аналогичным образом – **002**, **003** и т. д.

Если требования к ведению справочника предполагают непременное использование смешанных буквенно-цифровых кодов, то для целей автоматической нумерации можно использовать коды вида **AA001**. Здесь первая часть кода – символы **AA** – является текстовым префиксом, а вторая часть – символы **001** – будет интерпретироваться системой

как число и использоваться при автоматическом присвоении очередного кода.

Например, если самым первым кодом в справочник введен **AA001**, то следующим автоматически присвоенным кодом будет **AA002**, затем – **AA003**, и так далее по возрастанию.

Текстовый префикс можно задать вручную (при вводе в справочник нового элемента ввести такой «составной» код) или использовать возможности установки префикса из встроенного языка системы «1С:Предприятие» (метод [УстановитьНовыйКод\(\)](#)).

Допустимая длина кода – доступно в том случае, если свойство **Тип кода** установлено в значение **Строка**. С помощью свойства можно регулировать, строка какой длины будет хранить код. Если значение свойства равно **Фиксированная**, то длина строки, содержащая код элемента справочника, всегда будет равна значению, указанному в свойстве **Длина кода**. В противном случае длина строки будет равна реальному количеству символов, формирующих код элемента.

Реквизиты. Любой новый справочник можно представить в виде таблицы, которая имеет две колонки: код элемента справочника и его наименование. Система «1С:Предприятие» помимо кода и наименования позволяет хранить дополнительную информацию об элементе справочника. При редактировании справочника можно описать набор дополнительных реквизитов, предназначенных для хранения таких дополнительных сведений.

При показе справочника на экране эти реквизиты могут представляться в виде колонок табличного поля формы списка справочника. Кроме этого, сведения, хранящиеся в реквизитах, можно использовать при формировании в различных расчетах, при формировании отчетов и так далее.

Используя механизм реквизитов справочника, легко организовать, например, картотеку сотрудников. Для этого достаточно для справочника **Сотрудники** создать реквизиты для хранения сведений об образовании, паспортных данных и прочей кадровой информации. С помощью встроенных средств поиска в справочнике при использовании конфигурации нужная информация о сотруднике может быть легко найдена.

Табличные части. Для описания некоторых данных, относящихся к справочнику и не используемых самостоятельно, используют табличные части. Примером табличной части может служить состав семьи сотрудника (данные по каждому члену семьи описываются в реквизитах табличной части, а число членов семьи может быть произвольным), послужной список сотрудника и т. д.

Если бы в приведенном примере сведения по составу семей сотрудников могли использоваться самостоятельно, то их можно было бы выделить в отдельный справочник, подчиненный справочнику **Сотрудники**.

Основное отличие табличной части от подчиненного справочника в том, что на элементы справочника можно ссылаться, а на строки табличной части – нет. При обращении к элементу справочника он весь, вместе со всеми табличными частями, считывается из базы данных в память. Если табличная часть содержит достаточно большое количество строк, это может ухудшить производительность системы. Поэтому табличную часть стоит использовать, если не нужно хранить ссылки на элементы и количество элементов ограничено. Каждый справочник может иметь неограниченное число табличных частей.

Автоумерация. Установка свойства приводит к тому, что вновь введенному элементу в справочнике код будет присваиваться при записи. Автоматически присвоенный код можно исправить.

Контроль уникальности. Если код используется для однозначной идентификации конкретного элемента в справочнике, он должен быть уникальным (не должен повторяться). Если свойство **Контроль уникальности** установлено, проверка кода на уникальность будет проводиться автоматически при вводе в справочник нового элемента.

Основное представление – в свойстве задается представление элементов справочника. Например, значения типа **СправочникСсылка**, введенные в реквизит документа, справочника или константу, будут представляться в виде кода или наименования элемента справочника в зависимости от данного свойства. Для форм списков данная колонка становится колонкой по умолчанию. При открытии списка эта колонка становится активной.

Ввод на основании. На закладке **Ввод на основании** указывается, какие объекты конфигурации могут являться основанием для объектов данного типа и для каких объектов объекты данного типа могут являться основанием. По кнопке **Конструктор ввода на основании** запускается конструктор создания процедуры ввода на основании. Подробнее о работе с конструктором см. [здесь](#).

Примером ввода на основании может служить ввод документа **Передача на реализацию** на основании элемента справочника **Товары**.

5.8.2. Свойства реквизитов справочника

Помимо основных свойств реквизиты справочника имеют следующие свойства:

- **Тип** – указывается тип данных реквизита.
- **Использование** – для иерархических справочников определяет использование реквизита для групп и элементов.

5.8.3. Предопределенные элементы справочника

Разработчик конфигурации может создать для справочников набор предопределенных элементов и групп элементов (для иерархических справочников). Эти элементы не могут быть удалены пользователями в режиме 1С:Предприятие.

Форма для ввода предопределенных элементов открывается нажатием кнопки **Предопределенные** на закладке **Прочие** окна редактирования объекта конфигурации. В конфигураторе вводятся только основные свойства элемента (имя, код и наименование). Имя элемента можно использовать в выражениях встроенного языка. Значения других реквизитов предопределенного элемента вводятся в режиме 1С:Предприятие.

Визуально предопределенные элементы справочников в режиме 1С:Предприятие отличаются от элементов, созданных пользователями, видом пиктограммы.

ВНИМАНИЕ! Для справочника, имеющего владельца, нельзя создать предопределенные элементы. И наоборот, справочнику, имеющему предопределенные элементы, нельзя назначить владельца.

5.9. Документы

Документ – одно из основных понятий системы «1С:Предприятие». При помощи документов организуется ввод в систему информации о совершаемых хозяйственных операциях.

В большинстве своем документы, которые создаются в процессе настройки конфигурации задачи, являются электронными аналогами стандартных бумажных документов, однако использование этого типа данных может выходить далеко за рамки простой фиксации информации о хозяйственных операциях.

Каждый документ содержит информацию о конкретной хозяйственной операции и характеризуется своим номером, датой и временем. Дата и время – наиболее важные характеристики документов, так как позволяют устанавливать строгую временную последовательность совершения операций.

В конфигурации описывается только структура документа, конкретные экземпляры документов вводятся при работе с программой пользователем. Например, созданный в конфигурации документ **Накладная** при работе с системой «1С:Предприятие» позволит формировать накладные, которые будут иметь разное содержание, но одинаковый набор реквизитов, одинаковую логику поведения и так далее.

Далее для простоты вместо термина «структура документа» будет использоваться термин «документ», подразумевая под этим средства для ввода и визуализации документа.

Конфигуратор позволяет описать структуру документа, организовать формы для ввода информации в документ и описать алгоритм построения печатных форм документа.

Помимо даты, времени и номера документа можно создать набор реквизитов, позволяющих хранить дополнительную информацию.

Если объект предметной области, которой соответствует документ, имеет не только такие «простые» свойства, как, например, дату, номер и итоговую сумму, но и составные (списочные) свойства, документу может быть создан набор табличных частей.

Например, в документе **Расходная накладная** может быть создана табличная часть для списка продаваемой номенклатуры.

Для работы с документами в конфигурации может быть создано необходимое количество списков документов одного вида и журналов документов разного вида. Форма списка отличается от формы журнала в первую очередь тем, что в списке отсутствует колонка **Вид документа** (т. к. список содержит документы одного вида), а журнал обычно содержит эту колонку.

При создании документа можно указать перечень журналов, в которых будет осуществляться работа с документами этого вида. Для документов разных видов можно

указывать один журнал, что позволяет произвольным образом группировать документы в журналах. Назначенные документу журналы можно менять.

Документы могут изменять состояние регистров учета (проводиться). Если документ проведен, то данные, которые указал пользователь при вводе документа, отразились в учетных регистрах системы – изменились остатки товаров на складах, изменилась задолженность перед контрагентом и т. д. Документы могут проводиться в реальном времени (оперативное проведение) и прошлой датой (неоперативное проведение).

5.9.1. Свойства документа

Для работы с документами предназначена ветвь **Документы** дерева конфигурации. На этой же ветви располагаются служебные объекты конфигурации – нумераторы и последовательности.

В этом разделе будут описаны специфические свойства документа в дополнение к общим свойствам объектов конфигурации, описание которых см. [здесь](#).

Свойства документа редактируются в палитре свойств или окне редактирования (см. [здесь](#)). Данные по документу в окне редактора распределены по закладкам.

На закладке **Основное** размещены основные данные по документу.

На закладке **Данные** вводятся реквизиты и табличные части.

В палитре свойств каждому реквизиту помимо основных свойств указывают тип. Если требуется обеспечить быстрый поиск или отбор нужной информации по реквизитам документа, для реквизитов нужно установить свойство **Индексировать** (подробнее см. [здесь](#)).

На закладке **Нумерация** объединены данные, которые используются для назначения правил нумерации документов.

У любого документа существуют обязательные реквизиты, которые создаются автоматически и которые удалить нельзя, – это дата, время и номер документа. В отличие от даты и времени, для номера документа можно задать несколько параметров, которые будут управлять поведением этого реквизита при работе с документами создаваемого вида. Совокупность этих параметров будет определять правила нумерации документов при работе системы «1С:Предприятие».

Автономная нумерация – установка свойства приводит к тому, что вновь введенному документу номер будет присваиваться автоматически. Автоматически присвоенный номер можно исправить.

Нумератор – документу может быть назначен нумератор из числа уже существующих в конфигурации. Для назначения документу нумератора из числа существующих в конфигурации следует выбрать имя нумератора в свойстве **Нумератор**. В этом случае прочие элементы группы **Нумерация документа** окна, за исключением свойства **Автономная нумерация**, становятся недоступными, то есть правила нумерации документов данного вида будут полностью определяться назначенным нумератором. О создании и свойствах нумераторов см. [здесь](#).

Использование нумераторов позволяет организовать сквозную нумерацию документов разных видов. Для этого всем документам, для которых требуется иметь сквозную нумерацию, должен быть назначен один нумератор. Контроль уникальности и присвоение очередного номера будут выполняться с учетом всех документов, для которых назначен этот нумератор.

Длина номера – устанавливает максимальную длину номера документа.

Тип номера – свойство аналогично свойству **Тип кода** справочника (см. [здесь](#)).

Допустимая длина номера – доступно в том случае, если свойство **Тип номера** установлено в значение **Строка**. С помощью свойства можно регулировать, строка какой длины будет хранить номер. Если значение свойства равно **Фиксированная**, то длина строки, содержащая номер документа, всегда будет равна значению, указанному в свойстве **Длина номера**. В противном случае длина строки будет равна реальному количеству символов, формирующих номер документа.

Контроль уникальности – если это свойство установлено, то при вводе нового документа его номер проверяется на уникальность в пределах, установленных в свойстве **Периодичность**.

Периодичность – свойство устанавливает пределы контроля уникальности номеров документов и период повторяемости номеров. Если свойство **Контроль уникальности** номеров документов установлено, то в свойстве **Периодичность** указывается, в каких пределах будет осуществляться этот контроль.

При установленном свойстве **Автономная нумерация** система «1С:Предприятие» будет присваивать очередной порядковый номер каждому новому документу. После завершения периода, установленного в свойстве **Периодичность**, нумерация документов начнется с 1.

На закладке **Движения** производится настройка поведения при проведении и отмене проведения документа.

Проведение – свойство устанавливает, разрешено ли проведение документа при записи. Если выбрано **Разрешить**, то документ осуществляет движения регистров (изменяет их состояние). Другим следствием такого выбора будет автоматический вызов обработчика события **ОбработкаПроведения** проведения документа в режиме 1С:Предприятие, возникающего при нажатии кнопки формы документа, для которой при конфигурировании выбрано предопределенное действие **Записать и закрыть** (обычно кнопка **OK**). Для формирования обработчика события **ОбработкаПроведения** можно воспользоваться конструктором движения регистров (см. [здесь](#)), кнопка вызова которого расположена на закладке **Движения** окна редактирования объекта **Документ**.

Оперативное проведение – свойство устанавливает, разрешено ли оперативное проведение документа. Для документов с разрешенным оперативным проведением при выборе не текущей даты проведение осуществляется в неоперативном режиме, т. е. учитывается уже свершившийся факт, который не требует контроля, осуществляемого в оперативном режиме (например, проверка остатка, указанного в расходной накладной товара).

Если для документа разрешено **Оперативное проведение**, то при создании нового документа время документа «нулевое». При проведении система получает оперативную отметку времени, которая может совпадать с текущей датой и текущим временем, а может быть и больше, чем текущее время, и присваивает ее документу, после чего осуществляется оперативное проведение. Более подробно об оперативной отметке времени см. [здесь](#).

В случае редактирования документа с установленным свойством **Оперативное проведение** время уже отличается от текущего, поэтому при проведении (указана текущая дата) документа на экран выводится запрос о виде проведения. Если на вопрос о виде проведения выбрать **Оперативное проведение**, то для документа устанавливается текущее время. Если при редактировании изменить время (например, указать будущее время текущего дня), то система также установит текущее время. Если на вопрос о виде проведения выбрать **Неоперативное проведение**, то, когда формат даты не предусматривает ввода времени, в документе указывается время начала дня.

Удаление движений – свойство доступно, если в свойстве **Проведение** выбрано **Разрешить**. Оно задает режим удаления всех записей при перепроведении или отмене проведения документа, которые документ записал в процессе проведения:

- Режим **Удалять автоматически** означает, что удаление производится системой при выполнении повторного проведения уже проведенного документа (перед записью новых движений) и при отмене проведения документа.
- Режим **Не удалять автоматически** означает, что удаление движений выполняется программно в специальных случаях. Этот режим используется, если нужно управлять удалением и при проведении, и при отмене проведения.
- Режим **Удалять автоматически при отмене проведения** означает, что система будет автоматически удалять движения только при отмене проведения. При перепроведении движения не будут удаляться автоматически. Этот режим устанавливается по умолчанию.

Если свойство **Удаление движений** у документа установлено в значение **Не удалять автоматически** или **Не удалять автоматически при отмене проведения**, то для избежание дублирования информации, следует выполнять очистку наборов записей коллекции **Движения** в начале операции проведения.

Запись движений при проведении – свойство определяет поведение системы при создании движений во время проведения документа:

- Режим **Записывать выбранные** (режим по умолчанию) означает, что набор записей коллекции движения будет записан только в том случае, если свойство набора **Записывать** установлено в значение **Истина** (вне зависимости от того, были модифицированы записи набора или нет).
- Режим **Записывать модифицированные** (устанавливается при конвертации из версий «1С:Предприятие 8.1» и более ранних) означает, что записаны будут те наборы записей, которые были модифицированы (для них система автоматически установит свойство **Записывать** в значение **Истина**).

По окончании записи документа система ставит всем наборам записей, регистрирующим движения документа, свойство **Записывать** в исходное состояние, даже если запись завершилась неудачно.

Заполнение последовательности – выбирается режим автоматического заполнения последовательностей. В окне редактирования на закладке **Последовательности** определяется вхождение документа в последовательности.

В окне редактирования документа на закладке **Журналы** можно отметить те журналы документов, в которых при работе пользователя с системой «1С:Предприятие» будут отражаться документы данного вида. Необходимые формы журналов документов могут быть созданы потом.

На закладке **Ввод на основании** размещено два списка объектов конфигурации. В верхнем списке необходимо указать объекты, на основании которых может вводиться документ, а в нижнем – для которых данный документ является основанием при вводе нового объекта.

ВНИМАНИЕ! Документ может вводиться как на основании другого документа, так и на основании объектов другого вида (элементов справочника, плана видов характеристик, плана счетов и плана видов расчета). И наоборот, документ может являться основанием для ввода не только другого документа, но и объектов другого вида.

Для создания процедуры, осуществляющей подготовку данных создаваемого объекта на основании образца, следует воспользоваться специальным конструктором, описание которого см. [здесь](#).

На закладке **Права** имеется возможность установки привилегированного режима при проведении (свойство **Привилегированный режим при проведении**) и/или при отмене проведения документа (свойство **Привилегированный режим при отмене проведения**):

- Если свойство установлено, то соответствующее действие (проведение или отмена проведения) система будет выполнять всегда в привилегированном режиме (при вызове проведения или отмены проведения на стороне сервера и в файловом варианте). Однако привилегированный режим не будет установлен, если проведение (или отмена проведения) документа выполняется в клиент-серверном варианте на стороне толстого клиента. В привилегированном режиме выполняется:
 - Автоматическое удаление движений.
 - Автоматическая запись движений.
 - Соответствующий обработчик (**ОбработкаПроведения** или **ОбработкаУдаленияПроведения**). Однако запись объекта выполняется в обычном (непривилегированном) режиме.
- Привилегированный режим включается системой после выполнения записи объекта перед началом проведения (перед удалением движений, если они удаляются автоматически). Аналогично и при отмене проведения.
- При создании новых документов свойства имеют значение **Истина**, если в свойствах конфигурации указан основной режим запуска – управляемое приложение, и **Ложь**, если основным режимом запуска указан обычный.

5.9.2. Механизм проведения документов

Информация, отражающая хозяйственную деятельность предприятия, хранится в регистрах (см. [здесь](#)). Документы могут изменять состояние регистров. Этот процесс называется **проведением**. Данный механизм является рекомендуемым механизмом изменения состояния регистров. Проведение может выполняться в **оперативном** или **неоперативном** режиме (свойство **Оперативное проведение**).

ПРИМЕЧАНИЕ. Как правило, механизм оперативного проведения используется для решения задач оперативного учета.

5.9.2.1. Оперативное и неоперативное проведение

Механизм оперативного проведения предназначен для того, чтобы разделить случаи, когда документ проводится в реальном времени, и случаи, когда проведение документа отражает уже свершившийся факт.

Проведение в реальном времени необходимо тогда, когда ввод и проведение документа не просто фиксируют в системе произошедшее событие, а участвуют в его формировании, помогая оператору правильно ввести информацию. Разумеется, это имеет смысл только в тот момент, когда данное событие происходит в реальной жизни.

Классическим примером является ввод и проведение документа, отражающего продажу товаров со склада. При вводе такого документа в задачу оператора входит не только правильный ввод списка товаров, которые приобретает покупатель, но и выполнение различных проверок. Прежде всего, необходимо проверить, что запрашиваемый товар имеется на указанном складе. При этом очень важно, чтобы проверка учитывала тот факт, что одновременно с этим оператором работают и другие операторы, которые могут одновременно выписывать те же самые товары. Соответственно, задачей системы является не допустить продажу одного и того же товара двум покупателям. Кроме того, может потребоваться и проверка доступного покупателю размера кредита или наличие факта оплаты счета, а также другие самые разнообразные проверки.

В то же время если документ вводится задним числом, то есть в момент его ввода известно, что такое событие уже произошло в жизни предприятия, например, конкретный товар уже отгружен клиенту, необходимость в таких проверках отпадает, и нужно просто отразить в учете произошедшее событие. В этом случае проведение документа только фиксирует событие, а не участвует в его формировании.

Таким образом, задача механизма оперативного проведения заключается в разделении этих двух вариантов проведения и с точки зрения пользователя, чтобы он понимал, какой вид проведения выполняется, и с точки зрения алгоритма проведения документа, чтобы тот отработал действия, соответствующие текущему варианту проведения.

Следовательно, оперативность или неоперативность проведения документа определяется по его дате. Если дата проводимого документа совпадает с текущей датой, то система будет проводить такой документ в оперативном режиме, не задавая вопросов, и в обработке проведения об этом можно узнать, чтобы выстроить определенный алгоритм проведения документа.

5.9.2.2. Расширение формы и проведение

Кроме свойства документа **Оперативное проведение** существует возможность задания режима проведения у расширения формы документа. Свойство **Использовать режим проведения** может принимать следующие значения:

- **Неоперативный** – документ всегда будет проводиться в неоперативном режиме. Если отсутствуют права на неоперативное проведение, будет выдано исключение.
- **Оперативный** – документ всегда будет проводиться в оперативном режиме. Если отсутствуют права на неоперативное проведение, то будет выдано исключение при попытке провести документ прошлого периода.
- **Запрашивать** – система всегда будет запрашивать текущий режим проведения.
- **Автоматически** – в этом случае система работает по следующему алгоритму:
 - если дата документа меньше текущей, выполняется неоперативное проведение;
 - если дата документа равна текущей, выполняется оперативное проведение;
 - если дата документа больше текущей, выдается исключение;
 - если невозможно провести документ в выбранном режиме (не хватает прав доступа и т. д.), также будет выдано исключение;
 - если на стороне клиента режим проведения неизвестен, то в параметр **РежимПроведения** обработчика события **ПередЗаписью** будет передано значение **Неопределено**.

Одной из задач оперативного проведения документов является размещение документов в хронологическом порядке на шкале времени. Такое размещение необходимо, в частности, для того, чтобы корректно списывались остатки по регистрам остатков (см. [здесь](#)) оперативного учета. Для размещения документов используются такие понятия, как **момент времени** и **оперативная отметка времени**. Рассмотрим эти понятия более подробно.

Для определения положения документа на оси времени используется реквизит документа **Дата**, который содержит время с точностью до секунды. Это позволяет контролировать последовательность записи документов. Однако при большом объеме создаваемых документов вероятна ситуация, когда несколько документов будут иметь одинаковое значение даты (т. е. будут созданы в течение одной секунды). Как в этом случае определить последовательность созданных документов?

5.9.2.3. Момент времени

Для обработки подобных ситуаций существует понятие **момент времени**. Момент времени представляет собой совокупность даты, времени и ссылки на объект базы данных.

Он позволяет однозначно идентифицировать любой объект ссылочного типа базы данных на оси событий, но имеет смысл в основном только для документов. Кроме того, момент времени позволяет идентифицировать и необъектные данные, например, записи регистров, подчиненных регистратору.

Понятие момента времени реализовано во встроенном языке при помощи универсального объекта [МоментВремени](#).

Для нескольких документов, имеющих одинаковую дату и время, последовательность их на оси событий определяется системой исходя из ссылок на эти документы. Она может не совпадать с последовательностью создания документов, и она недоступна для изменения пользователем, то есть нельзя каким-либо образом повлиять на последовательность документов внутри одной секунды или «вычислить», что один документ создан раньше, а другой – позже.

5.9.2.4. Оперативная отметка времени

Оперативная отметка времени – это значение типа *Дата*. Оперативная отметка времени – это «основа», которая собственно и позволяет выполнять оперативное проведение документов. Оперативная отметка времени создается системой каждый раз при оперативном проведении документа. Ее значение формируется исходя из текущей даты сеанса и последней созданной оперативной отметки.

5.9.2.5. Поясное время

При работе системы в различных часовых поясах необходимо учитывать эту особенность при получении оперативной отметки времени. Например, когда в единой информационной базе, физически расположенной в одном городе (часовом поясе), ведется учет нескольких удаленных предприятий (например, филиалы холдинга), которые расположены в других городах (и других часовых поясах). В этом случае необходимо, чтобы для каждого филиала выдавалась своя отметка времени.

Для учета часовых поясов существует понятие часового пояса информационной базы и часового пояса сеанса.

Часовой пояс информационной базы определяет часовой пояс, который по умолчанию будет установлен для нового сеанса. При создании информационной базы часовой пояс информационной базы не определен. Однако может быть установлен с помощью метода глобального контекста [УстановитьЧасовойПоясИнформационнойБазы\(\)](#). Информация о часовом поясе информационной базы сохраняется в базе данных и не меняется при операциях загрузки/выгрузки информационной базы. При создании начального образа информационной базы (с помощью механизмов распределенной информационной базы, см. [здесь](#)) в создаваемый образ переносится часовой пояс информационной базы, из которой создается образ.

Если часовой пояс информационной базы не задан, то используется часовой пояс компьютера, на котором установлен сервер «1С:Предприятия» (в клиент-серверном варианте), или часовой пояс локального компьютера (в файловом варианте).

Часовой пояс сеанса описывает тот часовой пояс, в котором работает конкретный сеанс. По умолчанию часовой пояс сеанса равен часовому поясу информационной базы.

Часовой пояс сеанса может быть установлен с помощью метода глобального контекста [УстановитьЧасовойПоясСеанса\(\)](#). Часовой пояс сеанса сохраняется до конца сеанса. Используется для определения текущей даты сеанса и получения **оперативной отметки времени**.

5.9.2.6. Получение оперативной отметки времени

В ходе оперативного проведения система изменяет время документа таким образом, чтобы очередной документ, проводимый в оперативном режиме, имел бы момент времени более поздний, чем у предыдущего оперативно проведенного документа. Для этого используется понятие **оперативной отметки времени**. Оперативная отметка получается системой автоматически при оперативном проведении, но может быть получена во встроенном языке в явном виде с помощью метода [ПолучитьОперативнуюОтметкуВремени\(\)](#) на основании текущей даты сеанса.

Текущая дата сеанса равна дате компьютера, приведенной к часовому поясу сеанса. Под приведением понимается пересчет местного времени компьютера в поясное время, заданное часовым поясом сеанса. Пересчет выполняется через универсальное координированное время (UTC).

Все пользователи обращаются к единому механизму оперативной отметки времени, а механизм выдает каждому пользователю очередную отметку. Механизм получения оперативной отметки времени обеспечивает получение даты, большей, чем предыдущая отметка, полученная этим или другим пользователем в данном часовом поясе. В качестве оперативной отметки система, как правило, возвращает текущее время сеанса. Однако если текущее время больше или равно последней выданной какому-либо пользователю отметке, то возвращается значение на секунду большее, чем значение последней выданной отметки. Таким образом, обеспечивается получение при каждом обращении значения, по возможности соответствующего текущему времени, но в обязательном порядке большего, чем предыдущее полученное значение.

Следует помнить, что различные сеансы с одинаковым часовым поясом будут использовать одно и то же время для получения оперативной отметки времени. Таким образом, будет существовать столько несвязанных оперативных отметок времени, сколько уникальных часовых поясов будет установлено в качестве часовых поясов сеансов.

5.9.3. Нумераторы

Нумератор представляет собой объект конфигурации, описывающий правила нумерации документов: тип и длину номера документа, его периодичность, необходимость контроля уникальности.

Основное назначение нумератора – обеспечить возможность сквозной нумерации документов разного вида. Для этого таким документам назначается одинаковый нумератор.

5.9.3.1. Управление списком нумераторов

Для работы с объектами конфигурации типа [Нумератор](#) предназначена ветвь дерева конфигурации, которая расположена «внутри» ветви [Документы](#) и начинается у ключевого слова [Нумераторы](#).

5.9.3.2. Свойства нумератора

В этом разделе будут описаны специфические свойства нумераторов в дополнение к общим свойствам объектов конфигурации, описание которых см. [здесь](#).

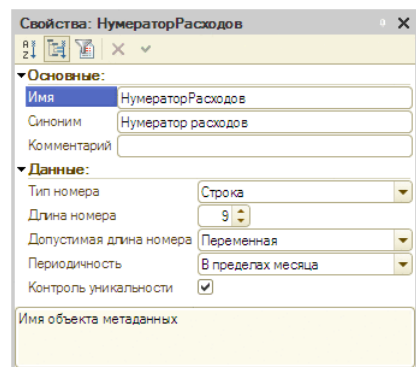


Рис. 130. Свойства нумератора

Тип номера – выбирается тип значения для номера документа – числовой или текстовый. Выбор текстового типа номера бывает полезным, когда используется сложная система нумерации документов, и номер документа может включать, помимо цифр, также буквы и символы-разделители.

Длина номера – устанавливает максимальную длину номера документа.

Периодичность – свойство устанавливает две важные характеристики нумератора: пределы контроля уникальности номеров документов и период повторяемости номеров.

Если свойство **Контроль уникальности** номеров документов установлено, свойство **Периодичность** устанавливает, в каких пределах осуществлять этот контроль.

Например, если установлена периодичность **В пределах дня**, то уникальность номеров документов будет контролироваться в пределах суток: на следующие сутки номера документов могут повторяться, но в пределах суток они будут уникальны.

При установленном свойстве [Автонумерация](#) (см. [здесь](#)) система «1С:Предприятие» будет присваивать очередной порядковый номер каждому новому документу. После завершения периода, установленного в свойстве [Периодичность](#), нумерация документов начнется с 1.

Контроль уникальности – если это свойство установлено, то при вводе нового документа его номер проверяется на уникальность в пределах, установленных в свойстве [Периодичность](#).

5.9.4. Последовательности документов

Последовательности документов являются вспомогательными объектами конфигурации. Они предназначены для обеспечения проведения определенных документов в строгой хронологической последовательности.

Все документы в системе «1С:Предприятие» образуют единую хронологическую последовательность. Для этого каждый документ имеет дату и время. Даже если два документа имеют одинаковую дату и одинаковое время, они все равно располагаются в определенной последовательности, определяемой моментом их ввода в систему. Дата и время документа могут быть изменены. Таким образом, независимо от порядка ввода документов они могут быть расположены в последовательности, отражающей реальную последовательность происшедших в хозяйственной жизни предприятия событий, которые данные документы отражают.

В системе «1С:Предприятие» в процессе проведения документ выполняет некоторые действия, которые отражаются данным документом в различных механизмах учета, поддерживаемых «1С:Предприятием».

Алгоритм проведения документа, как правило, отражает в учете данные, записанные в самом документе (в его реквизитах и табличных частях). Однако в некоторых случаях алгоритм проведения документа анализирует также и текущие итоги, используя их при проведении. Например, если документ списывает товары или материалы по средней себестоимости, то для определения суммы списания алгоритм проведения будет анализировать остатки товаров (материалов) на момент документа. Если списание выполняется по методам LIFO или FIFO, то алгоритм проведения будет анализировать существующие остатки товаров (материалов) в разрезе партий на момент позиции документа, определяемой датой и временем проведения документа.

Очевидно, что документы, основывающиеся при проведении на данные итогов, должны проводиться строго последовательно. Однако на практике из-за ошибок при вводе информации и несвоевременного поступления документов часто приходится вводить или исправлять документы задним числом. Разумеется, в этом случае движения регистров, сформированные всеми последующими документами (расположенными после того, который был исправлен), становятся некорректными. Например, если выяснилось, что в одной из приходных накладных, введенных в начале месяца, было неверно указано количество товара, то во всех последующих расходных накладных, списывающих имеющиеся в наличии партии, необходимо заново проанализировать остатки с учетом внесенных изменений и заново записать движения регистров. То есть все документы, анализирующие остатки и расположенные после измененного документа, должны быть перепроведены.

Для автоматического контроля необходимости перепроведения документов используются объекты ветви [Последовательности](#). Каждый введенный в конфигурации объект [Последовательности](#) обеспечивает контроль за порядком проведения документов указанных видов. Таким образом, в системе может существовать несколько независимых последовательностей.

5.9.4.1. Управление списком последовательностей

Работа по созданию объектов конфигурации типа [Последовательность](#) ведется в окне [Конфигурация](#). Для последовательности отведена отдельная ветвь дерева конфигурации, которая расположена «внутри» ветви [Документы](#) и начинается у ключевого слова [Последовательности](#).

5.9.4.2. Свойства последовательности документов

В этом разделе будут описаны специфические свойства последовательностей документов в дополнение к общим свойствам объектов конфигурации, описание которых см. [здесь](#).

Свойства последовательности редактируются в окне редактирования [Последовательность](#).

На закладке [Использование](#) выбираются документы, которые относятся к данной последовательности, и движения, влияющие на последовательность.

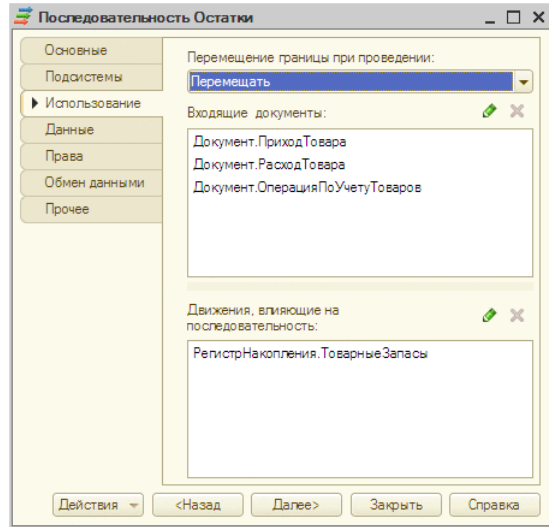


Рис. 131. Свойства последовательности

Перемещение границы при проведении. Если для свойства указано значение [Перемещать](#), то документ, зарегистрированный в этой последовательности, при своем проведении будет пытаться переместить границу этой последовательности документов. Если для свойства указано значение [Не перемещать](#), то документ не будет перемещать границу этой последовательности документов при своем проведении.

Документы, входящие в последовательность – в верхнем списке окна [Последовательность](#) указываются виды документов, которые относятся к данной последовательности.

В качестве документов, на проведение которых будет влиять данная последовательность, следует выбрать те виды документов, которые при проведении будут анализировать состояние различных регистров. Например, такими документами могут быть расходные накладные, накладные на передачу, на реализацию и т. д.

Движения, влияющие на последовательность – одно из основных свойств последовательности. Оно определяет, какие из движений будут влиять на необходимость перепроведения документов данной последовательности, то есть движения или итоги каких механизмов учета используются документами данной последовательности при проведении. Например, в качестве таких движений могут выступать движения регистров.

Для настройки данного параметра следует добавить в список те виды регистров, движения которых будут нарушать данную последовательность.

Измерения. Последовательности могут иметь подчиненные объекты, называемые измерениями, которые создаются на закладке [Данные](#) окна редактирования.

Если для последовательности не создано ни одного измерения, то при восстановлении данной последовательности будут перепроводиться все входящие документы. Если требуется, чтобы данная последовательность учитывала не все, а вполне определенные ситуации, то в последовательность включают измерение. В этом случае перепроводить нужно будет только те документы, которые изменяют состояние регистра с учетом свойств измерения.

Если изменяется состояние регистров, участвующих в последовательности, то неактуальными становятся более поздние документы с теми же значениями в реквизитах (перечислены в свойстве измерения [Соответствие реквизитам документов](#)), которые содержатся в реквизитах удаленных (добавленных) записей регистров (перечислены в свойстве измерения [Соответствие реквизитам движений](#)).

Например, последовательность учитывает изменение состояния регистров по документам [Приходная накладная](#) и [Расходная накладная](#). Если требуется учитывать

дополнительные критерии необходимости перепроведения указанных документов (например, нужно перепроводить документы по определенному значению номенклатуры), то в последовательность следует добавить измерение. В палитре свойств измерения указать его тип ([СправочникСсылка.Номенклатура](#)) и установить связь с реквизитами регистров.

Рис. 132. Свойства измерения последовательности

В соответствии с выбранным типом измерения в списки документов и регистров измерения для выбора включаются только те объекты, в состав которых входят указанные типы измерения.

Введение измерения позволяет сократить время перепроведения документов, что особенно важно при значительном числе документов, т. к. будут перепроводиться только те документы, которые содержат данные по указанному значению измерения.

5.9.4.3. Работа с последовательностями документов

При работе в режиме 1С:Предприятие для каждой введенной в конфигурации последовательности документов система будет автоматически поддерживать границу последовательности. В качестве границы последовательности будет выступать позиция документа. При последовательном проведении документов, входящих в данную последовательность, граница последовательности будет устанавливаться на каждый вновь проведенный документ. Однако если будет проводиться документ, относящийся к данной последовательности, но расположенный позже другого проведенного документа, относящегося к той же последовательности и находящегося после текущей границы последовательности, то граница последовательности сдвигаться не будет, так как нарушается последовательность проведения документов. Эта ситуация может быть проанализирована алгоритмом проведения документа.

При проведении документов задним числом, отмене проведения или удалении документов, если удаляются или записываются движения регистров, указанные как влияющие на данную последовательность, граница последовательности отодвигается на момент измененного документа. Перед перемещением границы назад производится проверка на наличие границ, которые необходимо переместить назад. Эта проверка производится без эксклюзивной блокировки границы.

Таким образом, граница последовательности будет продвигаться вперед при последовательном проведении документов, относящихся к данной последовательности, и будет отодвигаться назад при изменении задним числом движений регистров, относящихся к данной последовательности.

В режиме перепроведения документов (вызывается выбором пункта **Все функции – Стандартные – Проведение документов**) существует специальная возможность восстановления последовательности проведения документов. При ее использовании система автоматически выполняет перепроведение всех документов, относящихся к данной последовательности, от границы последовательности до указанного момента.

В приведенном нами примере с учетом товаров проводимые расходные накладные будут сдвигать границу последовательности вперед. Любое изменение в движениях по регистру, на котором ведется стоимостный учет товаров, записанное раньше границы последовательности, будет отодвигать границу последовательности назад, на момент этого документа. После этого проводимые документы, находящиеся позже границы последовательности, уже не будут двигать ее вперед, если между границей последовательности и проводимым документом окажутся проведенные документы из этой последовательности. Режим восстановления последовательности будет перепроводить все расходные накладные, хотя и влияют своими движениями на границу последовательности, перепроводиться не будут, так как они не используют в алгоритме проведения остатков и не включены в список документов, относящихся к данной последовательности. После выполнения восстановления последовательности проводимые после границы последовательности документы снова будут двигать границу вперед.

Режим восстановления последовательности позволяет автоматически выполнить перепроведение всех документов, относящихся к последовательности, от текущей позиции границы последовательности до указанного момента. В верхней части диалога следует выбрать позицию, до которой будет выполняться перепроведение документа.

5.9.5. Ввод документов на основании

Одним из режимов ввода новых документов в процессе работы пользователя с системой «1С:Предприятие» является режим ввода на основании. С точки зрения пользователя режим ввода на основании позволяет вводить документы или элементы справочников, заполняя их реквизиты путем копирования информации из другого объекта информационной базы – документа или объекта другого вида.

На закладке **Ввод на основании** можно выбрать объекты, которые могут являться основанием для выбранного вида документа (поле **Вводится на основании**), и те объекты, которые могут быть введены на основании данного вида документов (поле **Является основанием для**).

Для реализации механизма ввода на основании необходимо реализовать в модуле документа обработчик события **ОбработкаЗаполнения**. Описание работы механизма заполнения см. [здесь](#).

Текст обработчика может быть доработан специалистом, осуществляющим конфигурирование системы. В тексте обработчика следует предусмотреть выполнение тех или иных операций по переносу информации в зависимости от вида документа-образца, а также любые другие необходимые действия.

Конструктор ввода на основании (см. [здесь](#)) предназначен для облегчения создания этого обработчика.

5.10. Журналы документов

В системе «1С:Предприятие» журналы документов являются объектами, позволяющими осуществлять работу с документами разных видов. Работая с формами журнала, пользователь может вводить документы, просматривать их, редактировать и удалять.

Пользователь может искать любой документ в журнале по содержимому граф, выполнять поиск документов по их номерам, осуществлять отбор документов по различным признакам.

Конфигуратор позволяет создавать любое необходимое число журналов.

При создании журнала для него может быть создано произвольное число экранных форм, на которых располагаются табличные поля, содержащие колонки для отображения вида документа, номера, даты и времени, а также дополнительные графы для отображения значений любых других реквизитов документов из числа отображаемых в каждом журнале.

Если ни одной формы журнала не создано, в режиме 1С:Предприятие автоматически создается форма журнала по умолчанию.

5.10.1. Создание журнала

Для работы с журналами документов предназначена ветвь **Журналы документов** дерева конфигурации.

В системе «1С:Предприятие» процессы создания журналов и размещения документов по конкретным журналам тесно связаны между собой. Указание отображения информации документа в определенном журнале синхронизировано с данными журнала о документах, информация о которых представлена в журнале.

При создании и журнала, и документов в список журналов и документов автоматически будет добавлен созданный журнал или документ. Для отражения данных документа в журнале необходимо или в журнале, или в документе указать эту принадлежность. Указание принадлежности документа журналу можно производить как в журнале, так и в

5.10.2. Редактирование журнала

В этом разделе будут описаны уникальные свойства журналов в дополнение к общим свойствам объектов конфигурации (см. [здесь](#)), и приемы редактирования объектов конфигурации типа *Журнал*, отличные от общих приемов редактирования объектов конфигурации.

Редактирование свойств журнала (формирование списка дополнительных граф и определение их состава, форм журнала и макетов печатных форм и др.) выполняется в окне редактирования (см. [здесь](#)).

На закладке *Данные* формируется список документов, входящих в журнал, и список граф журнала.

В каждом подчиненном объекте, расположенном в ветви *Графы*, указывается реквизит всех документов, включаемый в журнал (см. [здесь](#)).

5.10.3. Графы журнала документов

Когда в конфигурации создается новый журнал документов, для работы с ним может быть создано неограниченное число форм журнала. Форма журнала создается с помощью конструктора форм объектов конфигурации (см. [здесь](#)). Конструктор размещает в форме табличное поле, содержащее набор граф для показа различных реквизитов документов. При создании формы журнала конструктор формы создает графы: картинка (для показа состояния документа), вид, дата и номер документа. Если требуется включить в журнал дополнительную информацию, то необходимо сформировать список дополнительных граф и разместить их в формах.

На закладке *Данные* окна редактирования располагается список документов, данные которых отображаются в журнале. Для создания дополнительной графы в нижнем списке нужно добавить подчиненный объект *Графа* и указать реквизиты документов, данные которых будут показываться в графе.

Для выбора реквизитов документов, которые будут размещены в графе журнала, в палитре свойств графы, в свойстве *Ссылки*, нужно нажать кнопку выбора. На экран выводится окно выбора реквизитов документов.

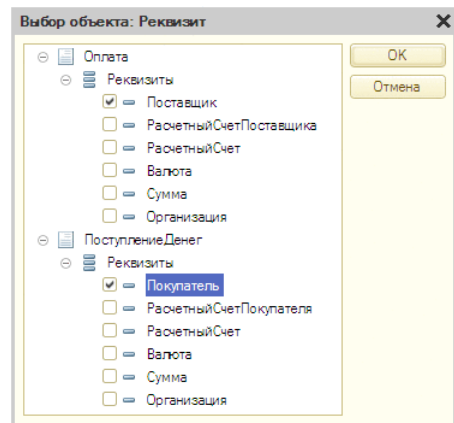


Рис. 133. Выбор реквизитов для графы

ВНИМАНИЕ! Нельзя выбрать несколько реквизитов одного и того же документа.

Если реквизит какого-либо документа не выбран, то в данной графе журнала документов будет отсутствовать информация по всем документам этого вида. При выборе следует руководствоваться исключительно здравым смыслом и не смешивать в одной графе абсолютно разные понятия (например, наименование контрагента и сумму документа).

Помимо обязательных граф документов (*Дата*, *Номер*, *Вид документа*) и граф, указанных в подчиненной группе объектов журнала *Графы*, в журнал можно добавить любое количество дополнительных граф.

Если все, входящие в состав журнала, документы со строковым типом номера имеют фиксированную длину номера, то графа журнала *Номер* также будет иметь фиксированную длину. Если хотя бы один документ, входящий в состав журнала, со строковым типом номера имеет переменную длину номера – графа журнала *Номер* создается с переменной длиной.

Новая графа добавляется в список граф выбранного объекта *Журналы*, а затем с помощью пункта *Форма – Вставить реквизиты* производится вставка графы в форму.

Наличие в журнале дополнительных граф дает возможность пользователю получить наиболее важные сведения о документе уже при просмотре журнала, не открывая сам документ.

Для элемента формы, отображающего графу журнала, автоматически применяются следующие свойства реквизитов, входящих в состав графы:

- **Режим пароля** – если такой режим установлен хотя бы у одного реквизита, входящего в состав графы.
- **Формат** – при полном совпадении у всех реквизитов, входящих в состав графы, на всех языках, определенных в конфигурации.
- **Подсказка** – если подсказки совпадают для всех реквизитов, выводится одна подсказка; если подсказки не совпадают, они выводятся через запятую.
- **Выделять отрицательные** – если данное свойство установлено у всех реквизитов, входящих в состав графы.
- **Многострочный режим** – если данное свойство установлено у всех реквизитов, входящих в состав графы.

5.10.4. Интерактивное создание нового документа

Если журнал документов включает в себя более одного типа документа, то в форме списка журнала существует два варианта создания нового документа:

- В командной панели размещена кнопка *Создать* и после ее нажатия (или после нажатия клавиши *Ins*) открывается диалог со списком типов документов, отображаемых в журнале.

Данный вариант автоматически используется в том случае, если свойство конфигурации *Режим совместимости* (см. [здесь](#)) установлено в значение *Версия 8.3.3* и младше.

- В командной панели размещена кнопка *Создать* с подменю, которое содержит перечень всех видов документов, отображаемых журналом. Такое же подменю формируется при нажатии клавиши *Ins*. Меню формируется у текущей ячейки таблицы. Если журнал отображает один тип документа, то никаких подменю не формируется.

Данный вариант автоматически используется в том случае, если свойство конфигурации *Режим совместимости* (см. [здесь](#)) установлено в значение *Версия 8.3.4* и старше или *Не используется*. Данное поведение соответствует командной панели динамического списка с автоматическим заполнением.

В случае автоматического формирования списков типов документов, эти типы отсортированы по представлению входящих документов.

Если командная панель динамического списка или формы (при условии, что основным реквизитом формы является динамический список) не использует автоматическое заполнение, то существует возможность вручную сформировать такое подменю. Для этого необходимо выполнить следующие действия:

1. Создать в командной панели группу кнопок с необходимым названием (например, *Создать*) и видом *Подменю*.

2. В созданную группу добавить стандартную команду динамического списка [Создать по параметру](#). Для кнопки, связанной с этой командой, указать в качестве значения свойства [Параметр](#), тип документа, который будет создаваться при нажатии на эту кнопку. Представлением кнопки будет служить представление соответствующего типа документа (указанного в свойстве [Параметр](#)).

3. Добавить столько команд, сколько документов планируется создавать с помощью данного меню.

При ручном создании кнопки с подменю следует учитывать следующий факт: при нажатии в динамическом списке клавиши [Ins](#), в подменю будет включен весь список типов документов, отображаемых динамическим списком, а не только те типы, которые размещены в подменю командной панели.

Если в форме списка журнала происходит обработка события таблицы [ПередНачаломДобавления](#), то в этом событии имеется возможность проанализировать, какой документ необходимо создать. Параметр процедуры обработчика события [Параметр](#) будет содержать тип создаваемого документа в том случае, если создание документа инициировано с помощью кнопки, связанной с командой [Создать по параметру](#) (с указанием этого параметра). Во всех остальных случаях значение этого параметра будет равно значению [Неопределено](#).

5.11. Перечисления

Перечисление представляет собой служебный тип данных, который не используется самостоятельно, а применяется в основном в совокупности с другими типами данных. Определить перечисление можно как список возможных значений реквизита.

Перечисления используются при вводе значений реквизитов документов, справочников, при вводе значений констант, в тех случаях, когда необходимо исключить неоднозначный ввод информации.

Рассмотрим в качестве примера такое понятие, как «статус покупателя». В простейшем случае покупатели бывают розничные и оптовые. Со статусом покупателя обычно связывают уровень предоставляемых скидок с продажной цены товаров.

Такой список «статусов» – розничный, оптовый – может служить примером простого перечисления. При выписке расходной накладной от пользователя системы требуется указать статус покупателя, выбрав его из этого списка. Выбранный статус покупателя, в свою очередь, определяет размер продажных цен.

Если статус покупателя вводится в процессе настройки конфигурации задачи как перечисление, то специалист, выполняющий конфигурирование системы «1С:Предприятие», может заранее ввести варианты расчета продажных цен в зависимости от указанного статуса.

Прежде всего, перечисление не может пополняться в процессе работы с ним: список его значений задается при настройке перечисления в конфигураторе.

Перечисление не имеет вложенности – все его значения находятся на одном уровне.

Основная особенность перечисления состоит в том, что список значений перечисления известен и доступен в конфигураторе – сама конфигурация использует конкретные значения перечисления.

Использование перечисления позволяет ограничить число возможных вариантов, например, при вводе реквизита документа. Так как список значений перечисления создается в конфигурации, то можно организовать проверку выбранного значения и описать действия, которые должны за этим выбором последовать.

Для работы с перечислениями предназначена ветвь [Перечисления](#) дерева конфигурации.

Редактирование перечисления заключается в создании списка значений перечислений. Для редактирования перечисления используется окно редактирования объекта [Перечисление](#). При создании нового перечисления свойство [Использовать стандартные команды](#) выключено.

На закладке [Данные](#) производится формирование значений перечисления.

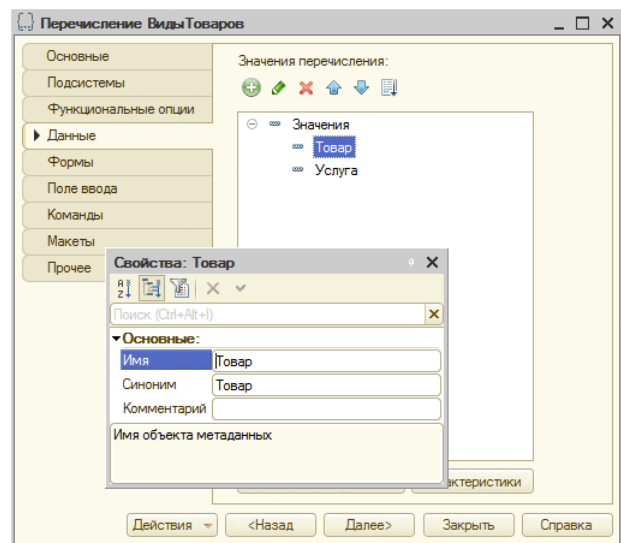


Рис. 134. Добавление значения перечисления

В палитре свойств указывается имя и синоним.

Список значений перечисления в режиме 1С:Предприятие используется следующим образом: каждое значение перечисления представляется синонимом; если синоним не задан, то используется его имя.

В приведенном на [рис. 134](#) примере значение перечисления [ВидыТоваров](#) будет представляться в виде [Товар](#) (по введенному синониму).

На закладке [Формы](#) создаются формы списка и выбора. Это позволяет создавать различные формы для выбора (в зависимости от применяемого контекста). Формы списка позволяют распечатывать списки перечислений. При создании формы списка по умолчанию автоматически включается свойство [Использовать стандартные команды](#).

На закладке [Макеты](#) могут быть созданы макеты печати.

5.12. Отчеты и обработки

Любая система автоматизации учета только тогда выполняет свои функции, когда она имеет средства обработки накопленной информации и получения сводных данных в удобном для просмотра и анализа виде. Как правило, для решения подобных задач в системе автоматизации учета существует возможность формирования отчетов. Конфигуратор позволяет формировать набор различных отчетов, достаточных для удовлетворения потребности пользователей системы в достоверной и подробной выходной информации.

Для получения отчетной информации в системе «1С:Предприятие» используются объекты конфигурации, расположенные в ветви [Отчеты](#) дерева конфигурации. Каждый объект этого типа может содержать алгоритм формирования «бумажного» или «электронного» отчета на внутреннем языке системы «1С:Предприятие» или схему компоновки данных, на основании которой система «1С:Предприятие» может автоматически выполнить отчет (см. [здесь](#)). Отчет может содержать одну или несколько форм, с помощью которых, при необходимости, можно организовать ввод каких-либо параметров, влияющих на ход алгоритма. Для вывода результатов выполнения алгоритма на экран и принтер отчет может иметь созданные с помощью конструктора макетов описания печатных форм (макеты).

Редактирование свойств объектов типа [Отчет](#) и [Обработка](#) и создание подчиненных объектов выполняются в окне редактирования (см. [здесь](#)).


```
// Откроем форму подключенной внешней обработки
ОткрытьФорму("ВнешняяОбработка." + ИмяОбработки + ".Форма");
```

Для программной работы внешняя обработка (отчет) может быть расположена:

- в файле, который расположен в конфигурации (например, в макете);
- в данных информационной базы;
- во временном хранилище (см. [здесь](#)).

Имена внешних обработок (отчетов) должны быть уникальны в пределах сеанса. Если осуществляется подключение (программное или интерактивное) внешней обработки (отчета) с именем, повторяющим имя уже загруженной в данном сеансе внешней обработки (отчета), то будет произведено отключение старой обработки и подключение новой.

ПРИМЕЧАНИЕ. При работе внешних обработок (отчетов) в толстом клиенте следует учитывать, что при работе в режиме управляемого приложения возможно открытие только управляемых форм, при работе в обычном режиме – только обычных форм.

При использовании внешних обработок (отчетов) нужно иметь в виду следующие особенности:

- если подключена новая обработка (отчет) с таким же именем, как и подключенная обработка (отчет), то открытые формы от «старой» обработки (отчета) перестают работать (генерируется ошибка);
- при подключении внешней обработки (отчета), если подключаемая обработка (отчет) двоячно идентична уже подключенной и совпадает признак безопасного режима, реального переподключения не происходит, ошибки при этом не выдается. В противном случае ранее подключенная внешняя обработка (отчет) отключается и выполняется новое подключение внешней обработки (отчета);
- при выполнении метода `Создать()`, параметр метода `БезопасныйРежим` игнорируется, если ранее было выполнено подключение внешней обработки (отчета) с помощью метода `Подключить()`. Если подключение ранее не выполнялось, то будет выполнено подключение внешней обработки (отчета) с использованием параметра `БезопасныйРежим` из метода `Создать()`;
- при получении формы для внешней обработки (отчета) будет найдена открытая форма независимо от того, открыта она для подключенной сейчас обработки или для той, которая была подключена ранее (с тем же именем);
- при открытии обработки (отчета) с помощью команды главного меню **Файл – Открыть**, форма обработки (отчета) открывается методом `ОткрытьФорму()` с параметром `Уникальность`, равным значению `Истина`, чтобы можно было открыть новую форму обработки в случае ее изменения.

5.12.1.3. Редактирование внешней обработки (отчета)

Редактирование внешней обработки (отчета) выполняется в конфигураторе.

Чтобы открыть существующую внешнюю обработку (отчет), следует выбрать пункт **Файл – Открыть**. В выданном на экран стандартном диалоге выбрать тип файла **Внешняя обработка (*.erf)** (**Внешний отчет (*.erf)**) и указать имя открываемого файла.

При открытии внешней обработки (отчета) в конфигураторе автоматически открывается окно редактирования объекта. В отличие от других объектов конфигурации, отладка внешней обработки (отчета) может производиться без перезапуска системы «1С:Предприятие». Достаточно после сохранения обработки (отчета) конфигуратором заново вызвать ее на выполнение в режиме 1С:Предприятие.

5.12.1.4. Справочная информация

Внешняя обработка (отчет) может быть снабжена пользовательским описанием. Для редактирования описания в палитре свойств внешней обработки нужно щелкнуть ссылку **Открыть** свойства **Справочная информация**.

В режиме 1С:Предприятие для просмотра описания внешней обработки (отчета) необходимо нажать клавишу **F1**.

5.12.1.5. Внешние обработки (отчеты) и объекты конфигурации

Существующие в конфигурации объекты типа **Отчет** и **Обработка** могут быть преобразованы во внешние отчеты и обработки, и наоборот, внешние отчеты и обработки могут заменять собой существующий объект конфигурации типа **Отчет** или **Обработка**. Также внешние отчеты и обработки могут быть добавлены в структуру конфигурации как новые объекты конфигурации типа **Отчет** или **Обработка**.

5.12.1.5.1. Копирование обработки (отчета) во внешнюю обработку (отчет)

Существующий объект конфигурации типа **Отчет** или **Обработка** может быть скопирован во внешнюю обработку или отчет. Для этого нужно выделить наименование объекта конфигурации в окне **Конфигурация** и в контекстном меню объекта конфигурации выбрать пункт **Сохранить как внешнюю обработку, отчет**. Затем в выданном на экран стандартном диалоге сохранения файла выбрать тип файла **Внешняя обработка (*.erf)** (**Внешний отчет (*.erf)**) и указать имя файла внешней обработки (отчета).

В результате будет создана внешняя обработка (отчет), которая будет скопирована с выбранного объекта конфигурации. Сам объект конфигурации при этом не изменится.

Выполнение этой операции целесообразно для последующей отладки создаваемого отчета или обработки. По окончании отладки внешняя обработка или отчет может быть вставлена в конфигурацию взамен существующего объекта конфигурации.

5.12.1.5.2. Замена обработки (отчета) на внешнюю обработку (отчет)

Внешние отчет или обработка могут заменить собой существующий объект конфигурации типа **Отчет** или **Обработка**.

Для замены объекта конфигурации внешней обработкой (отчетом) необходимо выделить его наименование в окне **Конфигурация** и использовать пункт **Заменить на внешнюю обработку, отчет** контекстного меню объекта конфигурации. Затем в выданном на экран стандартном диалоге открытия файла выбрать тип файла **Внешняя обработка (*.erf)** (**Внешний отчет (*.erf)**) и указать имя файла внешней обработки (отчета).

5.12.1.5.3. Добавление внешней обработки (отчета) в структуру конфигурации

Существующая внешняя обработка (отчет) может быть вставлена в структуру конфигурации как новый объект конфигурации типа **Отчет** или **Обработка**. Для этого необходимо в структуре конфигурации выделить наименование любого объекта конфигурации типа **Отчет** или **Обработка** и использовать пункт **Вставить внешнюю обработку, отчет** контекстного меню объекта конфигурации. В выданном на экран стандартном диалоге открытия файла необходимо выбрать тип файла **Внешняя обработка (*.erf)** (**Внешний отчет (*.erf)**) и указать имя файла внешней обработки (отчета), которую требуется вставить в структуру конфигурации.

В результате этих действий в дереве конфигурации появится новая обработка (отчет).

5.12.1.6. Сравнение и объединение внешних обработок (отчетов)

Внешние обработки (отчеты) можно сравнивать и объединять с обработками (отчетами), расположенными в конфигурации, а также сравнивать и объединять с другими внешними обработками (отчетами).

Для сравнения и объединения с отчетом или обработкой конфигурации в окне **Конфигурация** следует указать нужный объект, в контекстном меню этого объекта выбрать пункт **Сравнить, объединить с внешней обработкой, отчетом...** В стандартном диалоге выбора файла выбрать нужную внешнюю обработку (отчет).

На экран выводится окно **Сравнение и объединение...** Приемы работы в окне полностью совпадают с приемами работы при объединении конфигураций (см. [здесь](#)).

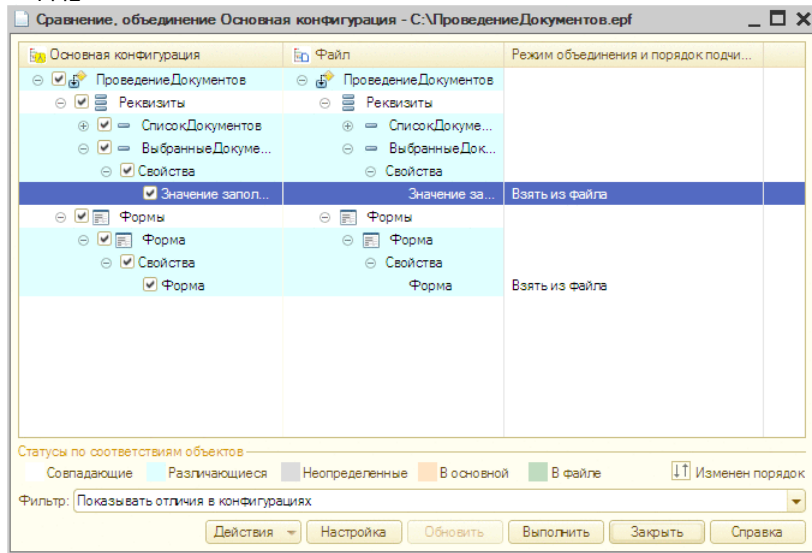


Рис. 136. Объединение обработок

Для сравнения или объединения внешней обработки (отчета) с другой внешней обработкой (отчетом) нужно открыть исходную внешнюю обработку (отчет), в окне редактирования нажать кнопку **Действия** и в выпадающем меню выбрать пункт **Сравнить, объединить с внешней обработкой, отчетом**. В стандартном диалоге выбора файла выбрать нужную внешнюю обработку (отчет). Описание дальнейших действий см. [здесь](#).

5.13. Планы видов характеристик

В системе «1С:Предприятие» объекты **Планы видов характеристик** предназначены для описания видов характеристик объектов аналитического учета.

Примером использования объектов данного типа является описание характеристик товаров, контрагентов. Они используются для реализации аналитического учета по субконто (не по субсчетам) при создании плана счетов. Основной особенностью использования планов видов характеристик является то, что объекты данного типа не описывают напрямую конкретный товар или счет, а ссылаются на такое описание. Так, для управленческого учета часто требуется описать не только такие обязательные свойства номенклатуры, как наименование, цена, артикул, поставщик, но и другие – цвет, срок годности, размер, вес, вкус и т. д. Очевидно, что для разных видов товара будет разный набор характеристик (для обуви желательно указать размер, полноту, цвет, материал и другие особенности; но эти характеристики не нужны для описания компьютерной техники). В этом случае достаточно, чтобы в конфигурации на объектном уровне были созданы все необходимые схемы описаний, а для конкретной позиции номенклатуры был выбран нужный вид описания (вид характеристики).

Конфигуратор системы «1С:Предприятие» позволяет организовать любое количество планов видов характеристик в соответствии с требованиями полноты аналитического учета на предприятии.

С точки зрения основных приемов работа с объектами типа **План видов характеристик** очень схожа с работой над объектами типа **Справочник**. Объекты также могут образовывать иерархическую структуру, они имеют одинаковый состав подчиненных объектов, их создание и редактирование можно выполнять в форме элемента, в форме списка или обоими способами и т. д.

Для объекта **План видов характеристик** разработчик конфигурации может создать набор predetermined элементов. Эти элементы не могут быть удалены пользователями в режиме 1С:Предприятие.

Настройка объектов **План видов характеристик** имеет некоторые особенности.

Объект **План видов характеристик** имеет свойство **Тип значения характеристик**, которое позволяет определить набор возможных типов данных, используемых для видов характеристик. Выбор типа значения осуществляется на закладке **Основные** окна редактирования объекта. Обычно при указании типа используют составной тип данных. Это позволяет при вводе конкретной характеристики сразу указывать нужное значение. Так, например, для плана счетов при определении видов субконто используется объект плана видов характеристик **ВидыХарактеристикТоваров**, для которого тип значения характеристики определяется как составной (см. [рис. 137](#)).

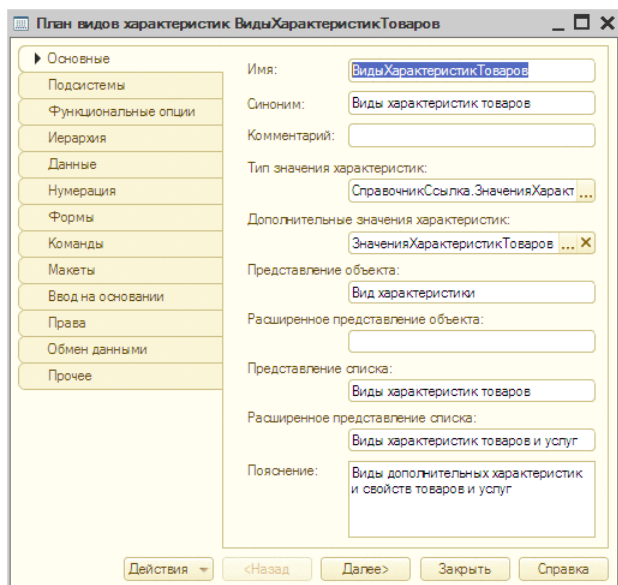


Рис. 137. План видов характеристик «Виды характеристик товаров»

Так, в случае использования объектов **План видов характеристик** для описания структуры плана счетов бухгалтерского учета виды субконто будут выбираться из predetermined видов характеристик. При создании конкретного счета указывают, какие виды субконто связаны с данным счетом.

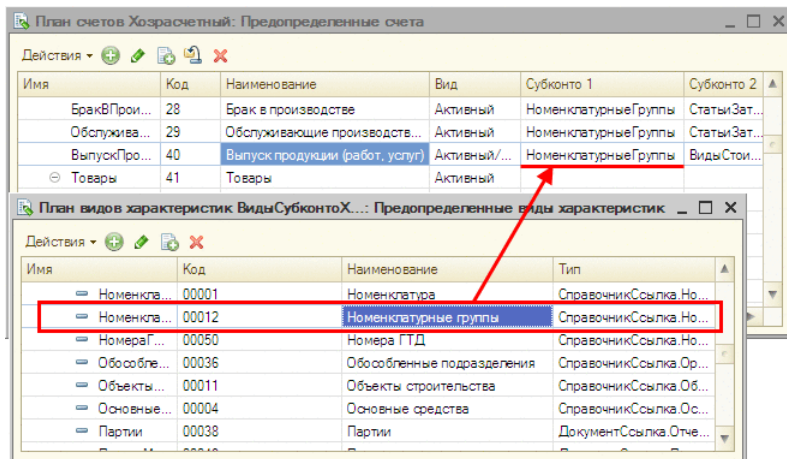


Рис. 138. Использование видов субконто

Как видно из рисунка, счет 40 «Выпуск продукции (работ, услуг)» имеет два вида субконто – **НоменклатурныеГруппы** и **ВидыСтоимости**, которые выбираются из набора предопределенных видов характеристик, определенных в данном плане видов характеристик.

Для ведения учета в разрезе характеристики, не имеющей описания (справочника) в конфигурации, используется свойство плана видов характеристик **Дополнительные значения характеристик**. Например, требуется вести учет в разрезе центров затрат, а соответствующего справочника в конфигурации нет. Тогда пользователь может создать собственный вид характеристики **ЦентрЗатрат** и указать, что значениями данного вида характеристик будут элементы дополнительного справочника. Так как данный справочник подчинен плану видов характеристик, то при выборе будут выдаваться только элементы справочника, подчиненные данному виду характеристик. Таким образом, значения от разных видов не будут смешиваться.

ВНИМАНИЕ! При выборе в свойстве **Тип значения** характеристик примитивных типов **Число**, **Строка** или **Дата** в диалоге редактирования типа данных следует указать размер или состав типа с таким расчетом, чтобы данное описание охватывало все возможные значения. Так, если дробная часть числовых типов не указана, нельзя будет ввести дробные числа. Изменение в описании числовых данных после ввода пользовательских значений может привести к потере этих данных.

Для создания предопределенных элементов в окне редактирования объекта типа **План видов характеристик** на закладке **Прочее** нужно нажать кнопку **Предопределенные**. На экран выводится окно со списком предопределенных элементов.

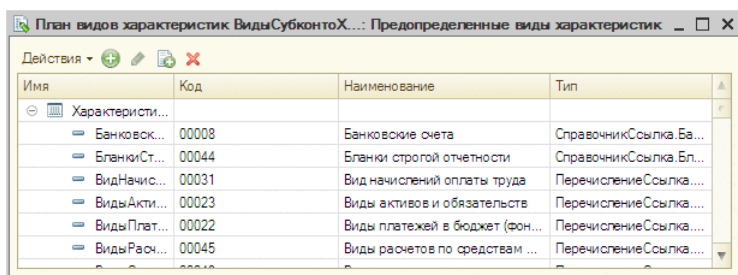


Рис. 139. Предопределенные элементы плана видов характеристик

Работа со списком предопределенных характеристик выполняется с помощью пунктов меню **Действия**.

Визуально в режиме 1С:Предприятие предопределенные элементы характеристик отличаются от элементов, созданных пользователями, видом пиктограммы.

В режиме 1С:Предприятие для элементов характеристик, созданных пользователями, тип значения можно менять.

Пример создания и использования плана видов характеристик

Можно привести примерную структуру данных для реализации хранения характеристик (несущественные для данного примера поля таблиц мы опускаем).

Справочник «Номенклатура»

Наименование
Телефон Vega 700
Телефон Vega 300
Копировальный аппарат Omega

План видов характеристик «ВидыХарактеристикНоменклатуры»

Наименование
Вес
Время работы
Формат бумаги
Основной поставщик

Регистр сведений «ХарактеристикиНоменклатуры»

Наименование (измерение)	ВидХарактеристики (измерение)	ЗначениеХарактеристики (ресурс)
Ссылка: Телефон Vega 700	Ссылка: Вес	70
Ссылка: Телефон Vega 700	Ссылка: Время работы	120
Ссылка: Телефон Vega 300	Ссылка: Вес	50
Ссылка: Телефон Vega 300	Ссылка: Время работы	80
Ссылка: Копировальный аппарат Omega	Ссылка: Вес	1300
Ссылка: Копировальный аппарат Omega	Ссылка: Формат бумаги	«А4»

Справочник «Контрагенты»

Наименование
Ссылка: Приборпоставка

Внесем в конфигурацию объекты метаданных (справочники, план видов характеристик и регистр сведений) с приведенной в таблицах структурой.

При попытке реализовать в конфигурации приведенный пример встанет один существенный вопрос. Неизвестно, какой тип нужно выбрать для ресурса [ЗначениеХарактеристики регистра](#) [ХарактеристикиНоменклатуры](#).

Действительно, характеристики имеют не только различный смысл, но и различные типы значений. Здесь и начинают играть свою роль специфические особенности плана видов характеристик. Отличием плана видов характеристик от справочника является возможность описания типов значений характеристик. В метаданных для плана видов характеристик задается свойство [Тип](#), описывающее тип значений характеристик. Это свойство имеет тип [ОписаниеТипов](#) и должно содержать все типы значений, которые могут принимать различные характеристики. Например, в данном случае для плана видов характеристик [ВидыХарактеристикНоменклатуры](#) можно указать типы:

- [Строка](#), длина 20;
- [Число](#), формат 15.2;
- [СправочникСсылка.Контрагенты](#).

Такой набор типов должен обеспечить хранение всех приведенных в примере значений характеристик. После того как мы задали в плане видов характеристик значение свойства [Тип](#) (тип значения характеристик), в перечне типов, доступных для выбора, появляется тип [Характеристика.ВидыХарактеристикНоменклатуры](#).

Теперь при выборе типа ресурса [ЗначениеХарактеристики](#) мы можем выбрать тип ([Характеристика.ВидыХарактеристикНоменклатуры](#)), который определен планом видов характеристик. Следует обратить внимание на то, что здесь необходимо выбрать не тип [ПланВидовХарактеристикСсылка.ВидыХарактеристикНоменклатуры](#), а именно тип [Характеристика.ВидыХарактеристикНоменклатуры](#). Фактически выбором типа [Характеристика.ВидыХарактеристикНоменклатуры](#) мы определяем тип ресурса [ЗначениеХарактеристики](#) косвенно, то есть указываются не конкретные типы, а указывается, что состав типов должен определяться типами, выбранными в свойстве плана видов характеристик.

Таким образом, создав план видов характеристик, мы определили возможность хранения в базе данных перечня видов характеристик товаров и определили область допустимых значений характеристик.

Однако план видов характеристик позволяет не только описывать типы значений характеристик всех видов, но и хранить в базе данных типы значений каждого вида характеристик, ведь в метаданных мы задаем типы для значений всех возможных видов характеристик, а у характеристик конкретного вида могут быть значения определенных типов. Например, вес должен задаваться именно числовым значением.

Для решения этой задачи в плане видов характеристик поддерживается поле [ТипЗначения](#). Это поле имеет тип [ОписаниеТипов](#) и предназначено для описания допустимых типов конкретных видов характеристик. Таким образом, в нашем примере данные плана видов характеристик будут иметь следующий вид:

План видов характеристик «ВидыХарактеристикНоменклатуры»

Наименование	ТипЗначения
Вес	Число
Время работы	Число
Формат бумаги	Строка
Основной поставщик	СправочникСсылка.Контрагенты

Мы описали все необходимые объекты для хранения характеристик. Однако следует учитывать, что измерения [Номенклатура](#) и [ВидХарактеристики](#) с точки зрения системы никак между собой не связаны, и при вводе значений характеристик система никак не будет учитывать выбранный вид характеристики, а будет просто предлагать заполнять поле с выбором из всех типов, описанных в плане вида характеристик.

В описании структуры регистра сведений не задается информация о логической связи полей, в которых хранятся виды характеристик и значения характеристик. В реальных решениях такая логическая взаимосвязь может быть достаточно сложной. Вид характеристики может храниться в других объектах и определяться в конфигурации сложным алгоритмом, зависящим от особенностей предметной области. Поэтому реализация взаимосвязи между видом характеристики и значением характеристики выполняется разработчиком конфигурации.

Таким образом, в нашем случае необходимо для ввода записи регистра сведений реализовать взаимосвязь между видом характеристики и значением характеристики.

Для этого необходимо установить для ресурса [ЗначениеХарактеристики](#) свойство [Связь по типу](#) в значение [ВидХарактеристики](#).

Теперь, если пользователь будет менять вид характеристики, а существующее значение не будет соответствовать допустимым для выбранного вида типам характеристики, значение характеристики будет очищаться.

Для того чтобы попробовать приведенный пример, нужно внести еще небольшое изменение в конфигурацию. Для измерения [Номенклатура](#) регистра [ВидыХарактеристикНоменклатуры](#) нужно установить свойство [Ведущее](#), чтобы данные характеристик удалялись при удалении товара и чтобы в панели навигации формы справочника появилась бы команда открытия регистра сведений.

Теперь мы можем убедиться, что у нас реализован механизм хранения характеристик. Открыв элемент справочника [Номенклатура](#), мы можем выбрать гиперссылку панели навигации [Характеристики номенклатуры](#) и начать вводить характеристики конкретного элемента номенклатуры. При этом можно будет по мере ввода характеристик создавать новые виды характеристик и указывать их тип.

В реализованном решении есть существенный недостаток. Реализована возможность вводить характеристики примитивных типов, а также тех ссылочных типов, которые определены в конфигурации. В данном случае используется справочник контрагентов для ввода основного поставщика. Однако очевидно, что часть свойств должна выбираться из некоторого набора значений. С другой стороны, состав значений для разных свойств будет различный. Соответственно, значения таких свойств не получится выбирать из справочников, имеющихся в конфигурации. В нашем примере мы вводим формат бумаги в качестве строки. Конечно, правильнее выбирать значение данного свойства из перечня возможных форматов, но заводить в конфигурации справочники для всех видов характеристик невозможно, так как справочники создаются при разработке конфигурации, а новые виды характеристик будут вводиться при использовании прикладного решения.

В плане видов характеристик предусмотрена возможность решения этой задачи. Для хранения значений перечисляемых характеристик, которые не могут быть выбраны из имеющихся в конфигурации справочников, перечислений и других ссылочных данных, используется подчиненный справочник. Создадим подчиненный справочник [ЗначенияХарактеристик](#) и установим у него подчинение плану видов характеристик [ВидыХарактеристикНоменклатуры](#). Далее нужно выбрать этот справочник в качестве значения свойства [ДополнительныеЗначенияХарактеристик](#) плана видов характеристик. Кроме того, необходимо добавить тип [СправочникСсылка: ЗначенияХарактеристик](#) в свойство [Тип](#) плана видов характеристик.

Теперь для плана видов характеристик установлено, что он для перечисляемых значений характеристик может использовать справочник [ЗначенияХарактеристик](#). В примере изменим [ТипЗначения](#) у вида характеристики [Формат бумаги](#), выбрав в нем [Справочник ссылка: Значения характеристик](#). При заполнении значения характеристики будет предложен выбор из списка справочника ограниченный владельцем – видом характеристики [Формат бумаги](#).

Структура данных полученного примера будет выглядеть следующим образом.

Справочник «Номенклатура»

Наименование
Телефон Vega 700

Телефон Vega 300
Копировальный аппарат Omega

План видов характеристик «ВидыХарактеристикНоменклатуры»

Наименование	ТипЗначения
Вес	Число
Время работы	Число
Формат бумаги	СправочникСсылка: ЗначенияХарактеристик
Основной поставщик	Справочник ссылка: Контрагенты
Цвет корпуса	СправочникСсылка: ЗначенияХарактеристик

Регистр сведений «ХарактеристикиНоменклатуры»

Наименование (измерение)	ВидХарактеристики (измерение)	ЗначениеХарактеристики (ресурс)
Ссылка: Телефон Vega 700	Ссылка: Вес	70
Ссылка: Телефон Vega 700	Ссылка: Время работы	120
Ссылка: Телефон Vega 300	Ссылка: Вес	50
Ссылка: Телефон Vega 300	Ссылка: Время работы	80
Ссылка: Копировальный аппарат Omega	Ссылка: Вес	1300
Ссылка: Копировальный аппарат Omega	Ссылка: Формат бумаги	Ссылка: А4
Ссылка: Копировальный аппарат Omega	Ссылка: Основной поставщик	Ссылка: Приборпоставка

Справочник «Контрагенты»

Наименование
Приборпоставка

Справочник «ЗначенияХарактеристик»

Владелец	Код	Наименование
Ссылка: Формат бумаги	1	А3
Ссылка: Формат бумаги	2	А4
Ссылка: Цвет корпуса	3	Белый
Ссылка: Цвет корпуса	4	Серебристый

План видов характеристик предоставляет возможность вводить виды характеристик в процессе работы с информационной базой. Однако существует возможность определить в конфигурации и предопределенные виды характеристик. В основном рекомендуется создавать такие виды не как виды характеристик по умолчанию, а как значения, используемые в логике работы самой конфигурации. Например, это может быть процент новогодней скидки. Если такая характеристика введена для товара, то алгоритм расчета цен может использовать ее при определении отпускной цены в предновогодний период.

5.14. Регистры

Регистры «1С:Предприятия» предназначены для хранения и обработки различной информации, отражающей хозяйственную или организационную деятельность предприятия.

Объекты информационной базы типа [Документ](#) и [Справочник](#) предназначены для хранения информации об объектах предметной области, таких как сотрудники, товары, материалы, валюты. Соответственно, каждый объект базы данных отражает соответствующий объект предметной области.

В регистрах обычно хранится информация об изменении состояний объектов или другая информация, не отражающая непосредственно объекты предметной области. Например, в регистрах может храниться информация о курсах валют или информация о приходе и расходе товаров.

Объект базы данных существует независимо от значений его реквизитов и имеет самостоятельную ценность. Например, у сотрудника может поменяться фамилия, номер паспорта и любые другие реквизиты. При этом он будет оставаться тем же самым физическим лицом.

После удаления объект нельзя создать заново. Даже если завести все его реквизиты в соответствии с удаленным объектом, это будет уже другой объект. Для объекта система хранит внутренний идентификатор – ссылку. Ссылка уникальна в пределах всей информационной базы. Двух объектов с одинаковыми ссылками не может существовать на всем протяжении жизни информационной базы. Ссылки удаленных объектов не присваиваются вновь созданным объектам. Система предоставляет возможность хранить в полях базы данных ссылки на объекты базы данных.

Единицей хранения информации в регистрах является запись. Прикладная нагрузка записи регистра определяется исключительно хранящимися в ней данными. Например, сама запись о курсе валюты не представляет собой ничего существенного. Она не соответствует никакому объекту в предметной области. Существенным является только то, что в ней содержится валюта, дата и курс валюты, установленный на эту дату. Можно удалить эту запись и внести такую же – это не повлияет на логику работы системы. Соответственно, у записей регистров не существует ссылок, и в полях базы данных нельзя хранить ссылки на записи регистров.

В данной главе приведено описание регистров сведений и регистров накоплений. О регистрах бухгалтерии см. [здесь](#), а о регистрах расчета см. [здесь](#).

5.14.1. Регистры сведений

В этом разделе будет рассказано о понятии «регистр сведений» и об основах использования этих регистров.

5.14.1.1. Общая информация о регистрах сведений

Основная задача регистра сведений – хранить существенную для прикладной задачи информацию, состав которой развернут по определенной комбинации значений и при необходимости развернут во времени. Например, если мы хотим хранить информацию о ценах конкурентов на продаваемые нами товары, то собранная информация о ценах разворачивается по товарам и конкурентам. А если мы хотим отслеживать динамику изменений цен и будем заносить их периодически, то хранимая информация разворачивается также и во времени.

В системе «1С:Предприятие» для хранения подобных данных и работы с ними используется специальный механизм – регистр сведений.

Регистр сведений фактически представляет собой в общем случае многомерный массив данных, необходимый для реализации функции, которая может выдать нужную информацию по определенному набору аргументов. Аргументы функции называются **измерениями**, а результат функции – **ресурсами**. В приведенном выше примере двумерный регистр [ЦеныКонкурентов](#) будет содержать измерения [Конкурент](#) и [Товар](#) и ресурс [Цена](#). Ресурсов может быть больше чем один: например, можно хранить оптовую и розничную цены.

Помимо измерений и ресурсов для регистра сведений может быть создан набор реквизитов. Реквизиты позволяют включать в записи регистров различную дополнительную информацию. Реквизиты не влияют на значения ресурсов регистра и могут использоваться для анализа записей регистра.

Регистры сведений, информация в которых развернута во времени, называются **периодическими**. Для периодических регистров сведений система поддерживает такие

стандартные операции, как получение наиболее позднего или наиболее раннего значения (например, получение последней введенной цены по конкретному товару и конкретному конкуренту), а также получение среза наиболее поздних или ранних значений. Например, могут быть получены все последние введенные цены по различным товарам и конкурентам.

Для разворота информации во времени используется поле **Период** регистра. Оно не вносится в качестве измерения, а добавляется системой автоматически при создании периодического регистра.

Для регистров сведений можно не создавать измерений. В этом случае регистр будет представлять набор периодических данных. Такие регистры могут использоваться, например, для хранения фамилий различных должностных лиц, чьи подписи располагаются в документах. В процессе ведения хозяйственной деятельности документы создаются и подписываются должностными лицами, имеющими право подписи в определенный момент времени. [здесь](#) показывалось, как использовать значения констант для подобных целей. Недостатком данного приема в случае смены значения константы является то, что при открытии архивного документа будет указана новая фамилия ответственного лица, выбранная из константы. В таких случаях нужно использовать не константу, а периодический регистр сведений, который хранит данные об изменениях, а в документах используется выбор значений из регистра сведений по дате документа.

Наиболее характерный пример одномерной периодической величины – курс валюты. При выполнении каких-либо расчетов (например, при определении рублевой цены пересчетом валютной цены по курсу) важно знать его величину на момент вычисления.

Особенно важно знать курс валюты при выполнении каких-либо расчетов задним числом – в этом случае необходимо «вспоминать» курс за уже прошедшие даты.

Чтобы иметь возможность получать подобные сведения, необходимо создать таблицу, графами в которой были бы, очевидно, наименование валюты, дата курса и сама величина курса. Строки такой таблицы содержат курс нескольких валют на конкретную дату.

Дата	Валюта	Курс
31.10.2008	USD	26,5430
31.10.2008	EUR	35,0447
01.11.2008	USD	27,0981
01.11.2008	EUR	34,4092
02.11.2008	USD	27,0793
02.11.2008	EUR	34,4828

При обращении к подобной таблице следует иметь в виду, что в колонке **Курс** хранятся конкретные величины курса на определенную дату и подразумевается, что на все последующие даты, до новой величины курса, курс не меняется. Поэтому для получения курса на какую-то промежуточную дату следует брать величину курса на ближайшую предыдущую дату, на которую существует записанный курс.

Также следует понимать, что различные значения валют в колонке **Валюта** фактически означают, что ведется параллельная история курсов нескольких валют. Иначе говоря, приведенную выше таблицу можно отобразить по-другому.

Дата	Курс USD	Курс EUR
31.10.2008	26,5430	35,0447
01.11.2008	27,0981	34,4092
02.11.2008	27,0793	34,4828

Подобных колонок курсов в таблице может быть столько, сколько курсов разных валют требуется хранить.

Если регистр неперіодический, то поле **Период** для него не создается. В приведенном примере регистр **ЦеныКонкурентов** может быть неперіодическим, если мы не хотим хранить историю изменения цен, а хотим иметь только актуальные цены. Тогда функция регистра сможет ответить на вопрос: «Какая сейчас цена у такого-то конкурента на такой-то товар», но не сможет ответить на вопрос: «Какая была цена у такого-то конкурента на такой-то товар в начале года».

Из описанных принципов работы регистра сведений вытекает то, что в системе может быть только одна запись с определенным набором и периодом измерений. Действительно, по одному товару одного конкурента может быть только одна цена. Если по какой-либо причине мы можем получить несколько цен и хотим занести эту информацию в базу данных, то нам нужно создать еще одно измерение для хранения того значения, по которому эти цены могут различаться. Например, можно завести измерение **ИсточникиИнформации**. Тогда можно будет вводить цены конкурентов в разрезе источников.

Уникальность записей по набору измерений принципиально отличает регистры сведений от регистров накоплений, которые позволяют вводить несколько записей с одинаковым значением измерений и периодом.

Если при работе с регистром получение данных на самый первый или самый последний (текущий) моменты времени являются наиболее частыми вариантами использования, то для таких регистров имеет смысл разрешить системе поддерживать итоги для среза последних (свойство **Разрешить итоги: срез последних**) или среза первых (свойство **Разрешить итоги: срез первых**). В качестве примера такого использования можно привести регистр с отпускными ценами на товары. Цена товара может меняться с течением времени, но получение текущей цены является наиболее распространенным запросом к этому регистру.

Итоги регистра сведений будут использоваться в случае выполнения всех следующих условий:

- Для регистра разрешено использование итогов в конфигурации;
- Для регистра разрешено использование итогов в режиме «1С:Предприятие»;
- Получаются данные на самую первую дату (срез первых) или самую последнюю (текущую) дату (срез последних), без указания значения периода;
- Условия для виртуальных таблиц **СрезПервых** и **СрезПоследних** задаются только на значения измерений и разделителей, находящихся в режиме **Независимо и совместно**;
- В ограничениях доступа к данным используются только измерения и разделителей, находящихся в режиме **Независимо и совместно**.

Если все условия не выполняются, то для получения информации из регистра будет использоваться обычный запрос.

5.14.1.2. Записи регистра сведений

Строки регистра сведений, содержащие информацию о значениях ресурсов для определенных значений измерений и конкретного периода, называются **записями**. Для идентификации записи регистра сведений служит **ключ записи**. Записи в регистр сведений можно вносить двумя способами:

- вручную,
- документами.

Выбор режима внесения записей см. [здесь](#).

Эти два варианта влияют на способ внесения информации, а не на основную логику работы регистра.

Документ, которым вносится запись в регистр сведений, называется регистратором.

Регистры, записываемые независимо, могут свободно редактироваться вручную или средствами встроенного языка. При этом если измерение такого регистра назначено как «ведущее» и значением измерения является ссылка на объект базы данных, то будет считаться, что запись регистра имеет смысл, только пока существует этот объект. Например, если назначить ведущим измерение **Конкурент**, то считается, что запись имеет смысл только как информация по данному конкуренту. Соответственно, при удалении данных конкурента записи по нему будут удалены автоматически.

Если регистр записывается регистратором, то это значит, что записи будут жестко подчинены регистраторам – документам. Обычно это значит, что записи будут порождаться при проведении документов. Соответственно, при удалении документа записи будут удаляться автоматически. В отличие от ведущих измерений, регистратор может быть

только один.

Следует помнить, что ключ записи регистра сведений, созданный с помощью метода `ПустойКлюч()`, не равен ключу записи регистра сведений, значения измерений которого равны значениям по умолчанию для «своих» типов.

5.14.1.3. Редактирование регистра сведений

Для работы с регистрами сведений предназначена ветвь [Регистры сведений](#) дерева конфигурации.

При редактировании регистра определяются его свойства, разрабатывается структура регистра: создаются наборы измерений, ресурсов и реквизитов регистра, создаются экранные формы просмотра и редактирования записей регистра и, если необходимо, печатные формы регистра (см. [здесь](#)).

В этом разделе в дополнение к общим свойствам объектов конфигурации будут описаны уникальные свойства регистра сведений.

Регистр редактируется в окне редактирования [Регистр сведений](#). Свойства регистра собраны на закладках.

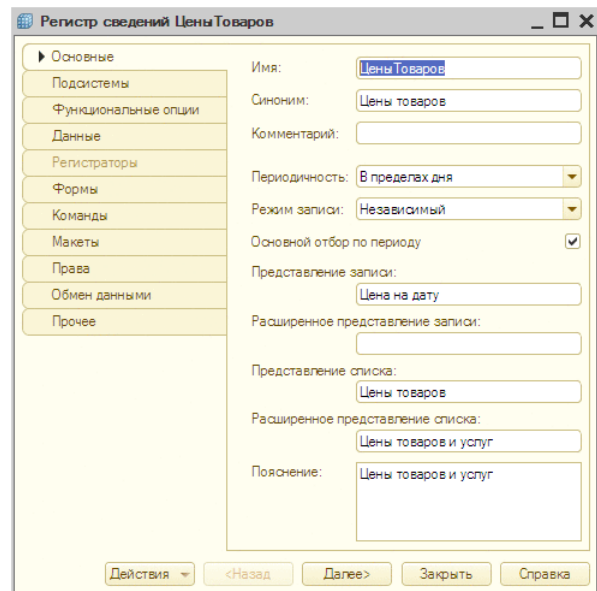


Рис. 140. Редактор регистра сведений

Периодичность – это свойство позволяет указать, с какой периодичностью регистр должен сохранять значения ресурсов.

Это свойство напрямую влияет на возможность получения значений ресурсов регистра методами встроенного языка. Для непериодического регистра можно получить только последнее введенное значение ресурсов регистра – информация за предыдущие периоды в таком регистре отсутствует. Для периодических регистров нельзя получить значение с меньшей периодичностью, чем установленная периодичность регистра.

Периодичность не зависит от способа редактирования регистра.

Режим записи – свойство определяет, каким образом будут вноситься записи: независимо (например, вручную) или будут подчиняться регистратору (например, документами).

Если для регистра указано, что он периодический и выбран независимый режим записи, то становится доступным свойство **Основной отбор по периоду**. Если флажок установлен, то наряду с основными измерениями и реквизитами регистра, участвующими в отборе записей при регистрации изменений, можно указывать отбор по полю **Период**.

На закладке **Данные** формируется структура данных регистра. Создаются измерения и ресурсы, а также реквизиты.

На закладке **Регистраторы** производится управление списком регистраторов. Закладка доступна, если в свойстве **Режим записи** установлено **Подчинение регистратору**.

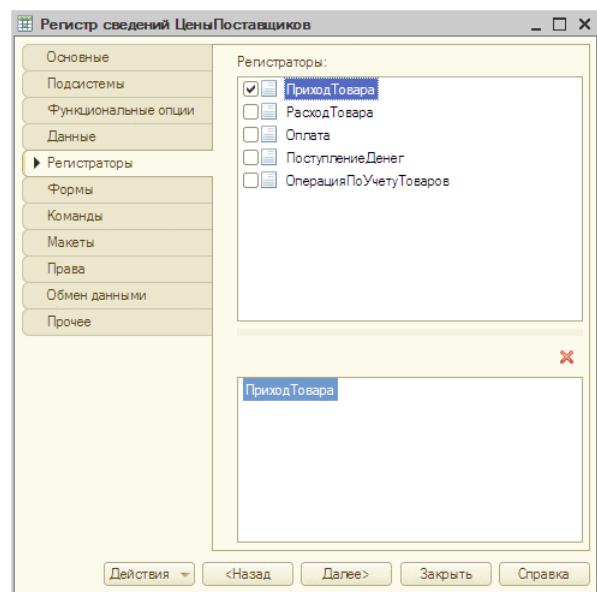


Рис. 141. Задание регистраторов регистра сведений

В верхнем списке производится управление списком регистраторов (устанавливаются или снимаются отметки), а в нижнем списке приводится список отмеченных объектов, являющихся регистраторами.

Аналогично описанному для регистров сведений механизму управления списком регистраторов производится управление списком регистраторов для других видов регистров.

Особенностью ресурсов регистров сведений является широкая типизация данных, в отличие от регистров других типов, где ресурсы могут быть только числовыми.

Описание приемов создания форм и макетов см. [здесь](#).

5.14.1.4. Разработка структуры регистра сведений

Разработка структуры регистра заключается в создании наборов измерений, ресурсов и реквизитов.

Для управления списком измерений, ресурсов и реквизитов регистра и редактирования их свойств служат управляющие элементы групп [Измерения](#), [Ресурсы](#), [Реквизиты](#) окна редактирования [Регистр](#). С точки зрения настройки элементы этих групп одинаковы. Описание порядка использования этих управляющих элементов см. [здесь](#).

5.14.1.4.1. Свойства измерения (ресурса, реквизита) регистра сведений

Свойства измерений, ресурсов и реквизитов редактируются при помощи палитры свойств. В основном они совпадают с общими свойствами объектов конфигурации. Ниже в этом разделе будут описаны уникальные свойства измерений, ресурсов и реквизитов.

Ведущее – установка этого свойства имеет смысл для измерений, тип данных которых – ссылка на объект конфигурации. В этом случае считается, что запись регистра сведений имеет смысл, только пока существует этот объект. При удалении объекта записи по нему будут автоматически удалены из регистра.

Запрет пустых значений – установка этого флажка включает механизм запрета записи регистра с пустым значением измерения.

Индексировать – для измерений свойство доступно для редактирования, если измерение не является ведущим. Для измерений, ресурсов и реквизитов с установленным свойством **Индексировать** создается отдельный индекс, что увеличивает производительность при работе с регистром. Для ведущих измерений индекс создается всегда.

При просмотре регистра в режиме 1С:Предприятие существует возможность сортировать записи регистра по индексированным измерениям, ресурсам и реквизитам. Необходимое число форм для просмотра и редактирования регистра должно быть создано в процессе разработки конфигурации.

5.14.1.4.2. Упорядочивание списка измерений регистра сведений

Порядок расстановки измерений регистра сведений имеет важное значение. Измерения, к которым необходим быстрый доступ, следует располагать в начале списка измерений.

Последовательность расстановки измерений регистра сведений влияет на возможность применения методов встроенного языка, использующих позиционный доступ к измерениям.

Также необходимо иметь в виду, что изменение порядка измерений требует реструктуризации информационной базы.

5.14.2. Регистры накопления

Регистры в системе «1С:Предприятие» используются для накопления информации о наличии и движении каких-либо величин – материальных, денежных и других. Вся информация о хозяйственных операциях, которая вводится с использованием документов или формируется при помощи расчетов, должна быть накоплена в регистрах. Тогда эту информацию можно будет извлечь, проанализировать и представить пользователю в виде отчетных форм.

В этом разделе будет рассказано о понятии регистр накопления и даны сведения об основах использования.

5.14.2.1. Общая информация о регистрах накопления

Регистр накопления – это объект конфигурации, предназначенный для хранения движений регистра и итоговой информации.

Проблема, которая обычно возникает при создании «хранилища» сводной информации, состоит в определении его структуры: в каких разрезах следует накапливать сводные данные, чтобы затем можно было извлечь нужную информацию без утомительной обработки. Система «1С:Предприятие» использует простые и в то же время гибкие средства для создания регистров накопления: достаточно просто задать, в каких разрезах и какие данные требуется хранить в регистре, а система сама обеспечит запись и получение нужных данных простыми языковыми средствами.

Методы встроенного языка позволяют получить остатки регистра накопления на заданный момент времени. Есть возможность фильтрации по значениям измерений, а также получения остатков в разрезе других измерений.

Рассмотрим пример. Предположим, что в создаваемой программе торгово-складского учета требуется хранить сведения о количестве и стоимости каждого товара на каждом складе. В дальнейшем предполагается получать информацию такого типа: «остаток конкретного товара на конкретном складе», «остаток конкретного товара всего, на всех складах», «стоимость всех товаров на конкретном складе».

В идеологии системы «1С:Предприятие» регистр накопления такого вида представляет собой прямоугольную систему координат, на одной оси которой находятся склады, а на другой – товары, а на пересечении конкретного склада и конкретного товара находятся цифры количества товара и стоимости товара.

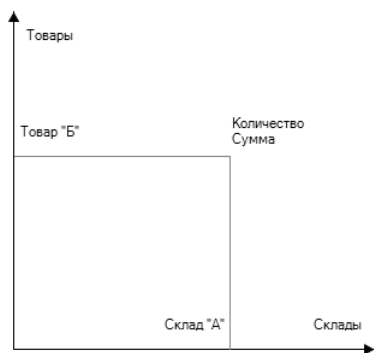


Рис. 142. Регистр накопления

Физический смысл регистра накопления сформулировать довольно сложно, и, скорее всего, регистр накопления не имеет материального аналога.

ВНИМАНИЕ! Поэтому определим, что регистр накопления – это n-мерная система координат, в узлах которой хранятся совокупные данные. Оси такой системы координат будем называть измерениями регистра, а хранящиеся в узлах данные – ресурсами регистра.

5.14.2.2. Движения регистра накопления

Изменение состояния регистров накопления выполняется обычно при проведении документа. Процедура проведения документа расположена в модуле документа и содержит алгоритм формирования сведений об изменениях в регистрах, которые необходимо выполнить при проведении документа. Эти сведения называются движениями регистра. Механизм подсчета итогов использует движения регистров для выполнения непосредственных изменений в регистрах накопления. Таким образом, движения регистров содержат только приращения (со знаком плюс или минус) значений ресурсов регистра, а не итоговые величины значений.

Специалист, выполняющий конфигурирование системы, имеет возможность предоставить конечному пользователю средства просмотра движений регистров. Конфигуратор позволяет создавать экранные и печатные формы для просмотра и анализа движений регистров.

В процессе разработки конфигурации можно создать неограниченное количество регистров накопления. Однако следует учитывать, что запись изменений в большом числе регистров при проведении документа может вызывать уменьшение скорости работы системы в целом.

Помимо измерений и ресурсов для регистра накопления может быть создан набор реквизитов. Реквизиты позволяют включать в движения регистров различную дополнительную информацию. Реквизиты не влияют на значения ресурсов регистра и могут использоваться для анализа движений регистра.

5.14.2.3. Итоги регистра накопления

Как было написано выше, изменения в регистрах вносятся **движениями** регистра. Движения регистров оказывают влияние на его итоги. **Итоги** – это сводная информация регистров, которая получается путем суммирования значений, вносимых движениями регистров.

Итоги регистра накопления можно представить в виде таблицы с количеством колонок, равным сумме измерений и ресурсов регистра накопления. Количество строк в таблице итогов регистра накопления будет зависеть только от количества уникальных значений измерений, и не будет зависеть от количества уникальных значений ресурсов.

Товар	Склад	Количество	Сумма
Стол	Розничный	10	5 000
Стол	Оптовый	5	2 500
Шкаф	Временный	7	10 500
Шкаф	Оптовый	2	3 000
Шкаф	Розничный	10	15 000

Из таблицы видно, что измерение **Товар** принимает значения **Стол**, **Шкаф**, а измерение **Склад** – **Временный**, **Оптовый** и **Розничный**. В колонках **Количество** и **Сумма**, отражающих ресурсы регистра накопления, записано количество и сумма каждого товара на каждом складе.

В отличие от движений регистра, нет возможности непосредственно просматривать итоги регистра накопления. Для обращения к итогам в конфигурации может быть создано необходимое число отчетов, которые будут обращаться к итогам и выдавать их в виде товарных отчетов, складских карточек и ведомостей и т. д.

5.14.2.4. Регистры остатков и регистры оборотов

В системе «1С:Предприятие» возможно использование регистров накопления двух типов: регистры остатков и регистры оборотов.

Для регистра остатков методы встроенного языка позволяют получить остатки регистра накопления на заданный момент времени. Есть возможность фильтрации по значениям измерений, а также получения остатков в разрезе других измерений.

Регистры оборотов предназначены для хранения информации, для которой понятие остатка лишено смысла, например, сумм продаж в разрезе покупателей.

Рассмотрим в качестве примера отслеживание взаиморасчетов с покупателями товаров, которые производит или продает предприятие (потребителями услуг, оказываемых предприятием, и так далее). Можно утверждать, что ведение подобного учета – обязательная часть общего учета на любом предприятии.

Для того чтобы оперативно получать информацию о взаимной задолженности предприятия и покупателя, потребуется регистр **Взаиморасчеты**, в котором для каждого покупателя будет храниться сумма задолженности. При совершении хозяйственной операции состояние регистра будет соответствующим образом изменяться, каждый раз отражая текущее состояние взаиморасчетов. Регистр **Взаиморасчеты** – это регистр остатков.

Однако быстро получить информацию об объеме закупок, совершенных данным покупателем за какой-либо период времени, из регистра **Взаиморасчеты** нельзя – он такой информации не хранит. Поэтому придется приложить дополнительные усилия для ее получения: например, можно включить в структуру регистра реквизит **Контрагент**, а затем отобрать движения регистра по нужному контрагенту и вычислить общую сумму закупок. Но когда необходимо получать эти сведения оперативно (например, при достижении определенного объема закупок покупателю должна предоставляться скидка), такой способ, конечно же, не подходит.

В этом случае решением проблемы может быть использование регистра оборотов. В таком регистре – назовем его **Объем закупок** – для каждого покупателя будет храниться информация об объеме закупок (об обороте покупателя).

Теперь при совершении хозяйственных операций необходимо будет изменять не только состояние регистра **Взаиморасчеты**, но и регистр **Объем закупок**. В этот регистр при совершении клиентом каждой покупки будет заноситься информация о сумме покупки. В результате в регистре **Объем закупок** будет постоянно накапливаться информация об общем объеме закупок клиента.

Из всего сказанного выше можно сделать выводы о преимуществах использования регистров.

Прежде всего, регистры используются для хранения информации, к которой требуется получать оперативный доступ. «Уровень оперативности» и, соответственно, целесообразность использования регистра должен определять специалист, выполняющий конфигурирование системы «1С:Предприятие», в соответствии с требованиями пользователей системы.

Также можно сказать, что регистры позволяют получать наиболее достоверную информацию о состоянии средств. Так как процессы сохранения документа и записи изменений в регистрах разделены (допускается сохранение документа без его проведения), может возникнуть расхождение между данными документов и информацией в регистрах. Но регистр, в отличие от документа, является хранилищем итоговой информации, поэтому именно запись изменений в регистры служит подтверждением того, что хозяйственная операция совершена.

5.14.2.5. Основные свойства регистра накопления

Для работы с регистрами накопления предназначена ветвь **Регистры накопления** дерева конфигурации.

Редактирование свойств объектов типа **Регистр накопления** и создание подчиненных объектов выполняются в окне редактирования (см. [здесь](#)).

При редактировании регистра накопления определяется его вид, разрабатывается структура регистра:

- создаются наборы измерений, ресурсов и реквизитов регистра;
- если необходимо, создаются экранные и печатные формы просмотра движений регистра.

ПРИМЕЧАНИЕ. В качестве типа измерения регистра накопления не могут выступать значения типа **УникальныйИдентификатор**, **ДвоичныеДанные**, строка неограниченной длины.

В этом разделе будут описаны уникальные свойства регистра накопления, в дополнение к общим свойствам объектов.

Вид регистра – если регистр предназначен:

- для хранения остатков (выберите из списка значение **Остатки**);
- хранения оборотов – значение **Обороты**.

Описание разницы между регистрами остатков и оборотов см. [здесь](#).

Основная форма списка – для регистра может быть создано несколько форм для просмотра его движений. Если форм для ввода и выбора несколько, то в свойстве **Основная форма списка** можно указать форму, которая будет использоваться по умолчанию.

Разрешить разделение итогов – если флажок установлен в значении **Истина** (значение по умолчанию), то будет задействован механизм разделителя итогов, который обеспечивает более высокую параллельность работы при записи в регистр. При одновременной записи движений несколькими сеансами система не будет обновлять одни и те же записи итогов, а будет записывать изменения итогов отдельно. При получении итогов эти данные складываются. Таким образом, обеспечивается и поддержание в актуальном состоянии итогов (для быстрого получения отчетов, например), и параллельность записи движений. Этот режим требует дополнительных расходов ресурсов (например, увеличивается количество данных в итоговых таблицах). Поэтому свойства есть и в конфигурации, и в языке для управления этим режимом.

Записи будут «размножаться» только при параллельно выполняемых транзакциях. Их количество по каждой комбинации измерений будет зависеть от максимального количества одновременно выполняемых транзакций. При пересчете итогов накопленные отдельные записи сворачиваются.

Режим разделения итогов может быть изменен пользователем в режиме работы 1С:Предприятие. По умолчанию свойство включено.

Использование в итогах – если флажок установлен в значение *Ложь*, измерение исключается из хранимых итогов регистра.

Также для регистра может быть создано несколько форм для просмотра его движений. Если форм для ввода и выбора несколько, то в свойстве **Основная форма списка** можно указать форму, которая будет использоваться по умолчанию.

5.14.2.6. Агрегаты оборотных регистров накопления

Для повышения производительности системы в случае использования оборотных регистров накопления предназначен механизм **агрегатов**. Можно сказать, что агрегаты – это специализированные хранилища, предназначенные для использования в рамках механизмов запросов «1С:Предприятия».

5.14.2.6.1. Основные понятия

При дальнейшем рассмотрении работы с агрегатами будут использоваться некоторые термины, которые мы определим в этом разделе.

Агрегат – физическая таблица базы данных, хранящая сводные обороты всех ресурсов регистра по выбранным измерениям с выбранной периодичностью и за определенный период. В регистре, для которого формируются агрегаты, не может быть более 30 измерений.

Агрегат характеризуется следующими параметрами:

- **Размер агрегата** – это размер таблицы агрегата. **Оценочный показатель.**
- **Эффект** – ожидаемое уменьшение среднего времени выполнения запроса с использованием агрегатов. Например, если эффект агрегата равен 90 %, то это означает, что среднее время выполнения запроса с использованием агрегатов будет на 90 % меньше, чем среднее время выполнения того же запроса, но с использованием итогов. **Оценочный показатель.**

Период агрегата – интервал дат, данные за который помещены в агрегат.

Периодичность агрегата – периодичность, с которой в агрегате хранятся данные.

Список агрегатов – набор агрегатов, заданный на этапе конфигурирования. Список может быть сформирован либо вручную, либо загружен из файла, полученного в результате расчета оптимальных агрегатов.

Статистика использования – информация о том, какие запросы (измерения, период, периодичность) выполняются к регистру. Используется для перестроения агрегатов и получения оптимального списка агрегатов.

Оптимальный список агрегатов – список агрегатов, соотношение размера и эффекта которых является оптимальным для текущего состояния регистра (его движений и статистики использования).

Режим агрегатов/итогов – если установлен режим агрегатов, то при выполнении запросов будут использоваться данные агрегатов; если установлен режим итогов, то при выполнении запросов будут использоваться данные итогов.

Использование агрегатов – выключение использования агрегатов означает, что при изменении движений не будет выполняться никаких операций над агрегатами. Выключать использование агрегатов имеет смысл на время массовой загрузки данных в регистры, однако последующее включение использования агрегатов может привести к ресурсоемкому процессу актуализации агрегатов (если изменяемые данные находятся внутри периода агрегатов).

5.14.2.6.2. Общая схема работы с агрегатами

Приведем общую схему работы с агрегатами:



Рис. 143. Общая схема работы с агрегатами (клиент-серверный вариант)

ПРИМЕЧАНИЕ. Данная схема описывает работу с одним регистром. Если необходимо работать с несколькими регистрами, то операцию, выполняемую на каждом шаге, следует выполнять для каждого регистра.

Более подробно распишем приведенную схему работы:

1. Нужно создать в конфигураторе необходимый список агрегатов. Данный шаг не является обязательным. Создание агрегатов в режиме конфигуратора может использоваться в том случае, если требуется обеспечить постоянное использование какого-либо агрегата в любой информационной базе.

Если агрегаты решено создавать в конфигураторе, то это можно сделать несколькими способами:

- Рассчитать (например, в режиме итогов) и загрузить оптимальные агрегаты (метод `ОпределитьОптимальныеАгрегаты()`). Если агрегаты ранее не использовались, то список оптимальных агрегатов будет получен только на основании таблицы движений регистра. Если агрегаты использовались, то список оптимальных агрегатов будет построен на основании таблицы движений и статистики использования.

- Создать собственный список агрегатов на основании анализа запросов к регистру.

2. После обновления конфигурации базы данных включить для регистра режим агрегатов (метод `УстановитьРежимАгрегатов()`).

3. Далее необходимо периодически выполнять перестроение агрегатов (метод `ПерестроитьИспользованиеАгрегатов()`). При выполнении этой операции система добавляет нужные агрегаты и удаляет неиспользуемые. Данная операция выполняется в том случае, если текущий список агрегатов не является оптимальным. Удаление производится только для тех агрегатов, которые создавались во время выполнения операции перестроения. Агрегаты, созданные в конфигураторе, автоматически не удаляются.

Если агрегаты созданы в конфигураторе, то при выполнении операции перестроения с ними будут выполняться следующие действия:

- Агрегат будет включен в том случае, если вариант использования агрегата установлен в значение **Всегда**. Оценка эффективности использования такого агрегата не влияет на его использование.
 - Агрегат будет включен на основании оценки эффективности использования в том случае, если режим использования агрегата установлен в значение **Авто**. Использование данного агрегата оценивается наряду с агрегатами, которые система создает автоматически. Если автоматически создаваемый агрегат окажется эффективнее агрегата, созданного в конфигураторе с режимом использования **Авто**, то будет использоваться автоматически создаваемый агрегат.
4. Затем необходимо выполнить обновление агрегатов (метод `ОбновитьАгрегаты()`). Обновление агрегатов производит перенос данных из таблиц движений выбранного регистра в соответствующие таблицы агрегатов. Переносятся те движения, которые были созданы в таблице движений после предыдущего обновления агрегатов.
5. Далее следует накопить статистику использования созданных агрегатов. Для этого следует в течение некоторого интервала времени (например, 1 месяц) выполнять типовые задачи, в которых используются данные из регистра, для которого был включен режим агрегатов. В процессе работы необходимо регулярно выполнять обновление агрегатов. По истечению этого периода следует выполнить перестроение агрегатов (шаг 3) и регулярно повторять операции шагов 3 – 5.

Рассмотренная схема описывает подход к работе с агрегатами регистра в клиент-серверном варианте.

В файловом варианте использования системы, работа с агрегатами выглядит несколько по-иному:

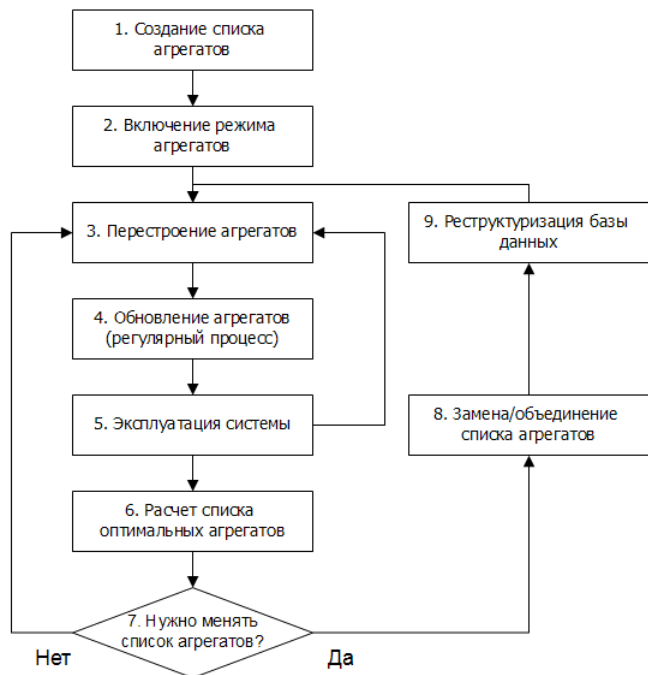


Рис. 144. Общая схема работы с агрегатами (файловый вариант)

Основное отличие этой схемы от общей схемы работы с агрегатами (см. рис. 144) заключается в следующем (будут описаны только отличия):

- На шаге 1 список агрегатов в конфигураторе следует создавать обязательно.
- Шаг 6. Пересчет оптимальных агрегатов (метод `ОпределитьОптимальныеАгрегаты()`) следует выполнять с некоторой регулярностью (например, 1 месяц). Во время этого периода (1 месяц) следует выполнять типовые задачи, в которых используются данные из регистра, для которого был включен режим агрегатов. В процессе работы необходимо регулярно выполнять обновление агрегатов.
- Шаг 7. После получения нового списка оптимальных агрегатов, необходимо определить, требуется изменение списка агрегатов в метаданных конфигурации или нет. Если изменение не требуется, то следует продолжить работу, перейдя на шаг 3.
- Шаг 8. Если необходимо выполнить обновление списка агрегатов, то следует выполнить загрузку необходимых (или всех) агрегатов (из списка оптимальных). Затем нужно обновить конфигурацию базы данных (произойдет реструктуризация информационной базы) и потом продолжить работу с шага 3.

Рекомендации по выполнению каждого из рассмотренных шагов рассматриваются в следующем разделе.

5.14.2.6.3. Рекомендации по использованию агрегатов

Формирование списка агрегатов

Формирование списка агрегатов можно выполнять несколькими способами. Если агрегаты планируется включить для регистра, уже существующего в конфигурации, но для которого не было включено использование агрегатов, то это можно сделать следующими способами:

1. Если в регистре отсутствуют движения или движений мало (не более 2–3 тысяч записей), следует выполнить анализ запросов, которые используют выбранный регистр, и получить перечень часто используемых комбинаций измерений, отборов, периодов запросов и периодичности получения данных. На основании полученной информации создать список агрегатов, при этом постараться минимизировать количество используемых агрегатов.
2. Если в регистре присутствует существенное количество записей (более 3 тысяч), то можно выполнить расчет оптимальных агрегатов на основании данных регистра с помощью метода `ОпределитьОптимальныеАгрегаты()` и загрузить полученный список агрегатов.

Если для используемого регистра был включен режим агрегатов и включено использование агрегатов (т. е. каким-то образом сформирован первоначальный список агрегатов), то можно также получить список оптимальных агрегатов и загрузить его в конфигураторе. Отличие этого способа от способа номер 2 заключается в том, что в этом случае для расчета оптимальных агрегатов будет использоваться статистика, которую ведет система. Поэтому полученный список агрегатов будет более эффективен, нежели получение оптимальных агрегатов для регистра, у которого не был включен режим агрегатов и использование агрегатов.

ПРИМЕЧАНИЕ. Использовать метод получения оптимальных агрегатов имеет смысл в том случае, если в таблице движений регистра находятся не менее 3–5 тысяч записей. На меньшем количестве записей построенный список агрегатов может оказаться неэффективным.

Выполнение перестроения

Операцию перестроения агрегатов следует выполнять не реже, чем операцию расчета списка оптимальных агрегатов. При этом данная операция не предполагает изменения списка агрегатов, а старается «обойтись» только существующими агрегатами.

Также следует помнить, что операция перестроения эффективна в том случае, если для ее работы накоплен достаточно большой объем статистических данных.

Однако рекомендации по периоду перестроения в общем случае дать затруднительно, но можно отметить основные факторы:

- есть вероятность изменения характера данных в регистре;
- есть подозрения на изменение характера запросов (что приведет к изменению накапливаемой статистики).

При выполнении операции следует указать два параметра:

- Максимальный относительный размер – задает ограничение на размер формируемого списка агрегатов (в процентах от таблицы движений). Если параметр равен 0, то ограничения на размер агрегатов не задаются.
- Минимальный эффект – процент, на который необходимо увеличить эффект старого списка при перестроении. Если новый список увеличивает эффект на заданное значение, то метод реально перестраивает список. Если параметр не указан или равен 0, это означает отсутствие требований к минимальному эффекту.

Также происходит перестроение текущего списка агрегатов, если он больше, чем параметр **Максимальный относительный размер**, или удалось построить список с эффективностью большей как минимум на значение параметра **Минимальный эффект**. В противном случае список агрегатов не перестраивается.

ПРИМЕЧАНИЕ. Операция перестроения является достаточно длительной и ресурсоемкой процедурой. Не рекомендуется выполнять ее во время интенсивной работы с информационной базой других пользователей.

Выполнение операции обновления агрегатов

Данную операцию рекомендуется выполнять чаще, чем выполняется операция перестроения агрегатов.

Обновление агрегатов может выполняться двумя способами:

- Полностью обновляются все агрегаты, отмеченные как используемые. Это может занимать существенное время.
- Так называемое «порционное» заполнение. В этом случае за одно обращение обновляется период, равный 1 месяцу в 10 агрегатах.

При выполнении операции следует указать параметр **Максимальный относительный размер**. Он задает ограничение на размер формируемого списка агрегатов (в процентах от таблицы движений). Если параметр равен 0, то ограничения на размер агрегатов не задаются.

ПРИМЕЧАНИЕ. Желательно выполнять данную операцию в периоды минимальной нагрузки на информационную базу.

СОВЕТ. Рекомендуется использовать режим разделения итогов при использовании агрегатов, особенно если обновление агрегатов выполняется регламентным заданием на фоне интенсивного проведения документов, связанных с регистром, у которого выполняется обновление агрегатов.

Использование агрегатов

Для определения агрегата, который будет использоваться для запроса, будет использован алгоритм, который будет описан далее.

В запросе выделяются используемые измерения регистра, потом подбирается перечень агрегатов, которые содержат все используемые измерения, и наиболее полно соответствуют запросу по периодичности и периоду. Из этого списка выбирается агрегат, с минимальным размером. Именно этот агрегат и будет использоваться.

В запросе может использовать агрегат, периодичность которого меньше периодичности, требуемой в запросе. В этом случае необходимые данные будут получены суммированием данных из меньших периодов.

Возможна ситуация, когда период, заданный в запросе, не совпадает с периодом агрегатов. В этом случае для исполнения запроса возможно использование двух агрегатов. Рассмотрим пример. В информационной базе существуют два агрегата, закрывающие весь период запроса:

- С периодичностью месяц;
- С периодичностью день.

Запрос исполняется за период с 15 сентября по 15 ноября. В этом случае будут использованы два агрегата:

- Агрегат с периодичностью месяц будет использован для получения данных за период с 1 октября по 31 октября;
- Агрегат с периодичностью день будет использован для получения данных за периоды с 15 сентября по 30 сентября и с 1 ноября по 15 ноября;
- Данные по разным агрегатам будут суммированы для получения окончательного результата.

Виртуальная таблица оборотов регистра, для которого включен режим агрегатов, всегда содержит актуальные данные.

Выполнение расчета оптимальных агрегатов

Данная операция может выполняться по необходимости, она не является регулярной. Ниже приведен перечень ситуаций, при наступлении которых рекомендуется выполнить расчет списка оптимальных агрегатов:

- по истечении некоторого времени после **первичного** формирования списка агрегатов;
- в случае **существенного** падения производительности на текущем списке агрегатов;
- при **существенном** изменении характера данных;
- при изменении состава запросов к регистру;
- если есть вероятность, что текущий список агрегатов перестал быть оптимальным.

ПРИМЕЧАНИЕ 1. Для клиент-серверного варианта данная операция не является обязательной. Формирование оптимальных агрегатов (при необходимости) выполняется автоматически во время выполнения перестроения агрегатов.

ПРИМЕЧАНИЕ 2. Данная операция является наиболее ресурсоемкой и продолжительной. Настоятельно рекомендуется выполнять ее только в тех случаях, когда с информационной базой не работают другие пользователи.

Использование регламентных заданий при работе с агрегатами

Операции перестроения и обновления агрегатов можно выполнять с помощью регламентных заданий. Необходимо соблюдать следующие рекомендации при составлении расписания выполнения регламентных заданий:

- для регламентного задания, выполняющего операции перестроения и обновления, рекомендуется заполнять свойство регламентного задания **Ключ** (см. [здесь](#)).
- рекомендуется задавать расписание регламентных заданий перестроения и обновления так, чтобы не возникла ситуация, когда операции обновления и перестроения будут выполняться в одно время.

Соблюдение рекомендаций приведет к предотвращению параллельного выполнения операций перестроения и обновления, что положительно скажется на производительности.

5.14.2.6.4. Редактирование агрегатов

Создание и редактирование списка агрегатов возможно только для оборотного регистра накопления (свойство **Вид регистра** имеет значение **Обороты**). Для вызова конструктора агрегатов следует использовать команду **Открыть агрегаты** контекстного меню соответствующего регистра.

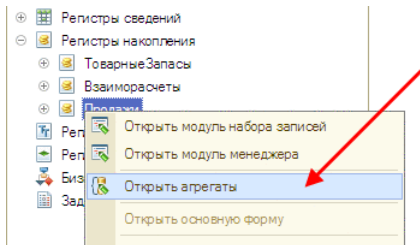


Рис. 145. Вызов конструктора агрегатов

Затем откроется окно конструктора агрегатов, в котором можно управлять агрегатами оборотного регистра накопления. Агрегатами можно управлять вручную, а также можно загрузить (для этого предназначена специальная кнопка командной панели) заранее подготовленный список оптимальных агрегатов.

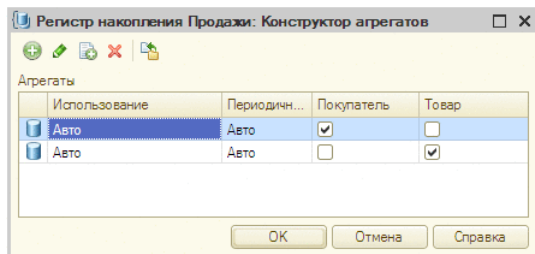


Рис. 146. Конструктор агрегатов

При создании агрегатов можно указать вариант использования. Если стоит **Авто** (по умолчанию), то система будет сама определять необходимость использования данного агрегата во время выполнения операции перестроения агрегатов. Если стоит **Всегда**, значит система будет использовать агрегат всегда.

Колонка **Периодичность** определяет минимальный период времени, за который агрегат будет хранить итоги по выбранным измерениям. Допускается иметь несколько агрегатов с одинаковым набором измерений, но с разной периодичностью. Не следует злоупотреблять количеством агрегатов. Большое количество агрегатов может привести к излишнему увеличению размера базы данных, но не приведет к повышению производительности работы запросов.

В правой части окна можно указать, какие измерения входят в редактируемый агрегат. Агрегат может включать произвольное количество измерений (но не более 30) и не включать их вовсе. В этом случае система хранит сводные обороты по регистру с заданной периодичностью.

Если существует xml-файл, содержащий список оптимальных агрегатов, то можно выполнить загрузку такого списка. Для этого стоит воспользоваться специальной командой конструктора агрегатов (см. [рис. 146](#)) и выбрать там предварительно подготовленный файл.

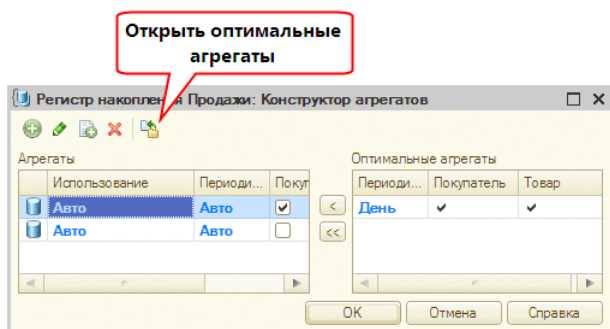


Рис. 147. Загрузка списка оптимальных агрегатов

Система выполнит сравнение списка из файла и текущего списка агрегатов и цветом отметит те агрегаты, которые рекомендуется добавить в систему (из списка **Оптимальные агрегаты**), и агрегаты, которые рекомендуется удалить из системы (список **Агрегаты**). Предложенные рекомендации можно не выполнять или выполнять в ограниченном объеме.

5.14.2.7. Разработка структуры регистра накопления

Разработка структуры регистра заключается в создании наборов измерений, ресурсов и реквизитов на закладке **Данные** (см. [здесь](#)).

5.14.2.7.1. Свойства измерения (ресурса, реквизита) регистра накопления

Свойства измерений, ресурсов и реквизитов редактируются при помощи палитры свойств. В основном они совпадают с общими свойствами объектов. Ниже в этом разделе будут описаны уникальные свойства измерений, ресурсов и реквизитов.

Тип данных. В отличие от измерений и реквизитов, объекты вида **Ресурс** могут хранить только один тип данных – **Число**.

Запрет незаполненных значений – установка этого флажка для измерения включает механизм запрета записи движений регистра с пустым значением измерения.

Индексировать – данное свойство можно установить только для измерений. Установка свойства позволяет ускорить выполнение операций, обращающихся к данным регистра, например, если выбираются движения по конкретному значению данного измерения. К таким операциям относятся запросы с условием равенства данного измерения указанному значению, а также временный расчет и методы обхода движений объекта **РегистрНакопления** встроенного языка, использующие отбор по определенному значению измерения.

Использование в итогах – если свойство не установлено, измерения исключаются из хранимых итогов регистра (свойство используется только для измерений оборотного регистра). Если такое измерение используется в запросе или в условии виртуальной таблицы, то виртуальная таблица не будет использовать хранимые итоги, а будет рассчитывать данные только по таблице движений.

5.14.2.7.2. Упорядочивание списка измерений регистра накопления

Порядок расстановки измерений регистра накопления влияет на оптимизацию доступа к итогам регистра: измерения, к которым необходим быстрый доступ, следует располагать в начале списка измерений.

Глава 6. Командный интерфейс

Командный интерфейс – это основное средство доступа пользователя к функциональности приложения, средство, которое позволяет перемещаться между формами и выполнять те или иные действия. Разработчик конфигурации не прописывает детально команды глобального интерфейса и командных панелей форм для каждой реализуемой в приложении роли пользователя (или комбинации таких ролей), а описывает те правила, по которым командный интерфейс будет автоматически формироваться для пользователя.

Формирование командного интерфейса выполняется в зависимости от прав пользователя (см. [здесь](#)), функциональных опций приложения (см. [здесь](#)) и, наконец, настроек самого пользователя.

Такое декларативное описание командного интерфейса помогает решить задачу согласования предоставленных пользователю команд и прав пользователя на выполнение тех или иных действий, а также задачу модификации командного интерфейса приложения при неполном внедрении возможностей приложения.

6.1. Общее устройство командного интерфейса

6.1.1. Разделы и подразделы основного окна приложения

Все глобальные команды основного окна приложения объединяются в разделы. Переход между ними осуществляется при помощи панели разделов. Состав этих разделов однозначно определяется составом подсистем верхнего уровня, для которых установлен признак [Включать в командный интерфейс](#).

Команды текущего раздела отображаются в панели навигации и панели действий основного окна.

При отображении команд того или иного раздела в панели навигации возможно появление одного или нескольких «подразделов», каждый из которых – это совокупность команд, соответствующих подчиненной подсистеме (также с признаком [Включать в командный интерфейс](#)). Например, в разделе [Торговый учет](#) возможно появление подразделов [Розничная торговля](#) и [Оптовая торговля](#) за счет существования соответствующих подчиненных подсистем.

6.1.2. Виды команд

Командный интерфейс пользователя можно разделить:

- на независимые глобальные команды,
- параметризуемые глобальные команды,
- локальные команды формы.

В рамках данного раздела мы будем рассматривать только глобальные команды (описание локальных команд формы см. [здесь](#)). С другой стороны, команды делятся:

- на стандартные команды (автоматически добавляемые системой в командный интерфейс);
- навигационные команды;
- команды действия;
- команды, созданные в конфигурации.

6.1.2.1. Стандартные команды

Для большинства объектов конфигурации система предоставляет стандартные команды, которые автоматически помещаются в командный интерфейс. Ниже приведен список таких объектов и стандартные команды, которые для них предоставляет система:

- Общая форма:
 - Открыть общую форму;
- Константы:
 - Открыть форму редактирования константы;
- Справочники:
 - Открыть форму списка;
 - Открыть форму нового;
 - Открыть форму новой группы;
 - Ввод на основании;
 - Перейти к списку с отбором по владельцу;
- Документы:
 - Открыть форму списка;
 - Открыть форму нового;
 - Ввод на основании;
- Журналы документов:
 - Открыть форму списка;
- Отчеты:
 - Открыть основную форму;
- Обработки:
 - Открыть основную форму;
- План видов характеристик:

- Открыть форму списка;
- Открыть форму нового;
- Открыть форму новой группы;
- Ввод на основании;
- Планы счетов:
 - Открыть форму списка;
 - Открыть форму нового;
 - Ввод на основании;
- Планы видов расчета:
 - Открыть форму списка;
 - Открыть форму нового;
 - Ввод на основании;
- Перечисления:
 - Открыть форму списка;
- Регистры сведений:
 - Открыть форму списка;
 - Открыть форму нового;
 - Перейти к списку с отбором по регистратору;
- Регистры накопления:
 - Открыть форму списка;
 - Перейти к списку с отбором по регистратору;
- Регистры бухгалтерии:
 - Открыть форму списка;
 - Перейти к списку с отбором по регистратору;
- Регистры расчета:
 - Открыть форму списка;
 - Перейти к списку с отбором по регистратору;
- Бизнес-процессы:
 - Открыть форму списка;
 - Открыть форму нового;
 - Ввод на основании;
 - Перейти к списку задач с отбором по бизнес-процессу;
- Задачи:
 - Открыть форму списка;
 - Открыть форму нового;
 - Ввод на основании;
- Планы обмена:
 - Открыть форму списка;
 - Открыть форму нового;
 - Ввод на основании;
- Критерии отбора:
 - Открыть форму списка.

6.1.2.1.1. Формирование и размещение стандартных команд

Стандартные команды открытия формы списка и создания нового элемента формируются всегда, если не выключено свойство [Использовать стандартные команды](#).

Стандартная команда отчета формируется, если у отчета задана основная схема компоновки данных или задана основная или дополнительная форма.

Стандартная команда обработки формируется, если у обработки задана основная или дополнительная форма.

Стандартные команды для ввода на основании создаются системой в том случае, если соответствующим образом задано свойство [Ввод на основании](#). Например, если

для справочника [Товары](#) и для справочника [Партии товаров](#) сказано, что на их основании возможен ввод документов [Приходная накладная](#) и [Расходная накладная](#), то у справочников появится стандартная команда ввода на основании.

Стандартные команды для открытия формы списка с отбором по владельцу формируются в том случае, если соответствующим образом задано свойство справочника [Владельцы](#), а для регистра сведений есть одно или несколько измерений с признаком [Ведущее](#).

Стандартная команда формы списка с отбором по регистратору формируется для регистров, подчиненных регистратору.

Стандартные команды размещаются следующим образом:

Панель	Размещаемые команды
Панель навигации	Команды открытия списков
Панель действий	<ul style="list-style-type: none"> Команда открытия формы редактирования констант. Команды открытия форм новых объектов. Команды открытия форм отчетов и обработок
Панель навигации формы	Команды открытия формы списка с отборами
Командная панель формы	Команды ввода на основании

6.1.2.1.2. Параметризуемые стандартные команды

Некоторые стандартные команды являются параметризуемыми, т. е. могут быть выполнены в контексте той или иной формы, получив в качестве параметра некоторое значение.

Команда	Тип параметра
Ввод на основании	Ссылка на объект-основание
Открытие списка с отбором по владельцу	Ссылка на объект-владелец
Открытие списка с отбором по регистратору	Ссылка на документ-регистратор
Открытие списка критерия отбора	Ссылка на значение критерия отбора

О порядке формирования типа параметра стандартных параметризуемых команд см. [здесь](#).

Так, если для справочника [Товары](#) и для справочника [Партии товаров](#) сказано, что на их основании возможен ввод документа [Приходная накладная](#), то тип параметра команды ввода на основании будет составным: [СправочникСсылка.Товары](#) и [СправочникСсылка.ПартииТоваров](#). Поэтому стандартная команда [Ввод приходной накладной на основании](#) будет автоматически размещена в формах элементов справочников товаров и партий товаров.

6.1.2.2. Независимые и параметризуемые глобальные команды

Независимые глобальные команды предназначены для выбора пользователем той или иной функциональности в рамках приложения в целом. Выполнение такой команды не требует дополнительной информации (параметров). Это, например, такие команды, как:

- открытие списка справочника,
- открытие журнала документов,
- открытие формы того или иного отчета,
- открытие формы нового элемента справочника и т. д.

Параметризуемые глобальные команды зависят от контекста выполнения и не могут быть выполнены без получения дополнительной информации (параметра выполнения команды). Это, например, такие команды, как:

- открытие списка подчиненного справочника (параметр – ссылка на элемент справочника-владельца);
- открытие списка записей регистра, подчиненного регистратору (параметр – ссылка на документ-регистратор);
- ввод одного объекта на основании другого (параметр – объект, служащий «основанием»).

Параметризуемые команды могут отображаться в панели навигации вспомогательного окна и непосредственно в командной панели формы.

Глобальные команды отображаются в командной панели формы перед командой [Открыть справку](#). При этом команды группы [Важное](#) размещаются непосредственно в командной панели, а остальные группы команд (стандартная группа [Создать на основании](#) и другие группы категории [Командная панель формы](#)) размещаются в виде подменю. Выполнение глобальных команд, размещенных в командной панели формы, приводит к открытию нового вспомогательного окна приложения.

6.1.2.3. Навигационные команды и команды действия

Навигационными будем называть те команды, которые предназначены для перехода пользователя к очередной форме приложения, не покидая текущее (основное или вспомогательное) окно приложения. Навигационная команда открывает очередную форму в том же окне приложения, в котором команда была вызвана пользователем.

Навигационными могут быть как независимые команды глобального командного интерфейса, так и параметризуемые глобальные команды. Навигационные команды размещаются в панели навигации основного или вспомогательного окна приложения.

Примерами навигационных команд основного окна приложения могут служить команды перехода к списку справочника или журналу документов. Например, команда [Валюты](#) откроет в том же окне форму списка валют, а команда [Финансовые документы](#) откроет в том же окне форму списка документов.

Примерами навигационных команд вспомогательного окна приложения могут служить команды перехода к спискам, логически подчиненным тому объекту, который редактируется в данном окне. Например, в форме редактирования элемента справочника валют может быть команда перехода к регистру сведений с историей изменения курса валюты; в форме документа может быть команда перехода к его движениям по тому или иному регистру и т. д.

ПРИМЕЧАНИЕ. Отказ от открытия формы не прерывает выполнение навигационной команды. Использование отказа от открытия формы в навигационных командах приведет к открытию пустой формы.

Командами **действия** будем называть те команды, выполнение которых, как правило, приводит к открытию нового вспомогательного окна приложения. Такие команды на некоторое время переключают пользователя на выполнение другой задачи, т. е. существенно изменяют контекст его деятельности. Например, команда создания нового документа переводит пользователя от задачи навигации по приложению, выполняемой в основном окне приложения, к задаче ввода нового

документа.

Такие команды размещаются в панели действий основного окна приложения или в командной панели формы, отображаемой во вспомогательном окне приложения.

6.1.2.4. Команды, созданные в конфигурации

Кроме стандартных команд разработчик конфигурации может создать свои собственные команды, установить для них место размещения (группу команд), описать на встроенном языке действие, выполняемое при выполнении команд, и т. д.

Более подробно о свойствах объекта *Команда* см. [здесь](#) и см. [здесь](#).

6.1.3. Группы команд

Все глобальные команды по месту своего размещения и по своему характеру делятся на четыре категории.

Категория	Описание
Панель навигации	Для размещения независимых навигационных команд
Панель навигации формы	Для размещения параметризуемых навигационных команд, вызываемых из формы
Панель действий	Для размещения команд, приводящих к появлению нового вспомогательного окна приложения
Командная панель формы	Для размещения в форме параметризуемых команд, приводящих к появлению нового вспомогательного окна приложения

Приведенные здесь категории представляют собой перечень тех мест интерфейса приложения, в которых могут отображаться глобальные команды.

Для группировки глобальных команд система реализует стандартные группы команд.

Место размещения	Стандартные группы команд
Панель навигации	<ul style="list-style-type: none"> Важное, Обычное, См. также
Панель действий	<ul style="list-style-type: none"> Создать, Отчеты, Сервис
Панель навигации формы	<ul style="list-style-type: none"> Важное, Перейти, См. также
Командная панель формы	<ul style="list-style-type: none"> Важное, Создать на основании

Кроме того, при разработке конфигурации разработчик может создать собственные группы команд (объект конфигурации *Общие – Группы команд*, см. [здесь](#)), которые можно отнести к одной из перечисленных выше категорий (свойство *Категория* группы команд). Эти группы могут наряду с предопределенными группами использоваться для размещения в них разрабатываемых команд.

6.2. Построение глобального командного интерфейса

В данном разделе рассмотрены объекты и свойства объектов конфигурации, влияющие на построение глобального командного интерфейса конфигурации.

6.2.1. Подсистемы

Основа формирования глобального командного интерфейса основного окна приложения – структура подсистем конфигурации. Именно подсистемы формируют представление пользователя о функциональности приложения в целом. Структура подсистем описывает для пользователя общую функциональность системы. Таким образом, построение глобального командного интерфейса основного окна приложения «от структуры подсистем» налагает на разработчика определенную ответственность при разработке подсистем конфигурации. Фактически структура подсистем – это первое, что увидит пользователь при ознакомлении с приложением.

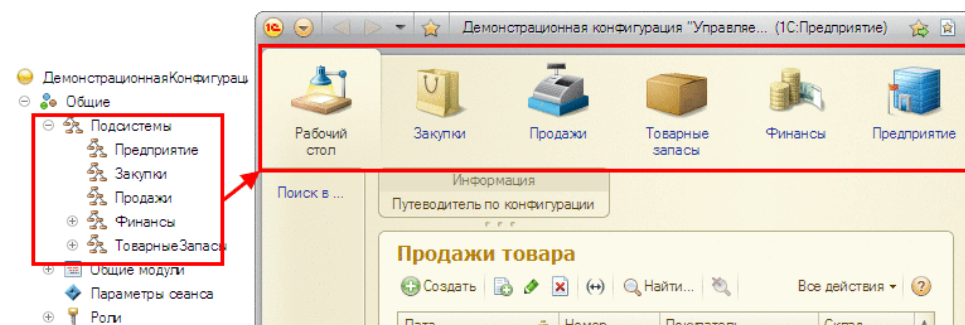


Рис. 148. Отображение подсистем в разделах

На структуру командного интерфейса влияют подсистемы, которым установлено свойство *Включать в командный интерфейс*. Но данное свойство устанавливается подсистемам по умолчанию, и, таким образом, предполагается, что подсистемы создаются в первую очередь именно для описания глобального командного интерфейса.

ПРИМЕЧАНИЕ. Если в конфигурации нет ни одной подсистемы с установленным свойством *Включать в командный интерфейс*, то панель разделов не отображается в основном окне приложения.

Подсистемы первого уровня приводят к появлению в приложении **разделов** – совокупности глобальных команд определенной предметной направленности. Разделы отображаются в панели разделов основного окна приложения. Выбор того или иного раздела изменяет состав команд панели навигации и панели действий.

Подсистемы более низкого уровня приводят к появлению в панели навигации **подразделов**, в которые собираются команды соответствующей подсистемы.

Принадлежность объекта к тому или иному набору подсистем определяет появление команд объекта в соответствующих фрагментах командного интерфейса конфигурации. Однако новые объекты по умолчанию не принадлежат ни одной подсистеме. Это означает, что команды объекта отсутствуют в командном интерфейсе. Поэтому для помещения стандартных и созданных разработчиком команд объекта в соответствующие фрагменты командного интерфейса необходимо указать, каким подсистемам принадлежит этот объект.

Принадлежность объекта к разным подсистемам является независимой, т. е. объект можно отнести одновременно как к «родительской» и «подчиненной» подсистемам, так и только к «подчиненной», если это целесообразно из соображений формирования командного интерфейса.

ПРИМЕЧАНИЕ. Если в конфигурации нет ни одной подсистемы с установленным свойством [Включать в командный интерфейс](#), то на рабочий стол попадают все команды, которые на нем можно было разместить вручную. В этой ситуации меняется внешний вид редактора командного интерфейса рабочего стола: в нем **исчезает дерево команд и пропадает возможность удалить команду с рабочего стола.**

Как только появляется первая подсистема, включенная в командный интерфейс, автоматическое добавление команд на начальную страницу перестает действовать и команды нужно размещать явно.

ПРИМЕЧАНИЕ. Режим автоматического размещения на начальной странице всех команд не действует, если у конфигурации установлен режим запуска [Обычный](#).

При формировании панели разделов используются все подсистемы первого уровня, которые соответствуют всем нижеперечисленным условиям:

1. Свойство [Включать в командный интерфейс](#) установлено в значение *Истина*;
2. Права доступа (см. [здесь](#)) обеспечивают доступность подсистемы для текущего пользователя;
3. Подсистема не отключена функциональными опциями;
4. Подсистема не отключена пользователем с помощью диалога настройки панели разделов.

В том случае, если в разделе все команды недоступны текущему пользователю (в силу отсутствия прав доступа или команды отключены функциональными опциями или пользовательскими настройками), такой раздел все равно будет отображаться в панели разделов.

6.2.2. Команды

Как говорилось выше, существуют стандартные команды, предоставляемые системой автоматически, и команды, создаваемые разработчиком в конфигурации.

Объект конфигурации *Команда* может быть создан как подчиненный объект для следующих объектов конфигурации:

- Справочники,
- Документы,
- Журнал документов,
- Отчеты,
- Обработки,
- Планы видов характеристик,
- Перечисления,
- Планы счетов,
- Планы видов расчета,
- Регистры сведений,
- Регистры накопления,
- Регистры бухгалтерии,
- Регистры расчета,
- Бизнес-процессы,
- Задачи,
- Планы обмена,
- Критерии отбора.

Кроме того, возможно создание общих команд (ветка дерева конфигурации [Общие – Общие команды](#), см. [здесь](#)).

6.2.3. Параметризация команды

Если команда является параметризуемой, то ее выполнение предполагает получение некоторого значения в качестве параметра. Источником такого значения могут служить только данные формы.

ВНИМАНИЕ! Задавать тип значения параметра команды имеет смысл только для тех команд, которые размещаются в формах (относятся к группам категорий [Командная панель формы](#) или [Панель навигации формы](#)).

Свойство [Тип параметра](#) команды фактически задает состав форм, в контексте которых возможно выполнение данной команды. Например, если команда имеет тип параметра [СправочникСсылка.Товары](#), то она может быть выполнена:

- в форме элемента справочника [Товары](#), где параметром послужит ссылка на редактируемый в форме объект;
- в форме списка справочника [Товары](#), где параметром послужит ссылка из текущей строки списка;
- в форме приходной накладной, где параметром послужит ссылка на товар из текущей строки табличной части документа и т. д.

Количество значений выбранного типа, которые будут переданы команде в качестве значений параметра, определяет свойство [Режим использования параметра](#). Если

это свойство установлено в значение **Одиночный**, то в команду передается одно значение указанного типа.

Если свойство установлено в значение **Множественный**, то в качестве параметра всегда передается массив значений (даже если выбрано одно значение). Этот режим имеет смысл выбирать тогда, когда источником данных для команды может являться таблица в режиме множественного выделения. В этом случае первым элементом массива будет выступать текущая строка (вне зависимости от последовательности выделения строк табличного поля). Так, если мы имеем список товаров, состоящий из элементов **Миксер**, **Пылесос**, **Холодильник**, **Чайник**, в котором выделены элементы **Миксер**, **Холодильник** и **Чайник**, а текущей строкой является **Холодильник**, то команда получит массив следующего содержания (по порядку элементов): **Холодильник**, **Миксер**, **Чайник**. Порядок элементов после первого не определен, несмотря на некоторый порядок, приведенный в примере.

ПРИМЕЧАНИЕ. Если текущая строка не входит в выделение, то первый элемент в массиве невозможно идентифицировать однозначно.

При размещении на форме нескольких таблиц управляемой формы, связанных с одним реквизитом формы типа **ДинамическийСписок**, неопределено, данные какой из этих таблиц будут использоваться для параметризации команды.

Формирование фрагментов командного интерфейса форм выполняется автоматически на основании типа основного реквизита. При редактировании формы возможно дополнение фрагмента командного интерфейса разработчиком формы. Например, если форма предназначена для редактирования товара, то в нее автоматически попадут все команды, тип параметра которых – **СправочникСсылка.Товары** (например, **Печать карточки товара**, **Переход к списку цен товара** и т. д.). Если товар имеет реквизит **Производитель**, то при разработке формы в нее также можно будет включить и команды с типом параметра **СправочникСсылка.Контрагент** (например, **Карточка контрагента**, **Договора контрагента** и т. д.), так как в форме можно получить значение для параметра таких команд. Но такие команды (которые не относятся к основному реквизиту) автоматически не включаются в командный интерфейс, однако разработчик может включить их с помощью редактора формы.

Для группы справочника (вид иерархии **Иерархия групп и элементов**) и иерархического плана видов характеристик не происходит автоматического включения в командный интерфейс формы группы параметризуемых глобальных команд, однако разработчик может включить такие команды в командный интерфейс с помощью редактора формы.

6.2.4. Формирование командного интерфейса по умолчанию

На место размещения команд влияют свойства **Тип параметра команды**, **Группа** и принадлежность команды (или родительского объекта) к той или иной подсистеме.

Независимые (т. е. непараметризуемые) команды относятся к фрагментам глобального командного интерфейса, соответствующим подсистемам, в которые включен объект-владелец или сама команда (для общих команд).

Параметризуемые команды относятся к фрагментам глобального командного интерфейса, соответствующим тем объектам, типы которых заданы в свойстве **Тип** параметра команды.

Группа команд и в первом, и во втором случае – это своеобразное уточнение размещения команды в пределах одного фрагмента. Так, например, группы команд задают, в какой панели (навигации или действий) будут размещены независимые команды того или иного раздела. А для параметризуемых команд группа команд определяет, будет ли команда размещена в панели навигации формы или в командной панели формы.

Таким образом, для независимых команд (команд, не требующих параметра, таких как открытие списка или открытие формы нового объекта) возможность размещения по умолчанию в категории иной, чем панель навигации или панель действий, востребована крайне редко. В свою очередь, размещение параметризуемых команд в панели навигации или панели действия не имеет смысла, так как в контексте основного окна приложения такие команды не могут получить параметры, необходимые для их исполнения. Приведем таблицу расположения команд.

	Независимые команды	Параметризуемые команды
Навигационные команды	Панель навигации	Панель навигации формы
Команды действия	Панель действий	Командная панель формы

Следует помнить, что по умолчанию глобальные команды не попадают в командный интерфейс форм групп иерархических справочников и планов видов характеристик. Для того чтобы глобальная команда была доступна в командном интерфейсе такой формы, ее необходимо разместить там вручную, с помощью редактора формы.

При формировании командного интерфейса используются текстовые представления объектов конфигурации и различных команд. По умолчанию система формирует представления исходя из свойств **Синоним** и **Имя**. Однако есть возможность влиять на это представление. Для этого используются представления объекта и списка. Подробнее об этом см. [здесь](#).

6.2.5. Свойство «Командный интерфейс»

Свойства **Командный интерфейс** (для подсистем), **Командный интерфейс** (для конфигурации) и **Командный интерфейс основного раздела** (для конфигурации) предназначены для редактирования состава глобальных команд соответствующего раздела или состава команд основного раздела. Редактирование этих свойств выполняется в том случае, если не устраивает автоматически сформированный порядок команд, группы, к которым отнесены команды в контексте данного раздела, и установленная автоматически видимость по умолчанию.

В разных разделах одна и та же команда может отображаться в разных местах панели навигации или панели действия и может иметь разную видимость по умолчанию. Например, команда открытия списка справочника **Товары** с точки зрения раздела **Торговля** является важной и часто используемой, и разработчик может поместить ее в группу **Важное**. А с точки зрения раздела **Бухгалтерский учет** не так важна, и разработчик может поместить ее в группу **См. также**.

Подробнее о редакторе фрагмента командного интерфейса см. [здесь](#) и см. [здесь](#).

6.2.6. Редактирование состава команд

Кроме возможности редактирования свойства подсистемы **Командный интерфейс** при помощи редактора этого свойства существует возможность редактирования состава команд сразу всех разделов в редакторе **Все подсистемы** (см. [здесь](#)).

Редактирование состава глобальных команд, отображаемых в форме, выполняется в редакторе формы, на закладке **Командный интерфейс** (см. [здесь](#)).

6.2.7. Ролевая настройка видимости команд по умолчанию

При автоматическом формировании команд разделов и команд форм устанавливается следующая видимость по умолчанию.

По умолчанию **видимы**:

- команды открытия списков справочников, документов, журналов документов, планов видов характеристик, планов видов расчета, планов счетов, независимых регистров сведений, бизнес-процессов, задач и планов обмена;
- команды открытия форм отчетов и обработок;
- команда открытия формы редактирования констант;

- команды перехода к списку подчиненного справочника;
- команды перехода к списку логически подчиненного регистра сведений (т. е. регистра с «ведущими» измерениями);
- команды ввода на основании;
- произвольные команды, созданные разработчиком при конфигурировании.

По умолчанию **невидимы**:

- команды открытия списков перечислений, регистров сведений, подчиненных регистратору, регистров накоплений, регистров бухгалтерии, регистров расчета;
- команды открытия форм нового объекта и новой группы;
- команды перехода к спискам подчиненных регистров;
- команды перехода к спискам критериев отбора.

Эти значения видимости, формируемые системой автоматически, могут быть изменены как для параметризуемых, так и для независимых команд. Причем значение видимости по умолчанию может быть задано в разрезе ролей, применяемых в конфигурации. При отображении команд в режиме 1С:Предприятие команда будет видна по умолчанию, если хотя бы для одной из ролей пользователя задана видимость для данной команды.

При редактировании ролевой видимости следует исходить из того, что видимость команды для той или иной роли вступает в силу только тогда, когда команда доступна для этой роли. Таким образом, при настройке видимости для роли, сильно ограниченной в правах, почти никогда нет необходимости дополнительно скрывать команды, снимая им видимость по умолчанию (общее число доступных команд в том или ином разделе и так невелико).

Ролевое редактирование видимости по умолчанию – это средство, позволяющее настроить начальную «насыщенность» глобального командного интерфейса в первую очередь для пользователей с широкими правами доступа.

6.3. Сервисные возможности навигации

6.3.1. Ссылки

В «1С:Предприятии» имеется возможность получить текстовую ссылку на любой раздел командного интерфейса, отчет, обработку и на объекты информационной базы (документы, элементы справочников и т. д.).

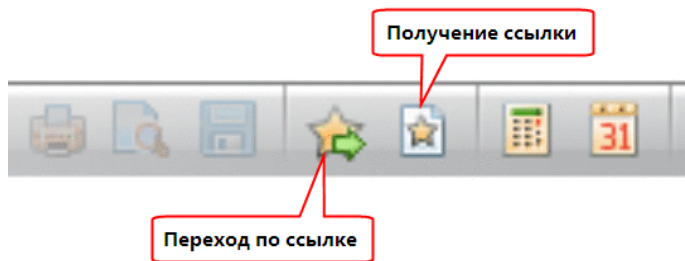


Рис. 149. Команды получения и перехода по ссылке

Полученную ссылку можно сохранить и использовать в дальнейшем для перехода по ней. Так как ссылки являются текстовыми, то их можно отправлять другим пользователям, например, по электронной почте. Описание форматов ссылок см. [здесь](#). Нельзя получить ссылку на инструменты, входящие в состав стандартных функций.

Существует возможность запускать веб-клиент с одновременным переходом по ссылке. Для этого необходимо получить внешнюю ссылку из веб-клиента или тонкого клиента, подключенного к информационной базе через веб-сервер.

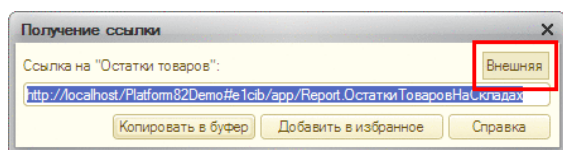


Рис. 150. Получение внешней ссылки

Для получения внешней ссылки следует воспользоваться обычной командой получения ссылки, убедившись, что в окне получения ссылки нажата кнопка **Внешняя** (см. [рис. 150](#)). Полученную ссылку можно ввести в адресной строке браузера, при этом будет загружено приложение и выполнен переход по ссылке (в примере на [рис. 150](#) произойдет открытие формы отчета [Остатки товаров](#)).

ПРИМЕЧАНИЕ. В Microsoft Internet Explorer не происходит загрузка приложения с переходом по ссылке, если ссылка относится к приложению, загруженному в текущем окне веб-браузера. Рекомендуется вводить ссылку в адресную строку нового пустого окна веб-браузера.

6.3.2. Оповещения пользователя

Механизм оповещений предназначен для информирования пользователей о том, что система выполнила то или иное действие. Оповещения могут создаваться системой или разработчиком прикладного решения. Система создает оповещения при интерактивной записи/изменении объекта, а разработчик – вызовом метода `ПоказатьОповещениеПользователя()`.

Оповещение отображается в окне, которое по умолчанию расположено в правом нижнем углу рабочей области экрана. Если с оповещением связана какая-либо ссылка, то пояснение будет одновременно являться гиперссылкой, ее нажатие приведет к открытию объекта, на который указывает ссылка. Пояснение также будет представлено гиперссылкой, если оповещение сформировано системой автоматически (при интерактивной записи/изменении объекта).

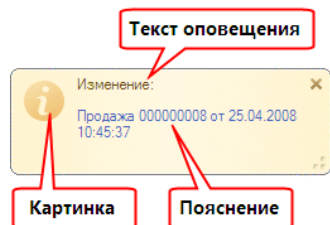


Рис. 151. Оповещение пользователя

Если при вызове метода не указан параметр [Пояснение](#), то переход по навигационной ссылке также будет недоступен. Для того чтобы пользователь имел возможность перейти по гиперссылке, необходимо одновременно указывать и параметр [Навигационная ссылка](#), и параметр [Пояснение](#).

Окно оповещения автоматически исчезает с экрана по истечении небольшого времени. Однако если навести на это окно курсор мыши, то окно будет существовать до тех пор, пока его принудительно не закроют или не уберут курсор мыши с площади окна оповещения.

Если вызвать метод отображения оповещения несколько раз подряд, то пользователь увидит только самое последнее оповещение.

ПРИМЕЧАНИЕ. Окно оповещения располагается поверх всех окон текущего приложения, кроме модальных окон. Модальное окно выводится поверх окна оповещения.

При отображении в панели избранного и истории список оповещений выравнивается по правому краю панели. Наиболее позднее оповещение будет отображаться у правого края панели. Если оповещений больше пяти, то отображаются только пять последних оповещений.

ПРИМЕЧАНИЕ. Список оповещений отображается только в течение жизни сеанса.

Окна оповещений, которые открываются в веб-клиенте, привязаны к текущему активному окну:

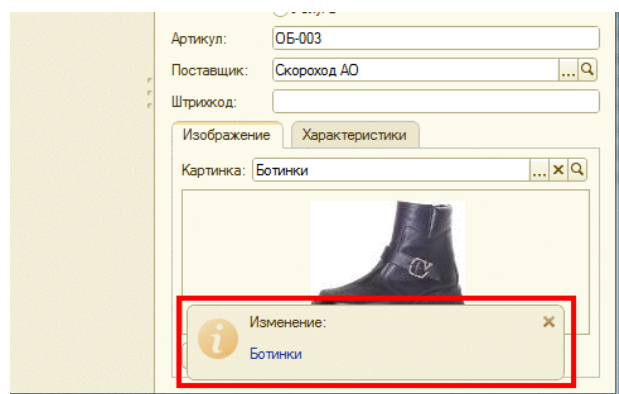


Рис. 152. Оповещение пользователя в веб-клиенте

При закрытии окна, в котором было показано окно оповещения, это окно «передается» родительскому окну и т. д. до достижения основного окна приложения.

ПРИМЕЧАНИЕ. В веб-клиенте нельзя перемещать окно оповещения и изменять его размер.

6.3.3. Отображение состояния длительных процессов

В процессе разработки конфигурации возникают ситуации, когда необходимо информировать пользователя системы о состоянии выполнения длительных процессов (например, расчет зарплаты по подразделению). Для этого предназначена панель состояния.

Панель состояния отображается вызовом метода [Состояние\(\)](#) и недоступна для вызова на стороне сервера. В том случае, если необходимо отобразить на клиенте состояние длительного процесса, протекающего на сервере, нужно реализовать этот процесс таким образом, чтобы он мог выполняться на сервере «порциями», которые будут инициироваться со стороны клиента. Тогда одновременно со стартом очередной «порции» на сервере можно будет отображать изменение состояния выполнения процесса. Панель состояния отображается в окне, которое по умолчанию расположено в правом нижнем углу рабочей области экрана.

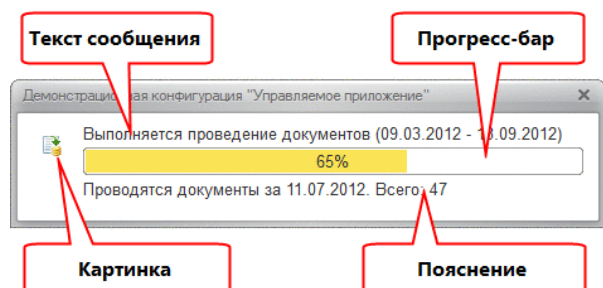


Рис. 153. Окно отображения состояния

Если требуется отображать ход выполнения процесса с помощью индикатора (например, мы знаем общее количество рассчитываемых сотрудников и хотим показать, сколько сотрудников уже рассчитано), то при вызове метода [Состояние\(\)](#) необходимо указывать третий параметр метода, который определяет абсолютное значение индикатора прогресса. Минимальное значение индикатора прогресса всегда равно 0, а максимальное значение – 100.

[Копировать в буфер обмена](#)

`Состояние("Выполняется обработка данных", Счетчик*10, "Обрабатывается порция: " + Счетчик, БиблиотекаКартинok.ПодсистемаТоварныеЗапасы);`

Если при вызове метода третий параметр не указан, то индикатор прогресса также не будет отображаться на панели состояния. При этом текст пояснения будет располагаться непосредственно под текстом основного описания:

Состояние ("Выполняется проведение за " + `Формат(ДатаПроведения, "ДФ=DD")` , , "Проведено " + `КоличествоПроведенных` , `БиблиотекаКартинок.Провести`);

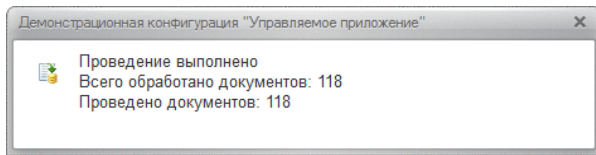


Рис. 154. Состояние без прогресса

Панель состояния автоматически исчезает с экрана по истечении небольшого времени после окончания работы фрагмента программы, вызвавшего появление панели состояния. Однако если навести на это окно курсор мыши, то окно будет существовать до тех пор, пока его принудительно не закроют или не уберут курсор мыши с площади панели состояния.

ПРИМЕЧАНИЕ. Панель состояния располагается поверх всех окон текущего приложения, кроме модальных окон. Модальное окно выводится поверх панели состояния.

В веб-клиенте панель состояния реализована в виде отдельного окна веб-браузера. Обновление этого окна имеет особенности:

- **Microsoft Internet Explorer.** Состояние обновляется после каждого вызова метода `Состояние()`.
- **Mozilla Firefox.** Состояние обновляется при каждом серверном вызове и после окончания выполнения программного кода на встроенном языке. Таким образом, если при выполнении программного кода на встроенном языке не было вызовов серверных методов, то панель состояния изменяться не будет.
- **Google Chrome и Safari.** Состояние обновляется только после окончания выполнения программного кода на встроенном языке.

6.3.4. Сообщения

В прикладном решении большинство сообщений логически связано с данными. Например, если при проведении документа на складе не хватает некоторого количества определенной номенклатуры, программист должен уведомить об этом пользователя.

Механизм сообщений позволяет разработчику сформировать сообщение, в котором можно указать, какой из реквизитов объекта стал причиной ошибки. При отображении в клиентском приложении сообщение может быть автоматически привязано к элементу формы, который редактирует этот реквизит, и рядом с ним будет выведено заданное сообщение.

Сообщения выводятся в панель сообщений, которая располагается в нижней части рабочей области. Высота панели сообщений не превышает 5 строк. Если в панели размещается большее количество сообщений – появляется полоса прокрутки, но панель по высоте не увеличивается. Вручную также невозможно изменить высоту панели сообщений.

Для функционирования этого механизма в платформе существует объект `СообщениеПользователя`.

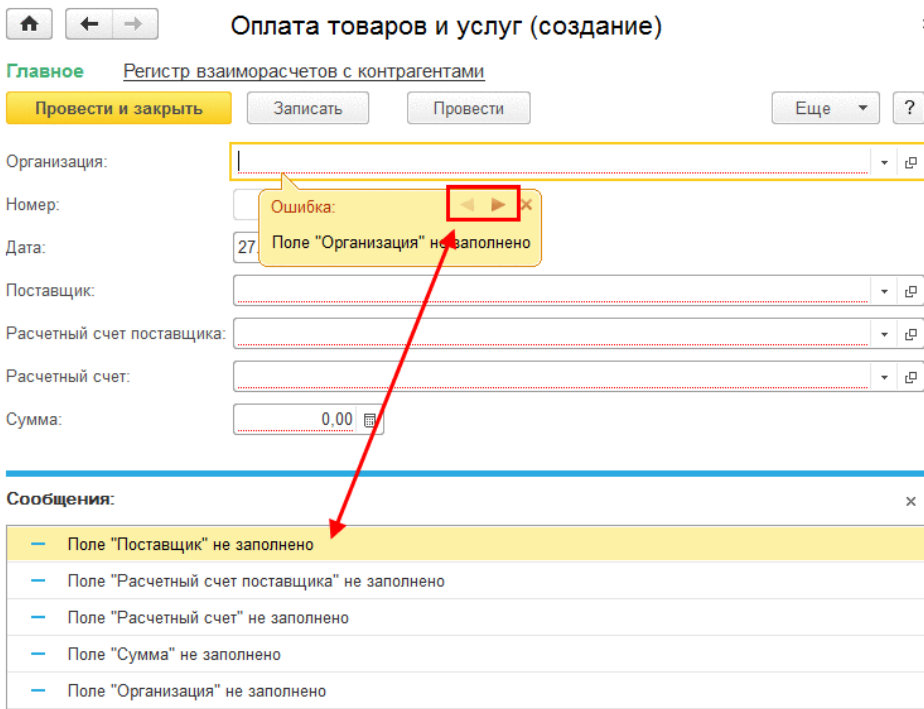


Рис. 155. Сообщения

Механизм сообщений ориентирован на то, чтобы сообщить пользователю, что запрошенное им действие выполнить не удалось. Например, проведение документа невозможно из-за отсутствия остатков товара на складе.

Если сообщение привязано к полю табличной части, то изменение порядка строк табличной части не оказывает влияния на корректность отображения сообщения.

Продажа 00000012 от 04.09.2012 10:20:25 *
×

[Главное](#)
[Регистр взаиморасчетов с контрагентами](#)
[Регистр продаж](#)
[Регистр товарных запасов](#)

Провести и закрыть
Записать
Провести
Еще ▾
?

Номер:
 Вид цен:

Дата:
 Валюта взаиморасчетов:

Организация:

Покупатель:

Склад:

Товары (2) Прочее

Добавить
Подбор
Еще ▾

N	Товар	Цена	Количество	Сумма
1	Колбаса	250,00		
2	Сметана	50,00	0,00	
Количество (итог):				

Сообщение:

Ошибка:
 Не заполнена колонка "Количество" в строке 2 списка "Товары"

Сообщения:

- Не заполнена колонка "Количество" в строке 1 списка "Товары"
- Не заполнена колонка "Сумма" в строке 1 списка "Товары"
- Не заполнена колонка "Количество" в строке 2 списка "Товары"
- Не заполнена колонка "Сумма" в строке 2 списка "Товары"

Рис. 156. Сообщения в табличной части

Если строка, к которой было привязано сообщение, удалена, то сообщение будет привязано к элементу формы (см. [здесь](#)), который отображает табличную часть.

В том случае, если сообщение привязано к строке таблицы, связанной с реквизитом формы типа `ДанныеФормыКоллекция`, `ДанныеФормыДерево` или `ДанныеФормыСтруктураИКоллекция`, в этой таблице пользователем интерактивно выполнен поиск (с помощью соответствующей стандартной команды), в результате поиска строка, для которой создано сообщение, оказалась скрытой, то поиск будет автоматически отменен и сообщение будет показано в нужном месте. Если в таблице установлен программный отбор (свойство `ОтборСтрок`), то такой отбор не отменяется при отображении сообщения и выполняется позиционирование сообщения на таблицу в целом.

ПРИМЕЧАНИЕ. Рекомендуется выводить сообщения только в том случае, если запрошенное пользователем действие не может быть выполнено, а не для информирования его о выполняемых действиях. Для информирования о выполняемом действии можно использовать метод `ПоказатьОповещениеПользователя()`.

Чтобы сообщение об ошибке оказалось привязанным к нужному элементу формы, системе нужно знать следующую информацию: какой реквизит, какого объекта информационной базы, каким элементом формы отображается. Заполнение вышеописанных свойств позволяет ответить на все вопросы:

- Задавая значение свойства `ПутьКДанным`, мы указываем, какой реквизит формы хранит данные нашего объекта информационной базы.
- Задавая свойство `Поле`, мы указываем реквизит объекта, для которого будет отображаться сообщение. Можно сказать, что в сообщении пойдет речь о данных `ПутьКДанным.Поле`, которые в базе данных идентифицируются значением `КлючКДанным`. Далее система автоматически (на основании знаний о структуре формы) определит элемент формы, в котором отображается необходимая информация. Затем в привязке к этому элементу формы будет отображено сообщение.

Свойство `Поле` может содержать описание данных в следующем формате:

- для реквизитов – строка вида `ИмяРеквизита`, например, `Поставщик.Поставщик.Родитель`;
- для реквизитов табличных частей – строка вида `ИмяТабличнойЧасти[ИндексСтроки].ИмяРеквизита`, например, `Номенклатура[10].Количество`;
- для строк наборов записей – строка вида `[ИндексСтроки].ИмяРеквизита`, например, `[10].Валюта`.
- Свойство `ИдентификаторНазначения` описывает, к какой форме должно быть привязано сообщение. Например, если сообщение формируется из формы, отображаемой на начальной странице, указание свойства `ИдентификаторНазначения` позволит отобразить сообщение, привязанное к той форме, из которой выполнялось формирование сообщения.

Если в сообщении свойство `КлючДанных` заполнено и не совпадает с ключевым значением основного реквизита текущей формы (ссылкой объекта или ключом записи регистра сведений), то открывается новая форма объекта, соответствующая значению свойства `КлючДанных`.

- Если у сообщения свойство `ПутьКДанным` не заполнено, то в него проставляется имя основного реквизита открытой формы объекта, вызывается показ сообщения во вновь открытой форме.

Кроме того, выполняется перенос всех сообщений со свойством `КлючДанных`, равным значению этого свойства текущего сообщения, из окна текущей формы в открытую форму. При этом перенесенным сообщениям аналогичным образом проставляется свойство `ПутьКДанным`, если оно не установлено.

Для заполнения пути к данным используется специальный механизм системы, который сохраняет информацию о соответствии элементов формы и объектов, а потом использует ее при вызове метода `УстановитьДанные()`. Таким образом, до выполнения процедур модуля объекта, в которых формируются сообщения, должно быть установлено соответствие между реквизитами формы и объектами, а в самом модуле разработчику нужно вызывать метод `УстановитьДанные()`.

При выполнении стандартных действий с объектом в форме, например, при проведении документа, расширение формы само создает объект и само устанавливает соответствие. Поэтому в этих случаях разработчику не нужно предпринимать никаких действий.

ПРИМЕЧАНИЕ. Если при записи нового объекта (плана обмена, справочника, документа, элемента плана счетов, элемента плана видов характеристик, элемента

плана видов расчета, бизнес-процесса, задачи) в транзакции формировались сообщения пользователю, то при отмене транзакции в сообщениях очищаются ссылки на данный объект (свойство `КлючДанных`):

Ниже описывается, как установить соответствие, если объект создается из данных формы программно.

Связь между объектом и формой может быть задана явно, путем вызова метода, или неявно. Для явной установки соответствия используется метод `УстановитьСоответствиеОбъектаФормы()`. В качестве второго параметра в метод передается информация о том, в какой форме отображается объект (свойство `ИдентификаторФормы`) и в каком реквизите указанной формы хранится объект данных (свойство `Реквизит`) для последующей установки правильного соответствия сообщений и элементов управления формы:

```

Параметры = Новый Структура;
Параметры.Вставить("ИдентификаторФормы", УникальныйИдентификатор);
Параметры.Вставить("Реквизит", "Объект");
УстановитьСоответствиеОбъектаИФормы(Объект, Параметры);

```

[Копировать в буфер обмена](#)

Установленное соответствие будет существовать до тех пор, пока существует объект, для которого установлено соответствие.

Для того чтобы сообщение использовало данные соответствия объекта и имени реквизита, необходимо связать сообщение с данными через метод `УстановитьДанные()` объекта `СообщениеПользователю`.

[Копировать в буфер обмена](#)

```

ОбъектСправочник = СсылкаСправочник.ПолучитьОбъект();
Параметры = Новый Структура;
Параметры.Вставить("ИдентификаторФормы", ИдентификаторФормы);
Параметры.Вставить("Реквизит", "Объект");
УстановитьСоответствиеОбъектаИФормы(ОбъектСправочник, Параметры);
ОбъектСправочник.ПроверитьЗаполнение();

```

Нужно понимать, что в методе `УстановитьСоответствиеОбъектаИФормы()` указывается имя реквизита для той формы, которая будет открыта по умолчанию для отображения элемента используемого справочника (в примере `ОбъектСправочник`). Тогда сообщения, которые создаются в обработчике проверки заполнения (`ОбъектСправочник.ПроверитьЗаполнение()`), будут корректно привязаны к реквизитам новой формы. Если свойство `ИдентификаторФормы` структуры `Параметры` (в примере выше) содержит уникальный идентификатор, который не соответствует ни одному экземпляру формы, то для отображения сообщения будет создана и открыта новая форма.

Получить соответствие можно методом `ПолучитьСоответствиеОбъектаИФормы()`. Получить соответствие можно до тех пор, пока существует объект, для которого оно установлено.

[Копировать в буфер обмена](#)

```

ОписаниеСоответствия = ПолучитьСоответствиеОбъектаИФормы(Объект);

```

Если для переданного объекта существует соответствие с реквизитом, имя реквизита будет возвращено как результат работы функции. Кроме того, соответствие может быть установлено формой при использовании метода `РеквизитФормыВЗначение()`. В большинстве случаев рекомендуется использовать именно этот способ.

[Копировать в буфер обмена](#)

```

// Код в модуле документа.
&НаСервере
Процедура ОбработкаПроведения()
...
Сообщение = Новый СообщениеПользователю();
Сообщение.Текст = "В строке 10 табличной части " + "Номенклатура не хватает " + НедостажееКоличество + " " + ЕдиницаИзмеренияНоменклатуры;
Сообщение.Поле = "Номенклатура[9].Количество";
Сообщение.УстановитьДанные(ЭтотОбъект);
Сообщение.Сообщить();
// Сообщение будет показано в форме и привязано к
// элементу управления, связанному с полем
// Количество в 10-й строке табличной части
// Номенклатура.
...
КонецПроцедуры;

```

В случае если создание сообщений пользователю (с помощью объекта `СообщениеПользователю`) выполняется во время контекстного или неконтекстного серверного вызова из формы, вызова из общего модуля или из модуля общей команды, вывод сообщений блокируется. Созданные сообщения будут показаны пользователю после возврата управления на клиентскую сторону. Для получения списка непоказанных сообщений используется метод `ПолучитьСообщенияПользователю()`.

Если в поле ввода, с которым связано сообщение, находится некорректное значение, то в веб-клиенте можно перейти в окно сообщений. В других клиентских приложениях такой переход выполнить невозможно.

6.3.5. Работа сочетаний клавиш

В независимом вспомогательном окне срабатывают сочетания клавиш:

- команд панели навигации главного окна;
- команд панели действий главного окна;
- команд самого вспомогательного окна (как команд формы, так и команд панели навигации).

При этом сочетания клавиш команд основного окна срабатывают, даже если команда, вызываемая тем или иным сочетанием клавиш, невидима, например, была скрыта пользователем. Во вспомогательных окнах не срабатывают сочетания клавиш скрытых команд панели навигации формы и скрытых элементов формы.

В блокирующих окнах работают только сочетания клавиш самих окон, а сочетания клавиш команд главного окна не поддерживаются.

6.3.6. Работа со списком форм

Система позволяет получить список форм, которые в данный момент отображаются в системе, включая формы, расположенные на начальной странице (рабочем столе). Этот список можно получить с помощью функции глобального контекста `ПолучитьОкна()`. Затем следует проанализировать содержимое полученной коллекции `ОкнаКлиентскогоПриложения` (состоит из объектов типа `ОкноКлиентскогоПриложения`):

- Окно с установленным свойством `Основное` описывает основное окно приложения. Свойство `Содержимое` для такого окна всегда содержит пустой массив.
- Окно с установленным свойством `НачальнаяСтраница` описывает начальную страницу (рабочий стол). При этом свойство `Содержимое` содержит список форм, которые в данный момент размещены на начальной странице (рабочем столе). Более подробное описание программного управления составом форм начальной страницы (рабочего стола) см. [здесь](#).
- Остальные окна описывают вспомогательные окна клиентского приложения. При этом свойство `Содержимое` для такого окна содержит список форм, открытых в этом окне. Основная форма такого окна будет располагаться в первом элементе массива.

Следующий пример отобразит все окна клиентского приложения, выведет их свойства и список форм, находящихся в том или ином окне клиентского приложения (при наличии таковых).

Пример отображения списка окон:

Копировать в буфер обмена

```
Окна = ПолучитьОкна();
Для Каждого Окно Из Окна Цикл
    Сообщить ("Заголовок: " + Окно.Заголовок);
    Сообщить ("Основное окно приложения: " + Окно.Основное);
    Сообщить ("Начальная страница: " + Окно.НачальнаяСтраница);
    Сообщить ("Форм в окне: " + Окно.Содержимое.Количество());
    Если Окно.Содержимое.Количество() Тогда
        Для каждого Форма Из Окно.Содержимое Цикл
            Сообщить ("    Форма : " + Форма.ИмяФормы);
        КонецЦикла;
    КонецЕсли;
Сообщить ("-----");
КонецЦикла
```

6.4. Порядок разработки командного интерфейса

При разработке командного интерфейса обычно выполняется следующая последовательность действий:

- в начале разработки конфигурации определяется ее структура с точки зрения предметной области, т. е. создается дерево подсистем конфигурации;
- создается состав ролей, т. е. определяется, для кого предназначена данная конфигурация;
- при создании новых объектов конфигурации (справочников, документов и т. д.), как правило, они сразу относятся к тем или иным подсистемам, для объектов назначаются права доступа по ролям;
- если не устраивает стандартное расположение команд и их видимость по умолчанию, то выполняется редактирование свойства **Командный интерфейс** необходимых подсистем и редактирование состава команд тех форм, в которых автоматически размещены параметризованные команды необходимых объектов (перехода к подчиненному списку и ввода на основании).

Уже эти минимальные действия приведут к тому, что в соответствующих разделах командного интерфейса появятся команды открытия форм списков объектов, команды ввода новых объектов (если включена их видимость в редакторе фрагмента командного интерфейса). Кроме того, если это определено в свойствах объекта, в формах других объектов конфигурации появятся команды ввода на основании, перехода к списку по регистратору, перехода к списку по владельцу и т. д.

Рассмотрим пример создания документа **Расход товара**. При его создании мы сделаем следующее:

- укажем, что документ будет относиться к подсистемам **Торговый учет** и **Бухгалтерия**;
- установим ролям **Менеджер по продажам** и **Администратор** права на работу с этим документом;
- укажем, что документ вводится на основании документов **Счет** и **Заказ покупателя**;
- укажем, что он является регистратором для регистра **Остатки товаров на складах**.

Эти действия приведут к появлению команды **Расход товара** в панели навигации раздела **Торговый учет** в том случае, если пользователь обладает ролью **Менеджер по продажам** или административными правами.

Кроме того, в формах документов **Счет** и **Заказ покупателя** появится команда подменю **Создать на основании**, предназначенная для ввода расходной накладной. Нам не понадобилось редактировать интерфейсы для пользователей с такими ролями и формы документов **Счет** и **Заказ покупателя**.

Стандартное представление команды перехода к списку накладных (**Расход товара**) может выглядеть неточным. Для его переопределения заполним свойство **Представление списка** документа **РасходТовара** и укажем там **Продажи**.

Допустим, что нас не устраивает представление объекта по умолчанию (которое участвует в представлении команд) и мы хотим изменить его так, чтобы представление команды ввода на основании читалось следующим образом: **Продажа: создать на основании**. Для этого необходимо заполнить свойство **Представление** объекта документа **Расход товара** текстом **Продажа**.

Теперь обратим внимание на то, что при стандартном размещении команды она находится в середине списка команд раздела **Торговый учет**. Предположим, что команда достаточно важна для этого раздела. Мы переходим к редактору **Все подсистемы** (или к редактору командного интерфейса подсистемы **Торговый учет**) и переносим команду в группу **Важное**. Раздел **Бухгалтерия** при этом не трогаем, оставляя там размещение команды по умолчанию.

Наконец, обратим внимание на видимость команд по умолчанию. В форме документа **Расход товара** появилась команда перехода к списку записей регистра **Остатки товаров на складах**, но она по умолчанию невидима. Предположим, что эта команда важна для администратора системы и действительно по умолчанию не нужна менеджеру. Для того чтобы это отразить, создадим форму документа **Расход товара**, перейдем к редактированию командного интерфейса и для команды перехода к списку регистра установим видимость для роли **Администратор**, а видимость для менеджера менять не будем.

Для администратора системы с его широкими правами доступа число команд оказалось слишком большим. Предположим далее, что командой перехода к списку расходных накладных он будет пользоваться крайне редко. Для того чтобы облегчить его командный интерфейс, опять перейдем к редактированию команд в редакторе **Все подсистемы** и для нашей команды снимем видимость для роли **Администратор**.

Также можно для менеджера по продажам установить видимость по умолчанию для команды добавления документа **Расход товара** в панели действий, чтобы он смог вызывать ввод этого документа без обращения к списку документов.

В процессе работы с конфигурацией пользователь может настраивать видимость команд по своему усмотрению.

Глава 7. Формы

7.1. Общая информация

Форма – это объект, созданный для ввода и просмотра какой-либо информации, а также для выполнения управления различными процессами. С помощью форм программа запрашивает у пользователя ту информацию, которая необходима ей для дальнейшей работы, либо выдает какую-либо информацию пользователю для просмотра и редактирования.

Основное назначение формы – предоставить пользователю удобное средство для ввода и просмотра информации. Как и бумажный документ, форма позволяет быстро ввести необходимую информацию и запомнить ее для последующей обработки, а при необходимости вновь вернуться к ранее введенным данным для просмотра или корректировки.

Отображаемая часть формы (видимая пользователю) описывается как дерево, включающее **элементы формы**. Элементы могут представлять собой поля ввода, флажки, переключатели, кнопки и т. д. Кроме того, элемент может быть группой, включающей другие элементы. Группа может представляться как панель с рамкой, панель со страницами (закладками), собственно страница, командная панель. Помимо этого элемент может представлять собой таблицу, которая тоже включает элементы (колонки). Структура элементов описывает то, как будет выглядеть форма. А вся функциональность формы описывается в виде реквизитов и команд. **Реквизиты** – это данные, с которыми работает форма, а **команды** – выполняемые действия. Таким образом, разработчик в редакторе формы должен включить в форму необходимые реквизиты и команды, создать отображающие их элементы формы и, если необходимо, скомпоновать элементы в группы.

Система может автоматически создавать форму прикладного объекта, но разработчик может сам создать форму и определить состав ее реквизитов, команд и отображаемых элементов. На основе этого логического описания система автоматически формирует внешний вид формы для отображения пользователю. При этом системой учитываются различные свойства отображаемых данных (например, тип), чтобы максимально удобно для пользователя расположить элементы формы. Разработчик может влиять на расположение элементов различными установками. Он может определять порядок элементов, указывать желаемую ширину и высоту. Однако это является только некоторой дополнительной информацией, помогающей системе отобразить форму.

В формах разработчик может использовать не только команды самой формы, но и глобальные команды, используемые в командном интерфейсе всей конфигурации. Кроме того, реализована возможность создания параметризуемых команд, которые будут открывать другие формы с учетом конкретных данных текущей формы. Например, это может быть вызов отчета по остаткам на том складе, который выбран сейчас в форме расходной накладной.

В формах существует механизм привязки сообщений, выдаваемых пользователю, к данным формы. Это позволяет системе визуально отмечать и активизировать элементы формы, при заполнении которых пользователь допустил ошибку.

Также форма автоматически учитывает ролевую доступность данных. Так, например, если какой-либо реквизит отображаемого объекта недоступен для просмотра определенному пользователю, то система автоматически удалит связанный с этим реквизитом элемент формы и перестроит форму.

Форма создается в редакторе формы, который доступен в режиме Конфигуратор (см. [здесь](#)). В этом редакторе определяются:

- **Реквизиты формы** – предназначены для хранения данных, с которыми работает форма (см. [здесь](#)).
- **Параметры формы** – предназначены для организации связи между формами и управления функциональностью формы при ее открытии (см. [здесь](#)).
- **Команды формы** – предназначены для выполнения различных действий внутри формы (см. [здесь](#)).
- **Модуль формы** – содержит программный код, связанный с функционированием формы (см. [здесь](#)).
- **Элементы формы** – предназначены для отображения и редактирования реквизитов формы, а также для отображения и выполнения команд (см. [здесь](#)).
- **Командный интерфейс** – содержит команды, которые могут выполняться в форме и предоставлены глобальным командным интерфейсом и командами формы (см. [здесь](#)).

ВНИМАНИЕ! Формы одновременно присутствуют и на стороне сервера, и на стороне клиента. Об этом всегда следует помнить при разработке формы.

7.2. Реквизиты формы

Набор реквизитов формы описывает состав данных, которые отображаются, редактируются или хранятся в форме. При этом реквизиты формы сами по себе не обеспечивают возможности отображения и редактирования данных. Для отображения и редактирования служат элементы формы, связанные с реквизитами формы. Совокупность всех реквизитов формы будем называть **данными формы**.

ВНИМАНИЕ! Необходимо помнить, что все данные формы должны быть описаны в виде реквизитов. Не допускается использование переменных модуля формы в качестве источников данных для элементов формы.

В процессе разработки формы можно явно задать возможность просмотра и редактирования конкретных реквизитов формы, в разрезе ролей, с помощью свойств [Просмотр](#) и [Редактирование](#) (подробнее см. [здесь](#)). Кроме того, доступность того или иного реквизита в самой форме можно настраивать с помощью функциональных опций (подробнее о функциональных опциях см. [здесь](#)).

Имеется возможность назначить **Основной реквизит формы**, т. е. реквизит, который будет определять стандартную функциональность формы (расширение формы).

ВНИМАНИЕ! Следует помнить, что основной реквизит у формы может быть только один.

Расширение формы – это дополнительные свойства, методы и параметры формы объекта *УправляемаяФорма*, характерные для типа, определяющего основной элемент формы.

7.2.1. Типы данных формы

Форма оперирует ограниченным набором типов данных:

- Типы, которые непосредственно используются в форме, – это те типы, которые существуют на стороне тонкого клиента и веб-клиента (например, [Число](#), [СправочникСсылка.Товары](#), [ГрафическаяСхема](#), [ТабличныйДокумент](#)).
- Типы, которые будут преобразованы в специальные типы данных, – типы данных формы. Такие типы отображаются в списке реквизитов формы в круглых скобках, например ([СправочникОбъект.Товары](#)).
- Динамический список – специальный тип данных, который позволяет организовать отображение на форме произвольной информации из таблиц базы данных. Для этого необходимо указать отображаемую таблицу или описать получаемую выборку на языке запросов. Механизм основан на системе компоновки данных и предоставляет возможности для сортировки, отбора, поиска, группировки и условного оформления получаемых данных.

Некоторые прикладные типы (такие как [СправочникОбъект](#) и т. д.) не существуют на стороне тонкого клиента и веб-клиентов. Поэтому для представления в форме таких прикладных типов в платформе введены специальные типы данных, предназначенные для работы в формах. Эта особенность обуславливает необходимость выполнять преобразование прикладных объектов в данные формы (и обратно).

Используются следующие типы данных:

- **ДанныеФормыСтруктура** – содержит набор свойств произвольного типа. Свойствами могут быть другие структуры, коллекции или структуры с коллекциями. Таким типом представляется, например, в форме [СправочникОбъект](#).
- **ДанныеФормыКоллекция** – это список типизированных значений, похожий на массив. Доступ к элементу коллекции осуществляется по индексу или по идентификатору. В коллекциях, полученных на основе наборов записей регистров или табличных частей объектов, поле элемента коллекции [НомерСтроки](#) не соответствует реальному индексу элемента коллекции. Доступ по идентификатору может отсутствовать в некоторых случаях. Это обусловлено типом прикладного

объекта, который представлен этой коллекцией. Идентификатором может быть любое целое число. Таким типом представляется, например, в форме табличная часть.

- **ДанныеФормыСтруктураСКоллекцией** – это объект, который представлен в виде структуры и коллекции одновременно. С ним можно обращаться как с любой из этих сущностей. Таким типом представляется, например, в форме набор записей.
- **ДанныеФормыДерево** – объект предназначен для хранения иерархических данных.

ВНИМАНИЕ! Не рекомендуется использовать данные формы в качестве параметров процедур и функций, выполняющих передачу управления с клиента на сервер, а также в качестве возвращаемых значений функций, если при этом управление передается на клиента.

ВНИМАНИЕ! При вызове серверных процедур и функций и возврате управления на клиента выполняется передача только изменений в данных формы. Если в качестве типа реквизита формы выбран тип **Произвольный** и в качестве значения реквизита выбрано мутабельное значение, то для того, чтобы данные между клиентом и сервером (и обратно) передавались корректно, нужно после изменения мутабельного значения снова его присвоить реквизиту формы.

Прикладной объект представлен либо одним, либо несколькими элементами данных формы. В общем виде иерархия и состав данных формы зависят от сложности и взаимосвязи прикладных объектов формы.

Например, документ, содержащий табличную часть, будет представлен объектом типа **ДанныеФормыСтруктура** (собственно документ), которому подчинен объект типа **ДанныеФормыКоллекция** (табличная часть документа).

ВНИМАНИЕ! Во время разработки конфигурации важно помнить, что прикладные объекты доступны только на сервере, в то время как объектами данных форм можно пользоваться и на сервере, и на клиенте.

При формировании реквизитов формы необходимо учитывать ряд ограничений:

- Запрещено присваивание реквизитам формы значений типа **Массив** и **Соответствие**.
- Для реквизитов произвольного типа объектов **ДанныеФормыСтруктура**, **ДанныеФормыЭлементКоллекции**, **ДанныеФормыЭлементДерева**, **ДанныеФормыСтруктураСКоллекцией** запрещается присваивание значений типа **Массив** и **Соответствие**.
- Не рекомендуется использовать значения типов **Массив** и **Соответствие** в качестве элементов коллекций типа **Структура** или **СписокЗначений**. В вышеперечисленных случаях рекомендуется использовать фиксированные коллекции: **ФиксированныйМассив**, **ФиксированноеСоответствие**.
- Не рекомендуется использовать типы **Структура** и **СписокЗначений** в данных реквизитов формы (т. е. в реквизитах реквизитов).
- Если реквизит формы имеет тип **Произвольный** или в качестве значения такого реквизита выступает коллекция (массив, соответствие и т. д.), то не следует устанавливать таким реквизитам (или свойствам коллекций) значения, содержащие прикладные объекты:
 - **ДокументОбъект**,
 - **СправочникОбъект**,
 - **БизнесПроцессОбъект**,
 - **ЗадачаОбъект**,
 - **ПланВидовХарактеристикиОбъект**,
 - **ПланВидовРасчетаОбъект**,
 - **ПланОбменаОбъект**,
 - **ПланСчетовОбъект**,
 - **ВнешнийИсточникДанныхКубТаблицаИзмеренияОбъект**,
 - **ВнешнийИсточникДанныхТаблицаОбъект**.

Фактически можно сказать, что данные формы – это унифицированное представление данных различных прикладных объектов, с которыми форма работает единообразно, и которые присутствуют и на сервере, и на клиенте. То есть форма содержит некоторую «проекцию» данных прикладных объектов в виде своих собственных типов данных и выполняет преобразование между ними при необходимости. Однако, если разработчик конфигурации реализует свой алгоритм обработки данных, то преобразование данных (из специализированных типов в прикладные и обратно) он должен выполнять самостоятельно.

При редактировании реквизитов формы имеется возможность влиять на передачу данных между клиентом и сервером во время работы формы. Для этого служит колонка редактора реквизитов **Использовать всегда**. Действие этого свойства различается для трех типов реквизитов:

- Для реквизита, подчиненного динамическому списку (колонке динамического списка):
 - Свойство включено – реквизит всегда считывается из базы данных и включается в данные формы.
 - Свойство выключено – реквизит считывается из базы данных и включается в данные формы только тогда, когда есть видимый в данный момент элемент формы, связанный с реквизитом или его подчиненным реквизитом.
- Для реквизита, подчиненного коллекции движений:
 - Свойство включено – движения документа считываются из базы данных и будут присутствовать в данных формы.
 - Свойство выключено – движения документа не будут считываться из базы данных и не попадут в данные формы (если нет элемента формы, ссылающегося на движения документа).
- Остальные реквизиты формы:
 - Свойство включено – реквизит будет присутствовать в данных формы вне зависимости от того, есть или нет хоть один элемент формы, который связан с реквизитом или его подчиненным реквизитом.
 - Свойство выключено – реквизит будет присутствовать в данных формы только в том случае, если есть элемент формы, связанный с реквизитом или его подчиненным реквизитом. В отличие от реквизитов динамического списка, здесь не играет роли видимость элемента, связанного с реквизитом.

ПРИМЕЧАНИЕ 1. Следует помнить, что свойство, установленное у родительского реквизита, действует на все подчиненные реквизиты. Например, если свойство **Использовать всегда** снято у табличной части документа, то система считает, что это свойство снято и у всех подчиненных реквизитов (несмотря на фактическое состояние свойства).

ПРИМЕЧАНИЕ 2. Для реквизитов формы, имеющих объектный тип (**СправочникОбъект**, **ДокументОбъект** и т. д.), реквизиты **Ссылка**, **ПометкаУдаления** и **ЭтоГруппа** всегда присутствуют в данных формы (если соответствующий реквизит есть у объекта), вне зависимости от состояния свойства **Использовать всегда**.

ПРИМЕЧАНИЕ 3. Запрещено присваивание реквизитам данных формы значений других объектов данных формы.

Если в редакторе реквизитов (колонка **Тип**) описание типа отображается в круглых скобках, например (**СправочникОбъект.Товары**), то это означает, что прикладной тип

будет преобразован к типам, образующим данные формы.

Если в данных формы оказывается недопустимый тип (например, как реквизит реквизита формы), то в редакторе реквизитов формы, рядом с именем типа данного реквизита, будет выводиться надпись (**Недоступен в данных формы**). В режиме 1С:Предприятие поле формы, соответствующее данному реквизиту, не будет создано. Никаких ошибок выводиться не будет. При копировании и редактировании объектов, содержащих такие данные, следует самостоятельно реализовывать программный код, обеспечивающий копирование и работу с такими реквизитами в форме.

Далее рассмотрим некоторые особенности реквизитов форм разных типов:

- Для реквизитов формы типа **ТаблицаЗначений** и **ДеревоЗначений** имеется возможность добавлять колонки (команда **Добавить колонку реквизита**). Это определяет структуру данных создаваемой коллекции.
- Для реквизитов типа **ДанныеФормыКоллекция** (например, табличные части объектов) и **ДанныеФормыСтруктураСКоллекцией** (например, наборы записей регистров) имеется возможность задавать дополнительные колонки реквизитов (команда **Добавить колонку реквизита**), которые не имеют связи с данными, хранящимися в информационной базе.

Эти колонки будут создаваться системой в момент создания данных формы. Обращение к таким реквизитам доступно как на клиенте, так и на сервере.

Пример заполнения колонки реквизита, не связанной с данными:

[Копировать в буфер обмена](#)

```
&НаСервере
Процедура ПриЧтенииНаСервере(ТекущийОбъект)
    Для Каждого Строка Из Объект.Товары Цикл
        // ВидТовара - реквизит, не связанный с данными
        Строка.ВидТовара = Строка.Товар.Вид;
    КонечЦикла
КонечПроцедуры
```

- Для реквизита типа **СписокЗначений** имеется возможность установить тип значения (свойство **Тип значения**), которое будет хранить список. При этом система будет автоматически ограничивать тип добавляемых данных при интерактивном добавлении. Также при этом не запрещается программное добавление, но будет выполняться попытка приведения значения добавляемого типа к ограничивающему типу (или составному типу). Свойство **Тип значения** также можно связать с элементом формы и получить возможность интерактивно ограничивать типы данных, которые могут быть добавлены в список значений.
- Для реквизита типа **ДинамическийСписок** имеется возможность задать параметры списка: основную таблицу, настройки и т. д. Подробнее см. [здесь](#).

ВНИМАНИЕ! Колонки реквизитов, не связанные с данными, не участвуют в преобразовании значений между данными формы и объектами информационной базы и обратно.

Описание свойств [Просмотр](#) и [Редактирование](#) см. [здесь](#).

Изменения данных формы отображаются элементами формы после окончания выполнения встроенного языка или после принудительного вызова метода **ОбновитьОтображениеДанных()**. Рассмотрим пример.

Допустим, в форме есть реквизит **Счетчик** типа **Число**. Этот реквизит отображается на форме полем типа **Поле индикатора**. Пусть есть некоторое действие, ход выполнения которого должен отображаться индикатором. Действие вызывается нажатием кнопки на форме:

[Копировать в буфер обмена](#)

```
Процедура ОбработчикКомандыФормы()
    Для Счетчик=1 по 100 Цикл
        ВыполнитьДействие();
    КонечЦикла
КонечПроцедуры
```

Если нажать кнопку, инициирующую эту команду, в режиме 1С:Предприятия, то индикатор вначале будет в крайнем левом положении, а потом – сразу в крайнем правом (считаем, что метод **ВыполнитьДействие()** выполняется какое-то время, что должно позволить увидеть изменение индикатора). То есть индикатор будет обновлен только после того, как закончится исполнение обработчика команды.

Для того чтобы индикатор отображал ход выполнения процесса, нужно код обработчика заменить нижеследующим:

[Копировать в буфер обмена](#)

```
Процедура ОбработчикКомандыФормы()
    Для Счетчик=1 по 100 Цикл
        ВыполнитьДействие();
        ОбновитьОтображениеДанных();
    КонечЦикла
КонечПроцедуры
```

После такого изменения индикатор начнет изменяться во время исполнения кода, расположенного в обработчике команды.

ПРИМЕЧАНИЕ. В веб-клиенте вызов метода **ОбновитьОтображениеДанных()** приведет к обновлению элементов формы только после окончания выполнения кода на встроенном языке.

7.2.2. Преобразование данных прикладных объектов в данные формы и обратно

Для конвертирования прикладных объектов в данные формы и обратно существует набор глобальных методов:

- **ЗначениеВДанныеФормы()**,
- **ДанныеФормыВЗначение()**,
- **КопироватьДанныеФормы()**.

Методы, работающие с прикладными объектами, доступны только в серверных процедурах. Метод для копирования значений между данными формы доступен на сервере и на клиенте, так как не требует прикладных объектов в качестве параметров.

Во время конвертирования данных формы в прикладной объект нужно учитывать их совместимость.

- **ЗначениеВДанныеФормы()** – преобразует объект прикладного типа в данные формы.
- **ДанныеФормыВЗначение()** – преобразует данные формы в объект прикладного типа.
- **КопироватьДанныеФормы()** – производит копирование данных формы, обладающих совместимой структурой. Возвращает значение **Истина**, если копирование произведено, или **Ложь**, если структура объектов несовместима.

При преобразовании данных формы в прикладные объекты и обратно используется кеширование объектов, но при этом выполняется проверка актуальности версии объекта в кеше.

ПРИМЕЧАНИЕ. При выполнении стандартных действий (открытие формы, выполнение стандартной команды **Записать** и т. д.) в форме с основным реквизитом преобразование выполняется автоматически.

Приведем пример, как использовать преобразование данных в собственных алгоритмах.

[Копировать в буфер обмена](#)

```
&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
    ОбъектТовар = Товары.НайтиПоНаименованию("Кофейник").ПолучитьОбъект();
    ЗначениеВДанныеФормы(ОбъектТовар, Объект);
КонецПроцедуры
&НаКлиенте
Процедура Записать()
    ЗаписатьНаСервере();
КонецПроцедуры
&НаСервере
Процедура ЗаписатьНаСервере()
    ОбъектТовар = ДанныеФормыВЗначение(Объект, Тип("СправочникОбъект.Товары"));
    ОбъектТовар.Записать();
КонецПроцедуры
```

Также у объекта `УправляемаяФорма` существуют методы, доступные на сервере:

- `ЗначениеВРеквизитФормы()` – выполняет преобразование объекта прикладного типа в заданный реквизит формы.
- `РеквизитФормыВЗначение()` – преобразует реквизит данных формы в объект прикладного типа.

Использование данных методов обычно удобнее, так как они имеют, например, информацию о типе реквизита формы. Кроме того, метод `РеквизитФормыВЗначение()` выполняет установку соответствия данных формы и объекта, которая используется при формировании сообщений (см. [здесь](#)).

Также следует помнить, что при преобразовании в данные формы (как с помощью метода `ЗначениеВДанныеФормы()`, так и с помощью метода `ЗначениеВРеквизитФормы()`) объектов типа `ТаблицаЗначений` или `ДеревоЗначений` нужно учитывать следующую особенность: в преобразуемом объекте должны существовать все колонки, которые существуют в данных формы.

ВНИМАНИЕ! Колонки реквизитов, не связанные с данными (см. [здесь](#)), не участвуют в преобразовании значений между данными формы и объектами информационной базы и обратно. Колонки, отсутствующие в данных объекта, очищаются при преобразовании в данные формы.

ПРИМЕЧАНИЕ. В качестве первого параметра методов `РеквизитФормыВЗначение()` и `ДанныеФормыВЗначение()` могут выступать только реквизиты формы следующих типов: `ДанныеФормыСтруктура`, `ДанныеФормыКоллекция`, `ДанныеФормыСтруктураСКоллекцией`, `ДанныеФормыДерево`.

Приведем пример использования этих методов.

[Копировать в буфер обмена](#)

```
&НаСервере
Процедура ПересчитатьНаСервере()
    // Преобразует реквизит Объект в прикладной объект.
    Документ = РеквизитФормыВЗначение("Объект");
    // Выполняет пересчет методом, определенным в модуле документа.
    Документ.Пересчитать();
    // Преобразует прикладной объект обратно в реквизит.
    ЗначениеВРеквизитФормы(Документ, "Объект");
КонецПроцедуры
```

7.2.3. Свойства реквизитов

Данный раздел содержит описание некоторых свойств реквизитов формы.

Заголовок – текст, который используется в качестве заголовка элемента формы, связанного с данным реквизитом, если не задано свойство элемента формы `Заголовок`.

Основной реквизит – определяет, что данный реквизит формы является основным и определяет расширение формы.

Сохраняемые данные – если для реквизита установлено это свойство, то интерактивное изменение такого реквизита будет приводить:

- К попытке блокировки связанного реквизита формы.
- Установке признака изменения у формы (флаг `Модифицированность`).
- Если у реквизита установлено свойство `Сохраняемые данные` и форма находится в режиме `Только просмотр`, то все элементы формы, связанные с этим реквизитом, также будут находиться в состоянии `Только просмотр`.

Проверка заполнения – определяет необходимость проверки данного реквизита на заполнение (значение свойства равно `Выдавать ошибку`). Проверка заполнения возможна только для реквизитов следующих типов:

- примитивных типов данных (`Число`, `Строка`, `Булево`, `Дата`, любые типы ссылок, стандартный период),
- список значений,
- дерево значений,
- таблица значений.

ПРИМЕЧАНИЕ. Проверка заполнения реквизитов выполняется аналогично функции `ЗначениеЗаполнено()`. Проверка заполнения табличных частей подразумевает, что табличная часть считается заполненной, когда в ней присутствует хотя бы одна строка.

7.2.4. Условное оформление формы

Форма позволяет устанавливать условное оформление элементов формы. Настройка условного оформления выполняется аналогично настройке условного оформления системы компоновки данных. Вызов настройки условного оформления доступен из панели свойств корневого элемента формы.

Доступны следующие виды условного оформления:

- **Цвет фона** – позволяет изменять цвет фона и применяется для:
 - Элементов формы: поле ввода, поле текстового документа, таблица, обычная кнопка.
 - Полей таблицы: поле надписи, поле ввода, поле переключателя, поле картинки.
- **Цвет текста** – позволяет изменять цвет текста и применяется для:
 - Элементов формы: поле надписи, поле ввода, поле текстового документа, поле картинки, поле переключателя, таблица, обычная кнопка, гиперссылка, декорация-текст, декорация-картинка.
 - Полей таблицы: поле надписи, поле ввода, поле переключателя.

• **Шрифт** – позволяет изменять шрифт и применяется для:

- Элементов формы: поле надписи, поле ввода, поле текстового документа, поле картинки, поле переключателя, поле календаря, таблица, обычная кнопка, гиперссылка, декорация-текст, декорация-картинка.
- Полей таблицы: поле надписи, поле ввода.

ПРИМЕЧАНИЕ. Изменение шрифта элемента формы, выполненное с помощью условного оформления, не учитывается при определении размера элементов.

• **Выделять отрицательные** – позволяет задать выделение отрицательных значений. Применяется для следующих элементов формы: поле надписи, поле ввода и поля таблицы: поле надписи, поле ввода.

• **Горизонтальное положение** – задает горизонтальное положение значения и применяется для:

- Элементов формы: поле надписи, поле ввода, декорация-текст.
- Полей таблицы: поле надписи, поле ввода.

• **Отметка незаполненного** – позволяет задать отметку незаполненного значения и применяется для:

- Элементов формы: поле ввода, таблица.
- Полей таблицы: поле ввода.

СОВЕТ. Если управление свойством **Отметка незаполненного** выполняется с помощью условного оформления, то рекомендуется для оформляемого поля установить свойство **Автоотметка незаполненного** в значение **Нет**.

• **Текст** – используется для задания текста и применяется для полей таблицы: поле надписи, поле ввода.

• **Формат** – позволяет изменить формат вывода значения и применяется для:

- Элементов формы: группа вида **Страница**, группа вида **Обычная группа**.
- Полей таблицы: поле надписи, поле ввода.

• **Видимость** – используется только для отключения отображения полей таблицы: поле надписи, поле ввода, поле переключателя, поле картинки. Место ячеек с отключенным отображением занимает за счет изменения размера соседних ячеек. Алгоритм растягивания ячеек может различаться в различных клиентских приложениях.

ПРИМЕЧАНИЕ. Не рекомендуется использовать условное оформление для скрытия в таблице строк целиком. Использование условного оформления для скрытия строк может негативно влиять на производительность и приводить к некорректному отображению содержимого таблицы.

• **Доступность** – используется только для отключения доступности полей таблицы: поле надписи, поле ввода, поле переключателя, поле картинки.

• **Только просмотр** – используется только для включения режима **Только просмотр** для полей таблицы: поле надписи, поле ввода, поле переключателя, поле картинки.

• **Отображать** – используется только для отключения видимости значений полей таблицы: поле надписи, поле ввода, поле переключателя, поле картинки.

ПРИМЕЧАНИЕ. Для условного оформления используется текущая дата компьютера, скорректированная с учетом часового пояса сеанса «1С:Предприятия». Подробнее про часовые пояса см. [здесь](#).

Если с помощью условного оформления оформляется колонка таблицы формы, то при вычислении условия используются текущие данные той таблицы, к которой относится оформляемая колонка. Если оформляемое поле не относится к таблице, то для вычисления условия используются текущие данные первой (по порядку размещения в редакторе формы) таблицы формы.

Условное оформление позволяет задавать некоторые виды оформления элементов формы в зависимости от значений реквизитов формы, при этом оформление формы будет динамическим, т. е. будет учитывать изменение данных формы. Для изменения оформления формы не требуется никаких специальных действий.

Рассмотрим пример такого оформления.

Допустим, нам необходимо в документе **Расход товара** выделить те строки табличной части, в которых количество менее 10 штук.

Для этого нужно в условном оформлении формы задать следующие условия:

- Вид условного оформления – цвет текста. В качестве значения цвета выберем **Особый цвет** из стиля.
- В качестве условия выберем следующее выражение: **Объект.Товары.Количество Меньше "10"**. В этом выражении **Объект** – основной реквизит формы, **Товары** – табличная часть, **Количество** – реквизит табличной части.
- В качестве оформляемых полей выберем всю табличную часть **Товары**.

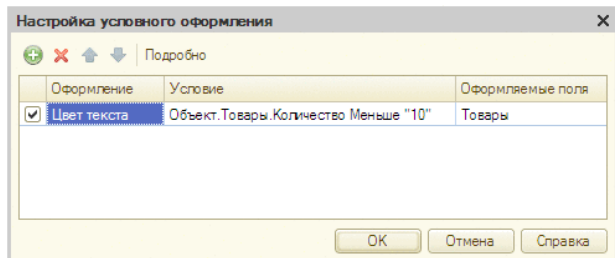


Рис. 157. Настройка условного оформления

Результат работы условного оформления можно увидеть на [рис. 158](#).

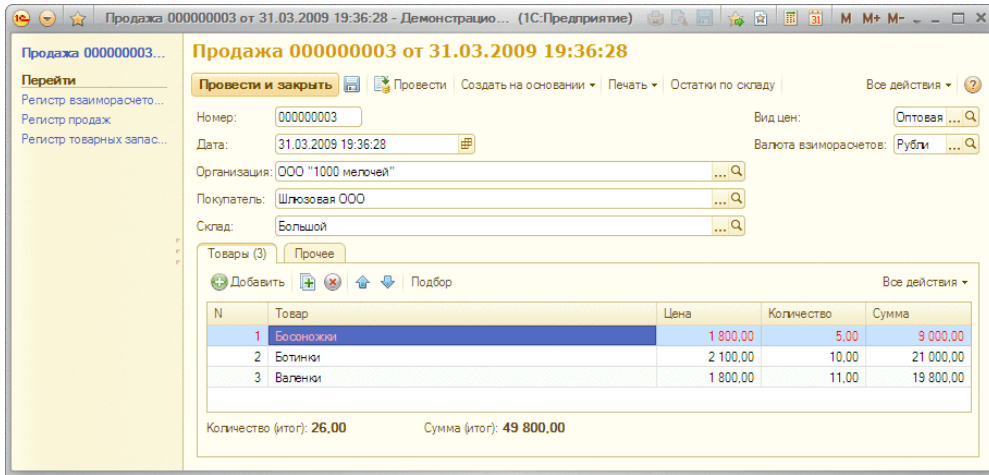


Рис. 158. Результат условного оформления

Таким образом, цвет шрифта первой строки табличной части стал красным, т. к. количество товара в этой строке меньше 10.

Однако выделять особым цветом всю строку не всегда возможно, и нужно выделять непосредственно ячейку табличной части с количеством.

Для этого изменим заданное условное оформление: в качестве оформляемых полей выберем не всю табличную часть **Товары**, а конкретное поле: **Количество**.

В этом случае будет выделено только одно поле:

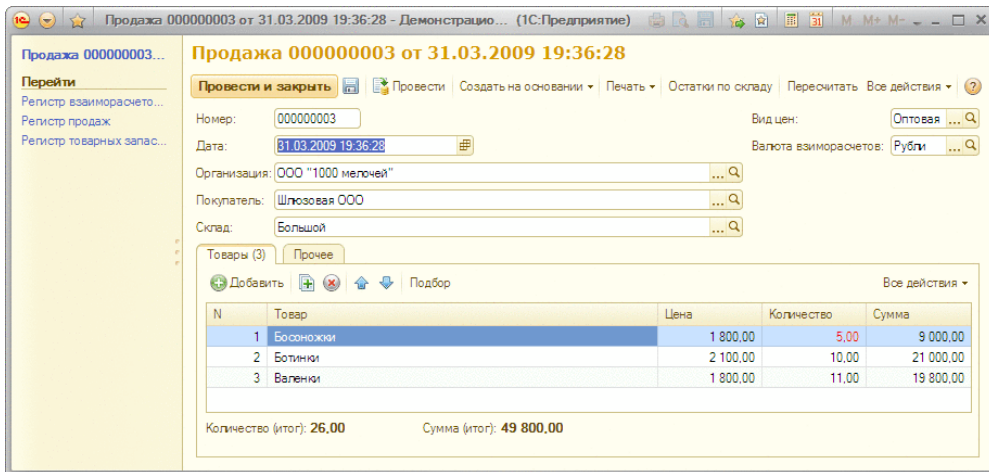


Рис. 159. Оформление одного поля табличной части

Также следует отметить возможность использовать значения колонок, не связанных с данными информационной базы, в выражениях условий.

7.2.5. Динамический список

7.2.5.1. Общая информация

Динамический список является специальным типом данных, который позволяет отображать на форме произвольную информацию из таблиц базы данных. Для этого нужно указать таблицу, данные из которой необходимо отобразить, или описать получаемую выборку на языке запросов.

Механизм основан на системе компоновки данных и предоставляет возможности для сортировки, отбора, поиска, группировки и условного оформления получаемых данных. При этом источником данных служит запрос, который либо формируется системой автоматически (на основании указанных данных), либо пишется разработчиком вручную.

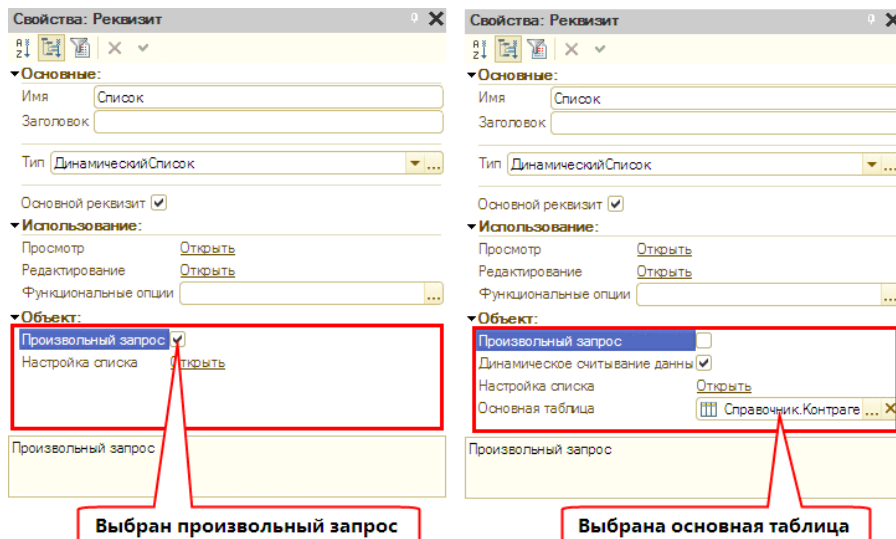


Рис. 160. Варианты создания динамического списка

При создании реквизита формы типа [ДинамическийСписок](#) разработчик может выбрать два способа формирования запроса к данным:

- Заданием основной таблицы – в этом случае достаточно просто указать таблицу (свойство [Основная таблица](#)), из которой необходимо получать данные, и система будет автоматически формировать запрос к данным (см. правую часть на [рис. 160](#)).
- Ручным формированием запроса – для этого нужно установить свойство [Произвольный запрос](#) (см. левую часть [рис. 160](#)). После этого будет доступно ручное формирование запроса получения данных из информационной базы.

С помощью запроса можно выполнять выборку данных из нескольких таблиц, поэтому можно указать основную таблицу. Это нужно для того, чтобы динамический список мог определить, какие данные главные, а какие – второстепенные, и мог правильно выбирать и отображать информацию, а также предоставлять стандартные команды. Однако если в запросе невозможно определить основную таблицу, то ее можно не указывать, но тогда динамический список не будет предоставлять команды, связанные с основной таблицей.

Для динамического списка, являющегося основным реквизитом формы, имеется возможность устанавливать значения отбора с помощью параметра формы [Отбор](#) (подробнее о параметрах формы см. [здесь](#)). Для этого необходимо, чтобы имя свойства структуры, находящееся в параметре [Отбор](#), совпало с именем поля отбора динамического списка. В этом случае значение свойства структуры будет установлено в качестве значения элемента отбора. Если в качестве значения элемента параметра [Отбор](#) формы динамического списка передан массив, фиксированный массив или список значений, то в отбор добавляется условие с вариантом [В Списке](#), в правое значение которого помещается список значений (в который преобразуются массив и фиксированный массив).

В качестве произвольного запроса в динамическом списке может выступать запрос, в котором для формирования значения какого-либо поля используется параметр, например:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    ВЫБОР
        КОГДА Доставка.Кoeffициент = 1
            ТОГДА &Представление
        ИНАЧЕ Доставка.Кoeffициент
    КОНЕЦ КАК Кoeffициент
ИЗ
    Документ.ДоставкаПродукции КАК Доставка
  
```

При этом если тип значения параметра отличается от типа реквизита объекта (например, [Реквизит1](#) имеет тип [Число](#), а значение параметра – тип [Строка](#)), то для корректного отображения поля следует выполнить явное приведение значения параметра к нужному типу:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    ВЫБОР
        КОГДА Доставка.Кoeffициент = 1
            ТОГДА ВЫРАЗИТЬ(&Представление КАК Строка(100))
        ИНАЧЕ Доставка.Кoeffициент
    КОНЕЦ КАК Кoeffициент
ИЗ
    Документ.ДоставкаПродукции КАК Доставка
  
```

Если поле, по которому устанавливается отбор, отключено с помощью функциональных опций, то отбор по такому полю не устанавливается, даже если значение отбора передано в качестве параметров формы или связей параметров выбора.

С помощью свойства [Динамическое считывание данных](#) динамическому списку указывается на необходимость считывать данные небольшими порциями. Независимо от этого признака действуют следующие условия:

- если установлен режим просмотра в виде иерархического списка, будут считываться только данные текущей группы;
- если установлен режим просмотра в виде дерева, то будут считываться только данные открытых узлов дерева;
- не поддерживается единовременная загрузка данных динамического списка в случае установленного иерархического просмотра (свойство [Отображение](#) установлено в значение [Дерево](#) или [Иерархический список](#)) и начальном отображении дерева, установленном в значение [Раскрывать все уровни](#). Для получения данных будет выполнено столько запросов к серверу, сколько узлов находится в отображаемом списке.

7.2.5.2. Ограничения и особенности

При установке отбора в динамическом списке следует помнить, что отбор не действует на группы, если для динамического списка выбран режим отображения [Иерархический список](#) или [Дерево](#).

Если выбрано ручное формирование запроса, то на запрос налагаются некоторые ограничения:

- Динамический список не поддерживает работу с пакетными запросами.
- Динамический список не поддерживает в запросе объединения, если задана основная таблица.
- В динамических списках не поддерживается группировка и сортировка по реквизитам табличных частей.
- Динамический список не должен содержать секции [Упорядочить по](#), если задана основная таблица. Нужно использовать запрос без основной таблицы или задавать необходимое упорядочивание через настройки динамического списка.
- В числе полей запроса нельзя использовать поля подзапросов, возвращающих множественное количество значений. Необходимо использовать запрос без основной таблицы.
- Запрос не может содержать группировок и агрегатных функций, если задана основная таблица. Нужно использовать запрос без основной таблицы или задавать необходимые группировки через настройки динамического списка.
- Если динамический список отображается в виде иерархического списка или дерева, запрос не должен содержать условий отбора по родителю.
- Если динамический список отображается в виде иерархического списка или дерева, запрос должен быть построен так, чтобы в результате запроса присутствовали как элементы, так и их родители.
- Не поддерживается сортировка в динамическом списке, если запрос содержит агрегатные функции.
- Не поддерживается сортировка по полям, которые являются субконто регистрами бухгалтерии.
- В случае указания основной таблицы динамического списка запрос не должен содержать инструкций [ПЕРВЫЕ](#) и [РАЗЛИЧНЫЕ](#).
- В случае указания произвольного запроса, этот запрос не должен содержать инструкцию [ПЕРВЫЕ](#).
- Не поддерживается указание в качестве основной таблицы динамического списка таблицы, которая используется в запросе только во внешнем соединении.
- Не рекомендуется использовать соединения таблиц внутри блока [ИЗ](#) запроса динамического списка, если эти соединения указаны с помощью расширения языка запросов для системы компоновки данных (в фигурных скобках). Рекомендуется переписать запрос, избавившись от таких соединений.

При создании динамического списка нужно учитывать следующие особенности:

- Не поддерживается указание табличных частей объектов в качестве основной таблицы динамического списка.

- В качестве основной таблицы не могут быть выбраны таблицы регистрации изменений (используемые в механизмах обмена данными), таблицы последовательностей и таблицы перерасчетов (используемые в механизмах периодических расчетов).

- Если запрос, заданный для динамического списка, не обеспечивает уникальность выбираемых строк, то отображение строк списка и прокрутка будут выполняться некорректно.

Другими словами, динамический список с указанной основной таблицей будет работать корректно в том случае, если в результате выполнения запроса, указанного в качестве источника данных, не увеличивается количество строк, получаемых из основной таблицы (с учетом наложенного отбора).

Если же в результате выполнения запроса количество строк, получаемых запросом из основной таблицы, увеличивается, это будет приводить к нарушению уникальности ключа записей таблицы, отображаемой списком. В этом случае необходимо отключить использование основной таблицы.

При программном изменении свойств динамического списка не происходит автоматического повторного заполнения командных панелей, связанных с этим динамическим списком.

Для динамического списка, который отображает список перечисления, отсутствует возможность интерактивной настройки списка.

Состав колонок и настройки динамического списка связаны с полями запроса по именам. Это означает, что при изменении имени поля запроса, формирующего динамический список, будут утеряны настройки реквизита динамического списка, элементов формы «потеряет» отображаемый реквизит, настройки динамического списка станут неверными.

Если форма списка предполагает отбор по связанным данным (например, отбор по владельцу в списке подчиненного справочника), то запрос формирования динамического списка в этом случае должен включать поле, содержащее ссылку на родительский элемент. Это поле должно иметь имя соответствующего стандартного реквизита: *Владелец*, *Родитель*, *БизнесПроцесс* и т. д. В том случае, когда это требование не соблюдается, стандартная команда перехода к подчиненному списку сработает, но отбор установлен не будет. Если по какой-то причине выполнить требование не представляется возможным, следует отказаться от использования стандартных команд и реализовывать собственные команды для открытия таких списков.

В том случае, если динамический список отображает данные, источником которых (свойство *Основная таблица*) является виртуальная таблица (см. [здесь](#)), для которой определены параметры *НачалоПериода* и *КонецПериода*, то при установке пользователем периода отображения (команда *Установить интервал дат...*), заданные границы периода подставляются непосредственно в параметры виртуальной таблицы.

В качестве параметра запроса динамического списка может выступать массив или список значений. Однако если параметром является список значений, то в качестве значения отбора будет использовано только первое значение списка. В случае если динамический список использует запрос с параметрами, первоначальная установка значений параметров должна выполняться в обработке *ПриСозданииНаСервере*.

7.2.5.3. Способы получения данных динамическим списком

При получении данных для отображения динамический список использует один из трех способов:

1. Считывание из базы данных выполняется порциями с количеством элементов данных, равным количеству строк, одновременно отображаемых списком. Не выполняется кеширование данных на сервере.
2. Считывание из базы данных выполняется порциями по 1000 элементов данных. Выполняется кеширование данных на сервере.
3. Считывание из базы данных выполняется порциями. Первая порция равна 1000 элементов. Каждая следующая порция увеличивается на 1000 элементов (при достижении конца предыдущей выборки). Чем ближе передвигается «точка просмотра» к концу отображаемых данных, тем большая выборка считывается из базы данных, в пределе становясь равной всем отображаемым данным. Выполняется кеширование данных на сервере.

В зависимости от того, что выбрано основной таблицей динамического списка и какое значение принимает свойство *Динамическое считывание*, используются те или иные способы считывания данных:

- В качестве значения свойства *Основная таблица* указана одна из следующих таблиц: план обмена, справочник, список документов, журнал документов, план видов характеристик, план счетов, план видов расчета, бизнес-процесс, задача, таблица точек бизнес-процесса:

- Ключ, идентифицирующий строку таблицы: *Ссылка*.
- Установлено свойство *Динамическое считывание*: используется способ 1 (описание способов приведено выше).
- Снято свойство *Динамическое считывание*: используется способ 2 (описание способов приведено выше).

- В качестве значения свойства *Основная таблица* указана одна из следующих таблиц: основная таблица регистра сведений, регистра накопления, регистра бухгалтерии, регистра расчета, виртуальная таблица регистра бухгалтерии *ДвиженияССубконто*:

- Ключ, идентифицирующий строку таблицы: *КлючЗаписи*.
- Установлено свойство *Динамическое считывание*: используется способ 1 (описание способов приведено выше).
- Снято свойство *Динамическое считывание*: используется способ 2 (описание способов приведено выше).

- В качестве свойства *Основная таблица* указана таблица критерия отбора:

- Ключ, идентифицирующий строку таблицы: *Ссылка*.
- Свойство *Динамическое считывание* не применимо.
- Используется способ 2 (описание способов приведено выше).

- В качестве свойства *Основная таблица* указана виртуальная таблица регистра сведений *СрезПервых* или *СрезПоследних*:

- Ключ, идентифицирующий строку таблицы: *КлючЗаписи*.
- Свойство *Динамическое считывание* не применимо.
- Используется способ 2 (описание способов приведено выше).

- В качестве свойства *Основная таблица* указана одна из виртуальных таблицы регистров, кроме перечисленных выше:

- Ключ, идентифицирующий строку таблицы: *Число*.
- Свойство *Динамическое считывание* не применимо.
- Используется способ 3 (описание способов приведено выше).

- Свойство *Основная таблица* не указано, используется произвольный запрос:

- Ключ, идентифицирующий строку таблицы: *Число*.
- Свойство *Динамическое считывание* не применимо.
- Используется способ 3 (описание способов приведено выше).

Для отображения данные передаются на клиента порциями следующего размера: количество элементов данных, одновременно отображаемых списком, увеличенное на две строки.

При создании формы, содержащей динамический список, первоначально на клиента передается 42 элемента данных каждого динамического списка (если в списке более 42 элементов). Если динамический список отображает более 42 строк, то при открытии формы на клиента будут переданы данные по стандартному алгоритму: количество элементов данных, одновременно отображаемых списком, увеличенное на две строки.

В своей работе динамический список использует значения следующих свойств реквизитов объектов метаданных:

- формат,
- формат редактирования,
- подсказка,
- признак выделения отрицательных значений,
- маска,
- признак многострочного режима,
- признак расширенного редактирования,
- режим пароля.

При отображении и редактировании отбора и параметров системы компоновки данных используется формат редактирования соответствующего поля.

7.2.5.4. Настройки динамического списка

Свойство [Настройка списка](#) – нажатие на гиперссылку [Открыть](#) приводит к открытию формы настройки отображения динамического списка. Настройка списка выполняется таким же образом, как и аналогичные операции в системе компоновки данных.

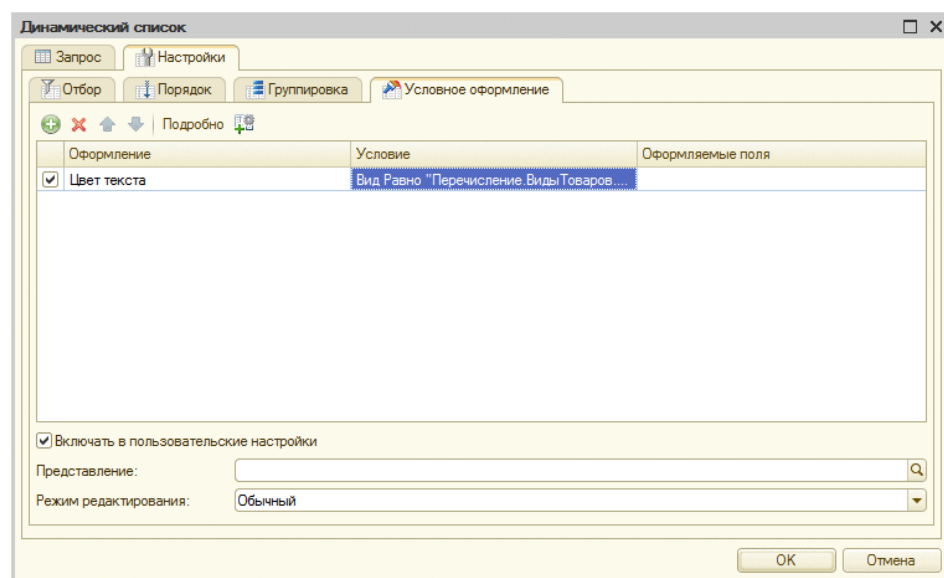


Рис. 161. Условное оформление динамического списка

Во время настройки динамического списка в конфигурации разработчик прикладного решения имеет возможность сделать следующее:

- задать поля, по которым необходимо проводить упорядочивание;
- описать отбор данных в списке;
- указать настройки условного оформления;
- задать поля, по которым необходимо группировать данные.

Задавать сортировку разработчику имеет смысл, если не устраивает устанавливаемая системой сортировка по умолчанию.

СОВЕТ. Следует помнить о том, что неудачный выбор полей сортировки (равно как отбора и группировки данных) негативно влияет на эффективность динамической выборки.

ПРИМЕЧАНИЕ 1. Не разрешается сортировка и группировка по полям, содержащим строки неограниченной длины.

ПРИМЕЧАНИЕ 2. Поля представлений недоступны для использования в отборе, группировке и сортировке. Также по полям представления невозможно выполнять поиск.

С точки зрения прикладного разработчика, настройки динамического списка состоят из нескольких частей, которые связаны между собой. Основным свойством, через которое можно управлять настройками динамического списка является [КомпоновщикНастроек](#). Это объект содержит три набора настроек, которые при работе системы определяют окончательные настройки, применяемые к динамическому списку:

- **Настройки** – настройки, созданные в режиме Конфигуратора. Свойство динамического списка [Порядок](#) предоставляет быстрый доступ к свойству [Настройки.Порядок](#) компоновщика настроек динамического списка, таким образом, следующие конструкции являются эквивалентными: [Список.Порядок](#) и [Список.КомпоновщикНастроек.Настройки.Порядок](#);
- **ПользовательскиеНастройки** – это настройки, которые изменяет пользователь в режиме «1С:Предприятие»;
- **ФиксированныеНастройки** – эти настройки задаются программно, из прикладного языка. Также в этой свойство попадают значения отбора, которые передаются в форму с помощью ее параметров. Свойства динамического списка [Отбор](#), [Параметры](#), [УсловноеОформление](#) предоставляют быстрый доступ к фиксированным настройкам компоновщика настроек динамического списка. Другими словами, данные обращения являются эквивалентными: [Список.КомпоновщикНастроек.ФиксированныеНастройки.Отбор](#) и [Список.Отбор](#).

При формировании окончательной настройки динамического списка, различные варианты настроек комбинируются следующим образом:

- Если какой-либо вид настроек целиком отмечен как пользовательский, то в результирующие настройки попадают пользовательские настройки (

Список.КомпоновщикНастроек.ПользовательскиеНастройки»). При этом если какие-либо элементы настроек отмечены как недоступные, то эти настройки будут помещены в результирующие настройки из свойства [Список.КомпоновщикНастроек.Настройки](#).

- Если какой-либо вид настроек отмечен как пользовательский не целиком, а поэлементно, то:
 - Элементы, отмеченные как пользовательские, попадут в результирующие настройки из свойства [Список.КомпоновщикНастроек.ПользовательскиеНастройки](#).
 - Элементы, отмеченные как недоступные, попадут в результирующие настройки из свойства [Список.КомпоновщикНастроек.Настройки](#).
- Фиксированные настройки ([Список.КомпоновщикНастроек.ФиксированныеНастройки](#)) добавляются в результирующие настройки «как есть». При этом недопустима ситуация, когда в фиксированных и пользовательских настройках есть одноименные настройки, например отбор с одинаковым левым значением в условии.

Если в настройках динамического списка присутствуют настройки, которые отключены с помощью функциональных опций, эти настройки будут удалены из списка доступных настроек при получении данных динамического списка.

Управление тем, какие настройки будут доступны пользователю, а какие – нет, выполняется в окне настройки динамического списка.

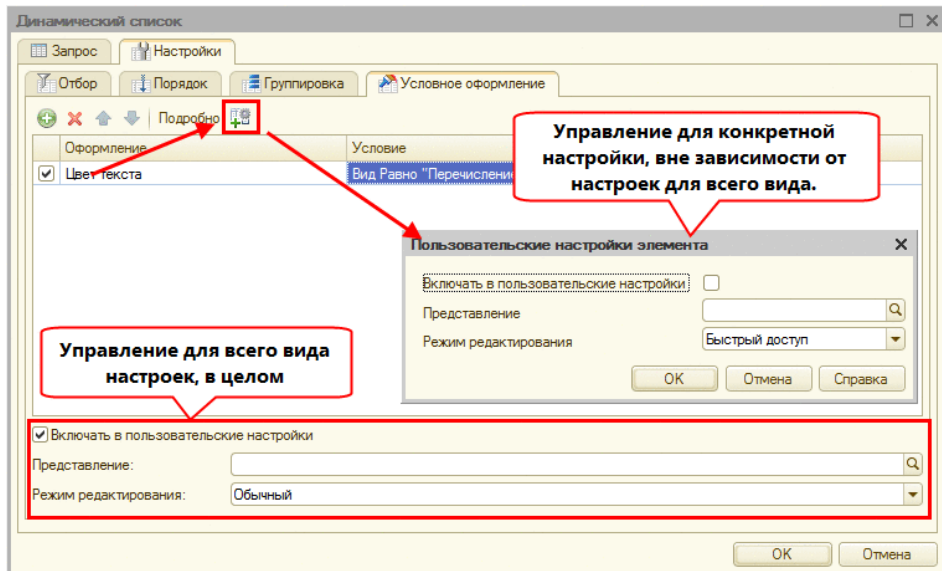


Рис. 162. Управление включением в пользовательские настройки

Флажок в нижней части окна (см. [рис. 162](#)) отвечает за размещение в настройках (обычных или быстрых) всего вида настроек. Эта возможность доступна для отбора, порядка, группировки и условного оформления. Если настройки указаны с режимом редактирования **Быстрый выбор**, то в свойстве **Группа пользовательских настроек** таблицы формы, отображающей динамический список, необходимо указать пустую группу формы, в которой будут располагаться элементы, связанные с быстрыми пользовательскими настройками динамического списка. Если группа не указана – быстрые пользовательские настройки не будут отображены на форме. Также имеется возможность явно вызвать создание пользовательских настроек с помощью встроенного языка с помощью метода `СоздатьЭлементыФормыПользовательскихНастроек()` расширения динамического списка.

Также имеется возможность выбирать возможность размещения в пользовательских настройках конкретных элементов настроек. Эта возможность доступна для элементов отбора и условного оформления (см. [рис. 162](#)).

Если необходимо, чтобы при открытии динамического списка были загружены какие-либо специальные настройки, то это можно сделать двумя способами:

- С помощью параметра формы динамического списка **ПользовательскиеНастройки**. Данные, содержащиеся в этом параметре, будут помещены в пользовательские настройки динамического списка.
- С помощью параметра формы динамического списка **КлючПользовательскихНастроек**. Если при открытии формы указать этот параметр, то в динамический список, являющийся основным реквизитом формы, будут загружены пользовательские настройки, расположенные в хранилище настроек с указанным ключом.

7.2.5.5. Поиск в динамическом списке

Динамический список, расположенный на форме, предоставляет возможность выполнять интерактивный поиск в отображаемых данных. Поиск может осуществляться с помощью следующих инструментов: строки поиска, диалога поиска, поиска текущего значения, с помощью истории поиска и установкой периода (для динамических списков, отображающих документы). Результатом поиска выступает ограниченный набор записей динамического списка (из доступных данному пользователю), которые соответствуют критериям поиска.

Для управления возможностями поиска в динамическом списке предназначены три свойства таблицы управляемой формы, отображающей динамический список:

- **Положение строки поиска** – определяет положение строки поиска. Может принимать следующие значения: **Авто**, **Командная панель**, **Нет**, **Верх**, **Низ**.

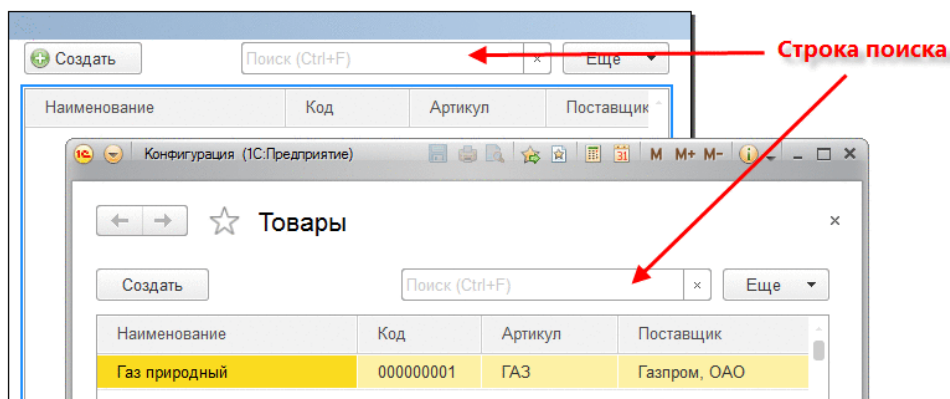


Рис. 163. Строка поиска в динамическом списке

Если значение этого свойства установлено в значение **Командная панель**, то в командной панели формы (если динамический список является основным реквизитом формы) или в командной панели, связанной с динамическим списком, будет отображаться строка поиска. Строка поиска, размещенная в командной панели, всегда прижимается к правому краю командной панели (вместе с кнопками, расположенными правее строки поиска).

Если свойство установлено в значение **Нет**, то строка поиска будет отсутствовать на форме, а при начале набора поисковой строки будет открываться диалог.

Если свойство установлено в значение **Верх**, то строка поиска будет расположена между командной панелью списка и таблицей, отображающей динамический список. Если свойство установлено в значение **Низ**, то строка поиска будет размещена сразу после таблицы, отображающей динамический список.

Если форма создана в «1С:Предприятии» версии 8.3.4 и младше – свойство установлено в значение **Нет**. Если форма создана в «1С:Предприятии» версии 8.3.5 и старше – свойство установлено в значение **Авто**. Реальное значение свойства в этом случае будет определяться следующим образом:

- Если свойство **Режим совместимости** установлено в значение **Версия 8.3.4** (и ниже) – значение **Нет**;
- Если свойство **Режим совместимости** установлено в значение **Не использовать** или старше значения **Версия 8.3.4** – значение **Командная панель**.

Переход к строке поиска осуществляется следующим образом:

- Нажатием сочетания клавиш **Ctrl+F**;
- Мышью;
- При начале набора текста в динамическом списке (с учетом значения свойства динамического списка **ПоискПриВводе**).
- **Положение состояния просмотра** – описывает, где будет отображаться состояние просмотра: по каким полям выполнялся поиск и какие значения ищались в каждом поле. Может принимать следующие значения: **Авто**, **Нет**, **Верх**, **Низ**.

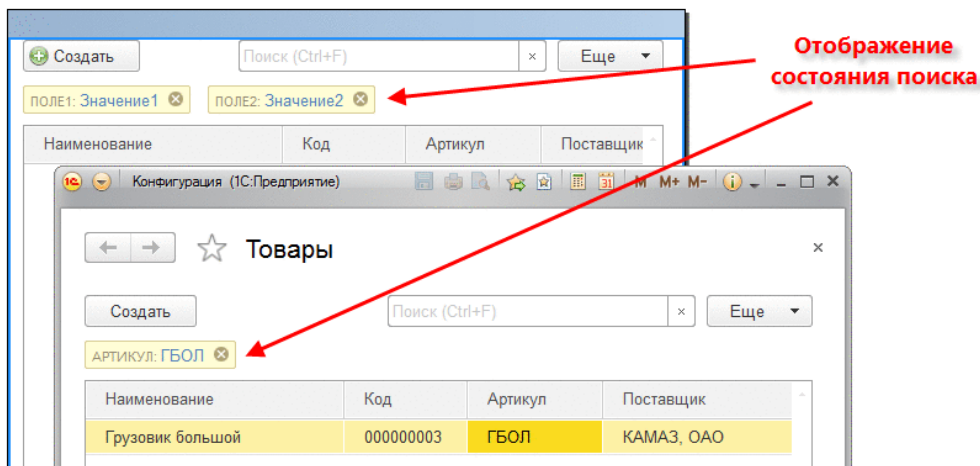


Рис. 164. Состояние поиска в динамическом списке

Если свойство установлено в значение **Нет**, то состояние просмотра будет отсутствовать на форме. В результате определить, выполнен поиск или нет, можно будет только по доступности кнопки **Отменить поиск**.

Если свойство установлено в значение **Верх**, то состояние просмотра будет расположено между командной панелью списка и таблицей, отображающей динамический список. Если свойство установлено в значение **Низ**, то состояние просмотра будет размещено сразу после таблицы, отображающей динамический список.

Если форма создана в «1С:Предприятии» версии 8.3.4 и младше – свойство установлено в значение **Нет**. Если форма создана в «1С:Предприятии» версии 8.3.5 и старше – свойство установлено в значение **Авто**. Реальное значение свойства в этом случае будет определяться следующим образом:

- Если свойство **Режим совместимости** установлено в значение **Версия 8.3.4** (и ниже) – значение **Нет**;
- Если свойство **Режим совместимости** установлено в значение **Не использовать** или старше значения **Версия 8.3.4** – значение **Верх**;
- **Положение управления поиском** – определяет, где будет отображаться кнопка управления поиском. Кнопка открывает меню, которое содержит следующую информацию: команды **Найти по текущему значению**, **Расширенный поиск**, **Отменить поиск**, **Установить период** (для списков документов и журналов) и история поисковых запросов (последние 5 запросов). Свойство может принимать значения: **Авто**, **Нет**, **Командная панель**.

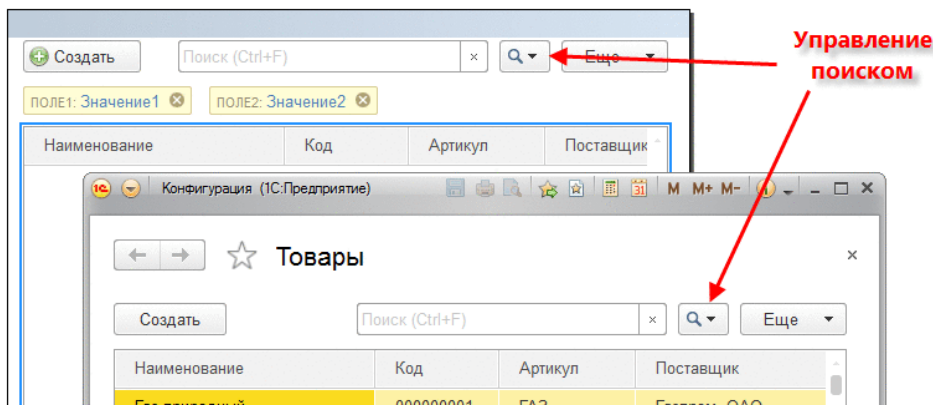


Рис. 165. Управление поиском в динамическом списке

Если свойство установлено в значение **Нет**, то кнопка управления поиском будет отсутствовать на форме (но команды будут доступны с помощью меню **Еще**). Значение свойства **Командная панель** размещает кнопку на командную панель, связанную с таблицей, отображающей динамический список.

Если форма создана в «1С:Предприятии» версии 8.3.4 и младше – свойство установлено в значение **Нет**. Если форма создана в «1С:Предприятии» версии 8.3.5 и старше – свойство установлено в значение **Авто**. Реальное значение свойства в этом случае будет определяться следующим образом:

- Если свойство **Режим совместимости** установлено в значение **Версия 8.3.4** (и ниже) – значение **Нет**;
- Если свойство **Режим совместимости** установлено в значение **Не использовать** или старше значения **Версия 8.3.4** – значение **Командная панель**;

Если на форме присутствуют несколько командных панелей, источником команд для которых является одна таблица управляемой формы (отображающая данные динамического списка), то строка поиска и кнопка управления поиском будет располагаться только в одной командной панели:

- Или в командной панели самого динамического списка (если для нее включено автоматическое заполнение)

- Или в любой из оставшихся командных панелей.

Рассмотрим особенности использования поиска в динамическом списке:

- Для того чтобы поиском было удобно пользоваться (в том числе и с точки зрения быстродействия) необходимо включить полнотекстовый поиск для всех объектов конфигурации, которые могут использоваться в качестве основной таблицы динамического списка. Также в полнотекстовом поиске должны участвовать все реквизиты объектов конфигурации, которые могут отображаться в динамическом списке и по которым может потребоваться поиск.

Если объект исключен из полнотекстового поиска, то рассматриваемый механизм поиска будет работать, но производительность такого поиска будет крайне низкой. Не рекомендуется использовать поиск по объектам, которые не индексируются полнотекстовым поиском.

- В прикладном решении должно быть регламентное задание, которое регулярно обновляет индекс полнотекстового поиска.
- Поиск выполняется не по всем колонкам динамического списка (и объекта конфигурации), а только по тем колонкам, которые отображаются в таблице.
- Для динамических списков с указанной основной таблицей используется полнотекстовый поиск по основной таблице, а также выполняется поиск по полям, отображаемым в списке из других таблиц. Результат полнотекстового поиска используется в качестве отбора по ссылочному полю. Дальнейшее наложение условий, связанных с формированием условий по словам из строки поиска, служит для сужения выборки, изначально сформированной полнотекстовым поиском. Если в объекте конфигурации, по основной таблице которого выполняется поиск, есть реквизит, который может содержать требуемые данные, но этот реквизит не входит в индекс полнотекстового поиска, то соответствующие элементы данных никогда не будут найдены в процессе поиска (при включенном полнотекстовом поиске). Без включенного полнотекстового поиска такие данные могут быть найдены, но сам поиск будет выполняться очень медленно.
- Строка поиска разбивается на слова. Это разбиение выполняется по следующим правилам:
 - строка разбивается, используя символы пробела и табуляции в качестве разделителей;
 - затем обрабатывается каждый получившийся фрагмент:
 - если фрагмент является представлением даты (с временем или без него) с учетом текущих региональных установок сеанса, то словом является этот фрагмент;
 - иначе фрагмент разбивается далее, используя в качестве разделителей символы ".,-/\". В этом случае в качестве слова принимается каждый получившийся фрагмент строки.
- Для каждого слова формируется свой набор условий, которые объединяются «по ИЛИ». Для каждого слова:
 - Для поля типа **Строка** условие имеет вид `ИмяПоля ПОДОБНО %Слово%`;
 - Для поля типа **Число** условие имеет вид `ИмяПоля=Значение`, где `Значение` – это слово, которое приведено к типу **Число**. Если приведение выполнить невозможно – поиск по полю выполняться не будет;
 - Слово ищется как подстрока в представлении по умолчанию для типа **Булево**, определенное для текущего сеанса. Если искоемое слово обнаружено в представлении, то в дальнейшем выполняется поиск значения, соответствующего представлению, в котором обнаружено слово. При этом для поиска не используются представления, которые заданы с помощью свойства элемента формы **Формат**.
 - Для поля типа **Дата** условие имеет вид `ИмяПоля>= НачалоДня(Слово) И ИмяПоля<=КонецДня(Слово)`;
 - Для ссылочных полей выполняется поиск по полям, используемым для формирования представления ссылки. В каждом из этих полей поиск выполняется по правилам, описанным выше. Для поиска не используются поля, используемые для формирования произвольного представления данных (см. [здесь](#)).

Набор условий для каждого слова объединяются «по И».

- Для значений с ведущими нулями, выполнять поиск можно как по строке с ведущими нулями, так и по строке, указанной без ведущих нулей.
- Если динамический список отображает список документов или журнал документов, то заданный интервал просмотра списка также отображается в области формы, отведенной для отображения состояния просмотра для необходимого динамического списка.
- Команда поиска по текущему значению не доступна в том случае, если основной таблицей динамического списка является критерий отбора.
- Найденные фрагменты строк выделяются при отображении в таблице.
- Для одной колонки поддерживается поиск только по одной поисковой строке. При добавлении нового поискового запроса для колонки, по которой уже выполняется поиск, произойдет замена поискового выражения, а не сложение двух поисковых запросов.
- Если на форме отсутствует дополнение элемента формы вида **Отображение строки поиска**, связанное с таблицей (свойство **Источник** дополнения элемента формы), отображающей динамический список, то нажатие сочетания клавиш **Ctrl+F** приводит к открытию диалога поиска.

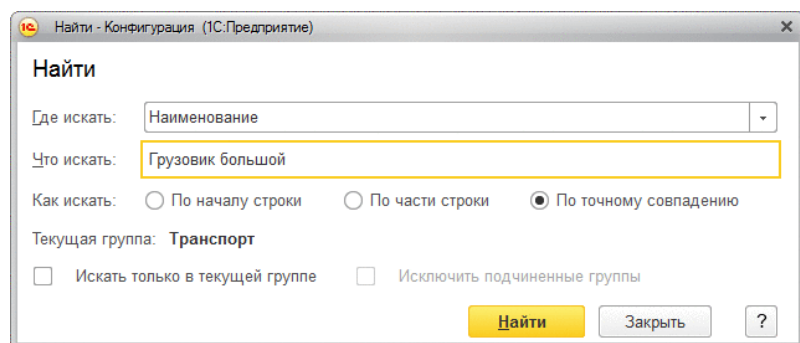


Рис. 166. Диалог поиска

Если на форме присутствует дополнение элемента формы вида **Отображение строки поиска**, связанное с таблицей (свойство **Источник** дополнения элемента формы), отображающей динамический список, то для открытия диалога поиска следует использовать команду **Расширенный поиск**.

- При использовании диалога поиска нужно помнить о следующих особенностях:
 - Открытие диалога поиска по сочетанию клавиш приводит к тому, что значение текущей ячейки оказывается в строке **Что искать**, а значение переключателя **Как искать** устанавливается в значение **По точному совпадению**.
 - Открытие диалога поиска путем непосредственного начала набора поисковой строки в динамическом списке приводит к тому, что значение переключателя **Как искать** устанавливается в значение **По части строки**, а набираемый текст попадает в поле **Что искать**.

7.2.5.6. Получение данных, отображаемых динамическим списком

При использовании динамических списков может потребоваться выполнить различные действия с данными, которые в данный момент отображаются динамическим списком, с учетом наложенных отборов и примененного поиска. К таким действиям относятся: обработка отображаемой информации, например, перепроведение

выделенных документов или установка какого-либо реквизита у выделенных объектов, формирование списка доступных объектов (с оформлением и т. д.), например, для печати или сохранения в табличный документ.

Для получения данных, которые отображаются динамическим списком, следует использовать методы `ПолучитьИсполняемуюСхемуКомпоновкиДанных()` и `ПолучитьИсполняемыеНастройкиКомпоновкиДанных()`.

Пример получения данных:

[Копировать в буфер обмена](#)

```
Схема = Список.ПолучитьИсполняемуюСхемуКомпоновкиДанных();
Настройки = Список.ПолучитьИсполняемыеНастройкиКомпоновкиДанных();
КомпоновщикМакета = Новый КомпоновщикМакетаКомпоновкиДанных();
МакетКомпоновки = КомпоновщикМакета.Выполнить(Схема, Настройки);
ПроцессорКомпоновки = Новый ПроцессорКомпоновкиДанных;
ПроцессорКомпоновки.Инициализировать(МакетКомпоновки);
ПроцессорВывода = Новый ПроцессорВыводаРезультатаКомпоновкиДанныхВТабличныйДокумент;
Возврат ПроцессорВывода.Вывести(ПроцессорКомпоновки);
```

Получение данных в коллекцию значений (таблица или список значений) выполняется аналогично (см. [здесь](#)).

Получение данных динамического списка указанным образом имеет ряд особенностей, которые нужно учитывать при разработке прикладных решений:

- Не поддерживается следующее оформление таблицы:
 - Чередование цветов строк;
 - Картинка шапки;
 - Картинка подвала;
 - Цвет фона подвала;
 - Цвет текста подвала;
 - Шрифт подвала;
 - Горизонтальное положение в подвале;
 - Режим пароля.
- Не поддерживается условное оформление, указанное для управляемой формы;
- При упорядочивании иерархической таблицы по возрастанию, по полю типа `Ссылка`, записи, содержащие пустую ссылку всегда размещаются первыми.

7.2.6. Форматированный документ

В процессе разработки прикладных решений возникает необходимость предоставить пользователю возможность выполнять редактирование текстов с оформлением (далее форматированный документ). В качестве примера такой необходимости можно привести написание писем электронной почты, различных служебных записок или сопроводительных документов.

Для работы с форматированным документом предназначен объект `ФорматированныйДокумент`, с помощью которого можно выполнять программную обработку документа, и поле ввода вида `Поле форматированного документа`, который предназначен для интерактивного изменения форматированного документа.

В информационной базе форматированный документ рекомендуется хранить в реквизите типа `ХранилищеЗначения`, в который помещается объект типа `ФорматированныйДокумент`.

Для того чтобы предоставить пользователю возможность интерактивного изменения документа, следует выполнить следующие шаги:

- Создать реквизит формы типа `ФорматированныйДокумент` и установить для него свойство `Сохраняемые данные`.
- Создать элемент формы типа `Поле ввода` вида `Поле форматированного документа` и связать его с ранее созданным реквизитом.
- При чтении данных формы (обработчик `ПриЧтенииНаСервере()`) загрузить в реквизит формы документ (метод `УстановитьHTML()`), который необходимо редактировать, предварительно получив его из информационной базы.
- Перед записью данных формы (обработчик `ПередЗаписьюНаСервере()`) получить результат (метод `ПолучитьHTML()`) редактирования документа и поместить его в реквизит, сохраняемый в информационной базе.

В процессе работы с документом можно использовать **закладки**. Закладкой называет позиция в документе. При определении позиции нужно учитывать следующие особенности:

- содержимое документа считается одной последовательностью символов;
- перенос строки считается одним символом;
- картинка считается одним символом.

С помощью закладок можно получать и устанавливать выделение или **позицию** курсора в редакторе, добавлять и удалять элементы текста в документе (собственно текст, картинка, перевод строки).

Если при получении выделения позиция начала и позиция окончания совпадают, это значит, что выделение отсутствует и получено положение курсора. Так же и при установке выделения: если начальная и конечная позиция выделения совпадают, происходит изменение позиции курсора, без выделения. Также следует помнить, что при программном изменении выделенного текста не происходит отмены выделения. Выделение остается в тех же позициях, в которых оно было до программного изменения текста.

Приведем пример получения и установки выделения в редакторе форматированного документа.

[Копировать в буфер обмена](#)

```
// Содержимое – реквизит формы типа ФорматированныйДокумент
// Редактор – элемент формы типа Поле форматированного документа
// Начало – реквизит формы типа Число. Указывает позицию начала выделения.
// Окончание – реквизит формы типа Число. Указывает позицию окончания выделения.
&НаКлиенте
Процедура ПолучитьВыделение()
    ПозицияНачала = 0;
    ПозицияОкончания = 0;
    Элементы.Редактор.ПолучитьГраницыВыделения(ПозицияНачала, ПозицияОкончания);
    Начало = Содержимое.ПолучитьПозициюПоЗакладке(ПозицияНачала);
    Окончание = Содержимое.ПолучитьПозициюПоЗакладке(ПозицияОкончания);
КонецПроцедуры
&НаКлиенте
Процедура УстановитьВыделение()
    ПозицияНачала = Содержимое.ПолучитьЗакладкуПоПозиции(Начало);
```

```

ПозицияОкончания = Содержимое.ПолучитьЗакладкуПоПозиции(Окончание);
Элементы.Редактор.УстановитьГраницыВыделения(ПозицияНачала,ПозицияОкончания);
КонецПроцедуры

```

При выполнении операций удаления закладка, описывающая начало удаляемого фрагмента, может оказаться некорректной. Если при этом необходимо сохранить позицию начала удаляемого фрагмента, то следует воспользоваться позицией начала фрагмента. Так, пример замены выделенного текста другим текстом будет выглядеть следующим образом.

```

Процедура ВставитьСтрокуВТекущуюПозицию(Редактор, Содержимое, Строка)
Перем Начало, Конец;
Редактор.ПолучитьГраницыВыделения(Начало, Конец);
Позиция = Содержимое.ПолучитьПозициюПоЗакладке(Начало);
Содержимое.Удалить(Начало, Конец);
Начало = Содержимое.ПолучитьЗакладкуПоПозиции(Позиция);
Содержимое.Вставить(Начало, Строка);
Закладка = Содержимое.ПолучитьЗакладкуПоПозиции(Позиция + СтрДлина(Строка));
Редактор.УстановитьГраницыВыделения(Закладка, Закладка);
КонецПроцедуры

```

[Копировать в буфер обмена](#)

Если в форматированном документе присутствуют гиперссылки и форматированный документ отображается в режиме **Только просмотр**, то гиперссылки доступны для перехода. Переход выполняется в новом окне веб-браузера.

Также имеется возможность программного доступа к тексту форматированного документа. Сам текст состоит из объектов типа **ПараграфФорматированногоДокумента**, который доступен через свойство **Элементы** объекта **ФорматированныйДокумент**. При этом каждый параграф состоит из объектов типа **ТекстФорматированногоДокумента**, **ПереводСтрокиФорматированногоДокумента** и **КартинкаФорматированногоДокумента** (элементы форматированного документа). Доступ к этой коллекции обеспечивается через свойство **Элементы** объекта **ПараграфФорматированногоДокумента**. Границей элемента не обязательно является граница слова.

Например, следующий пример показывает перебор всех параграфов форматированного документа, содержащих текст, и отдельную обработку каждого параграфа.

Пример:

```

Для каждого Параграф Из Документ.Элементы Цикл
Если Параграф.ТипПараграфа = ТипПараграфа.Обычный Тогда
ОбработатьОбычныйПараграф(Параграф);
ИначеЕсли Параграф.ТипПараграфа = ТипПараграфа.МаркированныйСписок Тогда
ОбработатьСписок(Параграф);
ИначеЕсли Параграф.ТипПараграфа = ТипПараграфа.НумерованныйСписок Тогда
ОбработатьСписок(Параграф);
Иначе
Продолжить;
КонецЕсли;
КонецЦикла;

```

[Копировать в буфер обмена](#)

Обработка элементов форматированного документа, входящего в состав параграфа, может оказаться полезной в том случае, если нужно, например, убрать все оформление в виде курсива или полужирного шрифта.

Пример:

```

Полужирный = Новый Шрифт(, , Истина);
Курсив = Новый Шрифт(, , Истина);
Обычный = Новый Шрифт;
Для каждого Элемент Из Параграф.Элементы Цикл
Если ТипЗнч(Элемент) = Тип("ПереводСтрокиФорматированногоДокумента") Тогда
Продолжить;
КонецЕсли;
Если Элемент.Шрифт = Полужирный ИЛИ Элемент.Шрифт = Курсив Тогда
Элемент.Шрифт = Обычный;
КонецЕсли;
КонецЦикла;

```

[Копировать в буфер обмена](#)

При программной работе с текстом также могут помочь методы **ПолучитьЭлементы()** и **СформироватьЭлементы()**. Эти методы возвращают массив элементов форматированного документа. Чтобы увидеть разницу в работе этих методов, рассмотрим пример форматированного документа, содержащего следующую строку: **012 456 890**. Строка состоит из 10 символов, где на месте символов 5 и 7 расположены пробелы. Необходимо получить элементы, которые расположены между позициями 2 и 9 этого документа.

[Копировать в буфер обмена](#)

```

ПозицияНачала = Документ.ПолучитьЗакладкуПоПозиции(2);
ПозицияОкончания = Документ.ПолучитьЗакладкуПоПозиции(9);
Результат = Документ.ПолучитьЭлементы(ПозицияНачала, ПозицияОкончания);
Для каждого Элемент Из Результат Цикл
Сообщить("Текст - " + Элемент.Текст);
КонецЦикла;
Результат = Документ.СформироватьЭлементы(ПозицияНачала, ПозицияОкончания);
Для каждого Элемент Из Результат Цикл
Сообщить("Текст - " + Элемент.Текст);
КонецЦикла;

```

Разница в работе этих методов будет в том, что метод **ПолучитьЭлементы()** возвращает все элементы, которые попадают между заданными закладками (включая все граничные элементы), т. е. рассматриваемом примере будет выведен следующий текст:

[Копировать в буфер обмена](#)

Текст – 012 456 890

А при работе метода **СформироватьЭлементы()** вывод будет следующий:

[Копировать в буфер обмена](#)

Текст – 2 456 8

Другими словами, при использовании метода **СформироватьЭлементы()** система формирует набор элементов таким образом, чтобы он включал в себя только то содержимое форматированного документа, которое ограничено закладками.

Нужно помнить, что метод **СформироватьЭлементы()** изменяет форматированный документ во время своего вызова и если (в рассматриваемом примере) после вызова метода **СформироватьЭлементы()** снова вызвать метод **ПолучитьЭлементы()**, то результат работы будет такой:

[Копировать в буфер обмена](#)

Текст – 2 456 8

Такой результат будет даже в том случае, если полученный, после вызова **СформироватьЭлементы()**, набор элементов никак не изменялся.

7.2.7. Планировщик

7.2.7.1. Общая информация

Объект **Планировщик** (далее планировщик) предназначен для визуализации различных операций, привязанных к дате и времени, с возможностью повторения этих событий с определенной периодичностью. Планировщик предназначен для отображения информации и не хранит каких-либо данных в информационной базе.

Общая схема работы выглядит следующим образом:

- Из информационной базы получаются данные, которые следует отобразить;
- Эти данные (в виде элементов) отображаются в планировщике;
- Пользователь редактирует элементы;
- Данные измененных элементов актуализируются в информационной базе.

Планировщик предоставляет возможность переопределять формы, используемые для редактирования элементов.

7.2.7.2. Устройство планировщика

7.2.7.2.1. Измерения

В планировщике всегда присутствует одно измерение – это **шкала времени**. Также прикладной разработчик может создать дополнительные **измерения** (коллекция **Планировщик.Измерения**), которые позволят конкретизировать назначение каждого элемента. Измерения не имеют приоритета друг перед другом, но на форме они «вкладываются» друг в друга согласно порядку добавления. Каждое измерение характеризуется значением произвольного типа, по которому в дальнейшем можно выполнять поиск требуемого измерения.

[Копировать в буфер обмена](#)

```
ИзмерениеЗал = Планировщик.Измерения.Добавить("Зал");
ЭлементИзмерения = ИзмерениеЗал.Элементы.Добавить(Перечисления.ВидЗала.Мужской);
ЭлементИзмерения = ИзмерениеЗал.Элементы.Добавить(Перечисления.ВидЗала.Женский);
ИзмерениеМастер = Планировщик.Измерения.Добавить("Мастер");
ЭлементИзмерения = ИзмерениеМастер.Элементы.Добавить("Иванов");
ЭлементИзмерения = ИзмерениеМастер.Элементы.Добавить("Петров");
ЭлементИзмерения = ИзмерениеМастер.Элементы.Добавить("Сидорова");
```

В этом примере планировщику добавляется два измерения:

- **Зал** – описывает тип зала некоторой парикмахерской. Для определения типа зала служит перечисление **ВидЗала** с двумя значениями: **Мужской** и **Женский**.
- **Мастер** – фамилия мастера, работающего в том или ином зале. Типом значения будет являться строка с фамилией мастера.

На форме планировщик будет выглядеть следующим образом:

	4 июня 2014 г.						6 июня 2014 г.						
	Мужской зал			Женский зал			Мужской зал			Женский зал			
	Иванов	Петров	Сидорова	Иванов	Петров	Сидорова	Иванов	Петров	Сидорова	Иванов	Петров	Сидорова	
00:00							00:00						
01:00							01:00						
02:00							02:00						
03:00							03:00						
04:00							04:00						
05:00							05:00						
06:00							06:00						
07:00							07:00						
08:00							08:00						
09:00							09:00						
10:00							10:00						
11:00							11:00						
12:00							12:00						
13:00							13:00						
14:00							14:00						
15:00							15:00						
16:00							16:00						
17:00							17:00						
18:00							18:00						

Рис. 167. Два измерения

Если поменять порядок добавления измерений, то на форме это будет выглядеть следующим образом:

	4 июня 2014 г.						6 июня 2014 г.						
	Иванов		Петров		Сидорова		Иванов		Петров		Сидорова		
	Мужской зал	Женский зал	Мужской зал	Женский зал	Мужской зал	Женский зал	Мужской зал	Женский зал	Мужской зал	Женский зал	Мужской зал	Женский зал	
00:00							00:00						
01:00							01:00						
02:00							02:00						
03:00							03:00						
04:00							04:00						
05:00							05:00						
06:00							06:00						
07:00							07:00						
08:00							08:00						
09:00							09:00						
10:00							10:00						
11:00							11:00						
12:00							12:00						
13:00							13:00						
14:00							14:00						
15:00							15:00						
16:00							16:00						
17:00							17:00						
18:00							18:00						
19:00							19:00						
20:00							20:00						
21:00							21:00						

Рис. 168. Измененный порядок измерений

7.2.7.2.2. Элементы

На «пересечении» измерений и шкалы времени находится **элемент**. Элемент имеет следующие основные характеристики:

- Описание элемента;
- Дата и время начала и окончания действия или события, описываемого элементом;

единицах будет измеряться шкала времени, и сколько таких единиц будет помещаться на шкале перед тем, как произойдет перенос в следующую колонку (или строку). Например, можно установить, что шкала времени будет измеряться в часах и в одной колонке (строке) будет помещаться 6 часов. В этом случае отображение в планировщике одного дня будет состоять из 4 колонок (строк), в зависимости от режима отображения шкалы времени: в сутках 24 часа и 6 часов в одной колонке (строке).

При отображении шкалы времени сверху, изображение будет выглядеть следующим образом:

6 июня 2014 г.	Мужской зал	Иванов	00:00	01:00	02:00	03:00	04:00	05:00
		Петров						
		Сидорова						
	Женский зал	Иванов						
		Петров						
		Сидорова						
6 июня 2014 г.	Мужской зал	Иванов	06:00	07:00	08:00	09:00	10:00	11:00
		Петров						
		Сидорова						
	Женский зал	Иванов						
		Петров						
		Сидорова						
6 июня 2014 г.	Мужской зал	Иванов	12:00	13:00	14:00	15:00	16:00	17:00
		Петров				15:00 Постричь...		
		Сидорова						
	Женский зал	Иванов						
		Петров						
		Сидорова						
6 июня 2014 г.	Мужской зал	Иванов	18:00	19:00	20:00	21:00	22:00	23:00
		Петров						
		Сидорова						
	Женский зал	Иванов						
		Петров						
		Сидорова						

Рис. 171. Кратность шкалы времени

На встроенном языке такое поведение задается следующим образом:

[Копировать в буфер обмена](#)

```
Планировщик.ЕдиницаПериодическогоВарианта = ТипЕдиницыШкалыВремени.Час ;
Планировщик.КратностьПериодическогоВарианта = 6 ;
Планировщик.ШкалаВремени.Положение = ПоложениеШкалыВремени.Верх ;
```

Если заголовки с отметкой часа не нужны над каждым 6-ти часовым фрагментом, то такие перенесенные заголовки можно отключить, присвоив свойство `Планировщик.ОтображатьПеренесенныеЗаголовкиШкалыВремени` значение `Ложь`.

Следующим вариантом настройки шкал времени является изменение текущей шкалы времени или настройка дополнительных шкал времени (свойство `Планировщик.ШкалаВремени.Элементы`). Изменение шкалы времени необходимо в тех случаях, когда стандартная шкала времени не устраивает по каким-то причинам. Например, единицей шкалы времени выбран день, кратность шкалы времени установлена в 7 (неделя), а планировщик отображает интервал времени, больший 1 недели (например, месяц). В этом случае места соприкосновения различных дней, в рамках кратности периодического варианта, не будет визуально никак выделяться на форме:

1 июня 2014 г.									
Мужской зал			Женский зал			Мужской зал			
Иванов	Петров	Сидорова	Иванов	Петров	Сидорова	Иванов	Петров	Сидорова	Сидорова
00:00						00:00			
01:00						01:00			
02:00						02:00			
03:00						03:00			
04:00						04:00			
05:00						05:00			
06:00						06:00			
07:00						07:00			
08:00						08:00			
09:00						09:00			
10:00						10:00			
11:00						11:00			
12:00						12:00			
13:00						13:00			
14:00						14:00			
15:00						15:00			
16:00						16:00			
17:00						17:00			
18:00						18:00			
19:00						19:00			
20:00						20:00			
21:00						21:00			
22:00						22:00			
23:00						23:00			
00:00						00:00			
01:00						01:00			
02:00						02:00			
03:00						03:00			
04:00						04:00			
05:00						05:00			

Граница двух дней.
Практически не заметна.

Рис. 172. Граница двух периодов

Для того чтобы визуально отметить границу, необходимо добавить еще одну шкалу времени. Добавляемая шкала времени должна описывать целиком один день:

[Копировать в буфер обмена](#)

```
ЭлементШкалыВремени = Планировщик.ШкалаВремени.Элементы.Добавить ( ) ;
ЭлементШкалыВремени.Единица = ТипЕдиницыШкалыВремени.День ;
ЭлементШкалыВремени.Кратность = 1 ;
```

В результате граница двух «соседних» дней станет более заметной:

		1 июня 2014 г.							
		Мужской зал			Женский зал				
		Иванов	Петров	Сидорова	Иванов	Петров	Сидорова	Иванов	
00:00	01.06.2014 0:00:00							00:00	08.06.2014 0:00:00
01:00								01:00	
02:00								02:00	
03:00								03:00	
04:00								04:00	
05:00								05:00	
06:00								06:00	
07:00								07:00	
08:00								08:00	
09:00								09:00	
10:00								10:00	
11:00								11:00	
12:00								12:00	
13:00								13:00	
14:00								14:00	
15:00								15:00	
16:00								16:00	
17:00								17:00	
18:00								18:00	
19:00								19:00	
20:00								20:00	
21:00								21:00	
22:00								22:00	
23:00								23:00	
00:00	02.06.2014 0:00:00							00:00	09.06.2014 0:00:00
01:00								01:00	
02:00								02:00	
03:00								03:00	
04:00								04:00	
05:00								05:00	

Появилось
визуальное
разделение двух
"соседних" дней

Рис. 173. Разделение двух периодов

В данном примере можно сделать два улучшения: перенести дату перед временем и сменить формат отображения даты:

```
ЭлементШкалыВремени = Планировщик.ШкалаВремени.Элементы.Добавить();
ЭлементШкалыВремени.Единица = ТипЕдиницыШкалыВремени.День;
ЭлементШкалыВремени.Кратность = 1;
ЭлементШкалыВремени.Формат = "Л=; ДЛФ=DD";
Планировщик.ШкалаВремени.Элементы.Сдвинуть(ЭлементШкалыВремени, -1);
```

[Копировать в буфер обмена](#)

Порядок следования шкал времени определяется их порядком в коллекции. В рассматриваемом примере добавляемая шкала оказывается после шкалы, созданной по умолчанию, поэтому созданная шкала будет сдвинута на 1 позицию ближе к началу коллекции. А с помощью свойства `Формат` задается то, каким образом будет отображаться время начала дня. В результате разделение периодов станет более понятным и очевидным:

		1 июня 2014 г.							
		Мужской зал			Женский зал				
		Иванов	Петров	Сидорова	Иванов	Петров	Сидорова	Иванов	
1 июня 2014 г.	00:00							8 июня	
	01:00								
	02:00								
	03:00								
	04:00								
	05:00								
	06:00								
	07:00								
	08:00								
	09:00								
	10:00								
	11:00								
	12:00								
	13:00								
	14:00								

Рис. 174. Изменен порядок шкал и формат представления

Также следует помнить, что из коллекции `ШкалаВремени` невозможно удалить элемент, если он единственный. Следовательно, вначале необходимо добавить новую шкалу времени, и лишь затем можно удалить автоматически добавленную шкалу времени.

7.2.7.2.4. Периоды отображения

Планировщик может отображать различные периоды времени. Период времени может быть один или несколько, периоды могут иметь произвольную длительность. Для установки периодов отображения служит коллекция `Планировщик.ТекущиеПериодыОтображения`. Периоды отображения следует задавать с учетом времени. Если задать период с помощью конструкции `Планировщик.ТекущиеПериодыОтображения.Добавить(Дата('20140601'), Дата('20140601'))`; то будет отображена первая секунда 1 июня 2014 года. Для того чтобы день окончания периода отображался целиком, следует использовать функцию `КонецДня()` или явным образом указывать время окончания периода.

Отдельного внимания заслуживает взаимодействие заданных периодов отображения и кратности временной шкалы. Все добавленные периоды считаются одной непрерывной шкалой, и переносы шкалы времени обрабатываются именно по этой шкале. Допустим, необходимо отобразить 7-дневные интервалы на временной шкале для следующего набора периодов:

- 1 июня – 5 июня;
- 10 июня – 19 июня;
- 21 июня – 23 июня.

Планировщик настраивается следующим образом:

```
Планировщик.ЕдиницаПериодическогоВарианта = ТипЕдиницыШкалыВремени.День;
Планировщик.КратностьПериодическогоВарианта = 7;
Планировщик.ТекущиеПериодыОтображения.Очистить();
Планировщик.ТекущиеПериодыОтображения.Добавить(Дата('20140601'), КонецДня(Дата('20140605')));
Планировщик.ТекущиеПериодыОтображения.Добавить(Дата('20140610'), КонецДня(Дата('20140619')));
Планировщик.ТекущиеПериодыОтображения.Добавить(Дата('20140621'), КонецДня(Дата('20140623')));
ЭлементШкалыВремени = Планировщик.ШкалаВремени.Элементы.Добавить();
ЭлементШкалыВремени.Единица = ТипЕдиницыШкалыВремени.День;
ЭлементШкалыВремени.Кратность = 1;
ЭлементШкалыВремени.Формат = "Л=; ДЛФ=DD";
Планировщик.ШкалаВремени.Элементы.Удалить(Планировщик.ШкалаВремени.Элементы[0]);
```

[Копировать в буфер обмена](#)

При такой настройке форма планировщика будет выглядеть следующим образом:

	1 июня 2014 г.						12 июня 2014 г.						19 июня 2014 г.		
	Мужской зал			Женский зал			Мужской зал			Женский зал			Мужской зал		
	Иванов	Петров	Сидорова	Иванов	Петров	Сидорова	Иванов	Петров	Сидорова	Иванов	Петров	Сидорова	Иванов	Петров	Сидорова
1 июня 2014 г.															
2 июня 2014 г.															
3 июня 2014 г.															
4 июня 2014 г.															
5 июня 2014 г.															
10 июня 2014 г.															
11 июня 2014 г.															

Рис. 175. Перенос шкалы времени с дискретными интервалами

На рис. 175 видно, что первые 7 дней целиком включили первый интервал (5 дней, с 1 июня по 6 июня) и 2 дня от второго интервала. Вторые 7 дней включили 7 дней второго интервала и третьи 7 дней включили остаток второго интервала и целиком третий интервал.

Еще одним способом задания отображения прерывистых интервалов служит задание свойств `ОтступСНачалаПереносаШкалыВремени` и `ОтступСКонцаПереносаШкалыВремени`. Данные свойства задаются в размерности свойства `ЕдиницаПериодическогоВарианта`. Другими словами, если единицей периодического варианта является день, то и отступ с начала или конца шкалы времени будет вычисляться в днях. Например, необходимо отобразить в планировщике июнь 2014 года таким образом, чтобы в каждом 7-дневном интервале отображались только первые 5 (или менее) дней. Для этого используется следующий код настройки планировщика (в примере не будет использоваться измерение `Мастер` из предыдущих примеров):

[Копировать в буфер обмена](#)

```
Планировщик.ЕдиницаПериодическогоВарианта = ТипЕдиницыШкалыВремени.День;
Планировщик.КратностьПериодическогоВарианта = 7;
Планировщик.ТекущиеПериодыОтображения.Очистить();
Планировщик.ТекущиеПериодыОтображения.Добавить(НачалоМесяца(Дата('20140601')), КонецМесяца(Дата('20140601')));
ЭлементШкалыВремени = Планировщик.ШкалаВремени.Элементы.Добавить();
ЭлементШкалыВремени.Единица = ТипЕдиницыШкалыВремени.День;
ЭлементШкалыВремени.Кратность = 1;
ЭлементШкалыВремени.Формат = "Л="; ДЛФ="DD";
Планировщик.ШкалаВремени.Элементы.Удалить(Планировщик.ШкалаВремени.Элементы[0]);
```

При такой настройке форма планировщика будет выглядеть следующим образом:

	1 июня 2014 г.		8 июня 2014 г.		15 июня 2014 г.		22 июня 2014 г.		29 июня 2014 г.	
	Мужской зал	Женский зал	Мужской зал	Женский зал	Мужской зал	Женский зал	Мужской зал	Женский зал	Мужской зал	Женский зал
1 июня 2014 г.										
2 июня 2014 г.										
3 июня 2014 г.										
4 июня 2014 г.										
5 июня 2014 г.										

Рис. 176. Отступ с конца переноса шкалы времени

В данном случае весь июнь разбивается на следующие периоды:

- 1 июня – 7 июня;
- 8 июня – 14 июня;
- 15 июня – 21 июня;
- 22 июня – 28 июня;
- 29 июня – 30 июня.

Затем из каждого интервала будет удалено то количество дней (с начала и конца интервала), которые заданы в свойствах `ОтступСНачалаПереносаШкалыВремени` и `ОтступСКонцаПереносаШкалыВремени` (соответственно) и получившиеся интервалы будут отображены. Последний интервал поглощается отступом целиком, но начало интервала все равно будет отображено (по аналогии с поведением при задании текущих периодов отображения).

Прикладной разработчик имеет возможность самостоятельно изменять периоды отображения при навигации пользователя по планировщику (например, с помощью колеса мыши). Для этого следует обработать событие `ПриСменеТекущегоПериодаОтображения`. Если параметр `СтандартнаяОбработка` в обработчике этого события будет установлен в значение `Ложь`, то периоды отображения, заданные в параметре `ТекущиеПериодыОтображения`, не будут применены и не произойдет смена периода отображения.

7.2.7.2.5. Переопределение форм редактирования элементов

Для редактирования элементов в планировщике существуют две формы: форма быстрого редактирования и форма редактирования события. Для переопределения форм редактирования используются следующие события:

- **ПередСозданием** – в данном событии можно переопределить форму, которая будет использоваться при создании нового события. По умолчанию используется форма быстрого редактирования события.
- **ПередНачаломБыстрогоРедактирования** – в данном событии можно переопределить форму, которая используется для быстрого редактирования события. По умолчанию используется форма быстрого редактирования события.
- **ПередНачаломРедактирования** – в данном событии можно переопределить форму, которая используется для обычного редактирования события. Событие срабатывает в следующих случаях:
 - В системной форме быстрого редактирования элемента нажата специальная кнопка. Если форма быстрого редактирования переопределена, то прикладному разработчику необходимо самому реализовать механизм открытия формы обычного редактирования события. В этом случае событие **ПередНачаломРедактирования** не используется.
 - В контекстном меню элемента выбрана команда **Редактировать**;
 - На элементе, отображаемом в планировщике, выполнен двойной щелчок мыши.

Для того чтобы получить параметры редактирования события следует воспользоваться свойством **ВыделенныеЭлементы** у элемента формы, отображающего планировщик.

Собственно переопределение формы выполняется следующим образом:

1. В обработчике нужного события открывается требуемая форма;
2. Параметр используемого обработчика **СтандартнаяОбработка** устанавливается в значение **Ложь**.

7.2.7.2.6. Повторяющиеся элементы

Элементы (см. [здесь](#)) бывают однократные и повторяющиеся. Повторяющиеся элементы обладают расписанием (или правилами повторения элементов). Расписание задается с помощью одноименного свойства (**Расписание**). Несколько элементов, образованных в результате задания расписания, образуют серию элементов. В рамках серии возможно как изменение параметров всей серии, так и изменение параметров одного, конкретного, элемента серии.

Повторяющиеся элементы по-разному представлены в коллекции элементов и при отображении на форме. В коллекции элементов повторяющийся элемент будет представлен в единственном числе и у этого элемента будет заполнено свойство **Расписание**.

При отображении на форме будут присутствовать все повторяющиеся элементы, однако возможность их перебрать система не предоставляет. В тоже время, при попытке отредактировать какой-либо элемент, образующий серию, свойство элемента формы **ВыделенныеЭлементы** будет содержать коллекцию из элементов планировщика, которые образуют серию для элемента с расписанием. У таких элементов не будет установлено свойство **Расписание**, но будет заполнено свойство **Родитель**.

Свойство **Родитель** будет ссылаться на элемент, расписание которого привело к созданию серии, часть которой предполагается редактировать.

Если в форме планировщика выполнить ручное редактирование одного из элементов серии повторяющихся элементов, то в родительском событии серии станет заполнено свойство **ЗамещающиеЭлементы**. В данной коллекции будут указаны те элементы серии, для которых были изменены параметры (относительно родительского события). При создании структуры хранения данных об элементах, следует помнить о том, что в этой структуре необходимо хранить не только родительские элементы, но и элементы, для которых некоторые параметры отличаются от заданных для серии в целом.

Если редактируется элемент из серии, который является замещающим, то свойство **ДатаЗамещения** указывает на ту дату и время, которые были получены в результате первоначального создания серии системой.

7.2.7.3. Общая схема редактирования данных планировщика

Как было сказано ранее (см. [здесь](#)), планировщик не является средством хранения данных. Он является средством их (данных) отображения. Поэтому для хранения информации о событиях в информационной базе должна быть создана соответствующая структура данных, например, справочник.

Общая схема работы выглядит следующим образом:

1. Определяются общие параметры планировщика, количество измерений, настройки шкалы времени и т. д.
 2. Определяется текущий период отображения планировщика;
 3. В соответствии с установленным периодом отображения из информационной базы получаются данные об отображаемой информации, и выполняется создание нужных событий в элементе формы. В качестве значения свойства **Значение** для элемента планировщика используется ссылка на элемент справочника, хранящего данные.
 4. При редактировании или удалении элементов запоминаются свойство **Значение** для событий, которые были изменены или удалены. Это можно сделать с помощью реквизитов формы, в которых будут сохраняться значения свойства **Значение** измененных или удаленных элементов.
- Новые элементы можно будет определить по незаполненному свойству **Значение** у элемента. Для этого необходимо перебрать всю коллекцию **Планировщик.Элементы**.
5. При смене периода отображения выполняется создание элементов, которые принадлежат новому периоду отображения.
 6. При успешном завершении работы с планировщиком, выполняется синхронизация данных планировщика и справочника, в котором хранятся данные элементов.

7.3. Параметры формы

Параметры формы (закладка **Параметры**) служат двум целям:

- Описать набор данных, которые будут влиять на открытие формы (параметризация формы). Для этого нужно перечислить все необходимые параметры и указать их типы.
- Определить параметры, которые будут влиять на ключ уникальности формы. Для этого необходимо установить свойство **Ключевой параметр** у тех параметров, которые должны участвовать в формировании ключа уникальности формы. При попытке открыть форму система производит поиск существующей формы с помощью сформированного ключа уникальности формы. Если в системе существует форма с полученным ключом уникальности, возвращается именно эта форма; если нет – создается новая форма.

При вызове формы значения параметров, созданных разработчиком, можно указывать в структуре параметров наряду с системными параметрами форм (если таковые имеются).

Параметры формы можно передать в форму в момент ее создания. Анализ переданных параметров можно выполнить в событии **ПриСозданииНаСервере()** (коллекция **Параметры** является свойством объекта **УправляемаяФорма**):

```
// В месте вызова.
// Формируем параметр формы.
Параметры = Новый Структура();
Параметры.Вставить("Важность", ПредопределенноеЗначение("Перечисление.Важность.Важно"));
// Открываем форму с указанием параметров.
ОткрытьФорму("ОбщаяФорма.ФормаПросмотра", Параметры);
...
// В модуле формы.
&НаСервере
```

```
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
Если Параметры.Важность = Перечисления.Важность.Важно Тогда
...
КонецЕсли;
КонецПроцедуры
```

ВНИМАНИЕ! После вызова обработчика события `ПриСозданииНаСервере` все неключевые параметры формы удаляются из коллекции `Параметры`.

СОВЕТ. Неключевые параметры формы, необходимые для дальнейшей работы, нужно сохранять в данных формы.

7.3.1. Стандартные параметры формы

Для того чтобы поддерживать автоматическое взаимодействие между формами, система предоставляет ряд стандартных параметров, которые используются для управления формами при их открытии. С помощью этих параметров системой реализуется в полях форм выбор из форм выбора, открытие форм объектов, работа стандартных команд и т. д. То есть они обеспечивают различные заложенные в систему сценарии работы интерфейса. Но разработчик также может использовать эти параметры во встроенном языке, передавая их при вызове метода `ОткрытьФорму()`.

Перечень стандартных параметров форм в зависимости от вида расширения формы можно посмотреть в разделах [Встроенный язык – Интерфейс \(управляемый\) – Управляемая форма – Расширение...](#) встроенной справки.

7.3.2. Пример работы с параметрами формы

Для демонстрации работы параметров формы рассмотрим реализацию выбора элемента в поле ввода. Сутью примера будет реализация механизма выбора элемента из списка на встроенном языке.

К моменту начала работы с примером нужно иметь конфигурацию, обладающую следующими свойствами:

- имеется справочник `Товары` с иерархией групп и элементов;
- имеется справочник `Аналоги` с реквизитом `ВыбранныйТовар` типа `СправочникСсылка.Товары`;
- оба справочника имеют формы элементов.

Теперь реализуем в этой конфигурации все механизмы, которые использует платформа для выбора элемента из списка, на встроенном языке. При этом мы увидим:

- как происходит использование стандартных параметров формы;
- каким образом их использует сама система;
- как их может использовать разработчик.

Добавим дополнительный параметр, который будет управлять закрытием формы выбора после выбора элемента. Назовем этот параметр `ЗакрыватьПослеВыбора` (тип `Булево`). Добавим его в качестве параметра формы `ФормаВыбора` справочника `Товары`.

Для того чтобы открыть форму выбора элемента, необходимо в форме элемента справочника `Аналоги` создать обработчик события `НачалоВыбора` у элемента формы `ВыбранныйТовар`.

[Копировать в буфер обмена](#)

```
&НаКлиенте
Процедура ВыбранныйТоварНачалоВыбора(Элемент, СтандартнаяОбработка)
СтандартнаяОбработка = Ложь;
ПараметрыВыбора = Новый Структура;
ПараметрыВыбора.Вставить("РежимВыбора", Истина);
ПараметрыВыбора.Вставить("ВыборГруппИЭлементов", ИспользованиеГруппИЭлементов.Элементы);
ПараметрыВыбора.Вставить("РазрешитьВыборКорня", Ложь);
ПараметрыВыбора.Вставить("ТекущаяСтрока", Объект.ВыбранныйТовар);
ПараметрыВыбора.Вставить("ЗакрыватьПослеВыбора", Ложь);
ОткрытьФорму("Справочник.Товары.ФормаВыбора", ПараметрыВыбора, Элементы.ВыбранныйТовар);
КонецПроцедуры
```

Следует отдельно остановиться на третьем параметре метода `ОткрытьФорму()`. Этот параметр определяет, кто будет владельцем формы выбора и кому будет приходиться оповещение о сделанном выборе. В данном случае мы указали владельцем формы выбора сам элемент формы, но также мы можем указать этим параметром и саму форму. В этом случае будет необходимо реализовывать обработчик `ОбработкаВыбора` модуля формы и в нем решать, в какой реквизит формы помещать выбранные данные.

ПРИМЕЧАНИЕ. Если мы не будем реализовывать обработчик события `НачалоВыбора`, то его действия выполнит сама система. Это справедливо и для всех дальнейших обработчиков, которые использованы в примере.

Теперь необходимо обработать переданные параметры в форме выбора. Сделаем это в обработчике `ПриСозданииНаСервере()` модуля формы выбора.

[Копировать в буфер обмена](#)

```
&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
СтандартнаяОбработка = Ложь;
Элементы.Список.ВыборГруппИЭлементов = Параметры.ВыборГруппИЭлементов;
Элементы.Список.РазрешитьВыборКорня = Параметры.РазрешитьВыборКорня;
Элементы.Список.ТекущаяСтрока = Параметры.ТекущаяСтрока;
ЗакрыватьПриВыборе = Параметры.ЗакрыватьПослеВыбора;
КонецПроцедуры
```

Для того чтобы проверить работоспособность установленных нами параметров формы, установим, с помощью конфигуратора, у таблицы формы выбора `Список` свойство `ВыборГруппИЭлементов` в значение `Группы` (без применения параметра не будет доступен выбор элементов справочника).

ПРИМЕЧАНИЕ. Если у таблицы `Список`, отображающей список товаров, свойство `РежимВыбора` не будет установлено в значение `Истина`, то выбор товаров будет недоступен.

Теперь нам необходимо обработать выбор желаемого элемента в форме выбора. Для этого нужно определить обработчик события `ВыборЗначения` таблицы формы.

[Копировать в буфер обмена](#)

```
&НаКлиенте
Процедура СписокВыборЗначения(Элемент, СтандартнаяОбработка, Значение)
СтандартнаяОбработка = Ложь;
ОповеститьОВыборе(Значение);
КонецПроцедуры
```

Нам осталось реализовать обработку выбора элемента в самом поле ввода. Для этого необходимо обработать событие `ОбработкаВыбора` нашего поля ввода `ВыбранныйТовар`.

[Копировать в буфер обмена](#)

```
&НаКлиенте
Процедура ВыбранныйТоварОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)
СтандартнаяОбработка = Ложь;
Объект.ВыбранныйТовар = ВыбранноеЗначение;
```

Мы самостоятельно реализовали системный механизм выбора значения в поле ввода на форме.

ВНИМАНИЕ! Данный пример не является законченным. Его единственным назначением является демонстрация механизмов работы с параметрами формы.

Если при создании параметров (обработчик `ВыбранныйТоварНачалоВыбора()`) заменить строку:

```
ПараметрыВыбора.Вставить("ЗакрыватьПослеВыбора", Истина);
```

[Копировать в буфер обмена](#)

на строку:

```
ПараметрыВыбора.Вставить("ЗакрыватьПослеВыбора", Ложь);
```

[Копировать в буфер обмена](#)

то форма выбора перестанет закрываться после того, как будет осуществлен выбор. Это можно использовать, например, для реализации формы подбора (выбор нескольких товаров без закрытия формы выбора).

7.4. Команды формы

7.4.1. Общая информация

Действия в форме выполняются с помощью команд формы. Сами команды служат лишь описанием выполняемых действий. Для того чтобы команда стала выполнять свою функцию, она должна быть привязана к элементу формы (типа [Кнопка](#)). Можно выделить несколько групп команд, которые присутствуют в форме:

- **Команды, создаваемые разработчиком** в процессе проектирования формы. Для них необходимо создавать обработчик непосредственно в модуле формы.
- **Стандартные команды**, которые предоставляются расширением основного реквизита формы и расширениями реквизитов, которые являются списками (например, табличная часть объекта, динамический список, набор записей регистра сведений и т. д.), если с этим реквизитом есть связанный элемент формы.
- **Глобальные команды** – команды, предоставленные глобальным командным интерфейсом (подробнее про командный интерфейс см. [здесь](#)). Такие команды могут быть непараметризованными и параметризованными. Параметризованные глобальные команды будут предоставлены форме только в том случае, если на форме есть источники параметров с соответствующими типами.

Доступность стандартных команд формы и элементов формы определяется свойством [Состав команд](#) соответствующего элемента формы.

Команды, которые предоставляет глобальный командный интерфейс (закладка [Глобальные команды](#)), могут быть размещены в любом месте формы, точно так же, как и команды формы.

В свойстве [Действие](#) указывается обработчик, который реализует действие, выполняемое командой. Если обработчик не задан, то команда будет недоступна для использования. К выбору в этом поле доступны только процедуры и функции без параметров, которые являются клиентскими (подробнее см. [здесь](#)).

Если команда изменяет данные формы, то следует указать на это установкой свойства [Изменяет сохраняемые данные](#). Тогда при попытке выполнения команды будут выполняться следующие действия:

- Выполнена попытка блокировки связанного реквизита формы. Если попытка закончится неудачей – команда не будет выполнена.
- Установлен признака изменения у формы (флаг [Модифицированность](#)).

Если для команды установлено свойство [Сохраняемые данные](#) и форма находится в режиме [Только просмотр](#), то все элементы формы, связанные с этой командой, также будут находиться в состоянии [Только просмотр](#).

Кроме того, исполнение параметризованной глобальной команды, для которой параметр предоставляется реквизитом с установленным свойством [Сохраняемые данные](#), для нового и несохраненного объекта приведет к попытке записать объект. При этом пользователю будет задан вопрос о необходимости записи. Если ответ будет отрицательный – команда не будет выполнена.

ПРИМЕЧАНИЕ. При автоматическом заполнении командных панелей и контекстных меню, для которых указан источник команд, стандартные команды не добавляются, если в данном элементе есть кнопки, добавленные вручную с такими же командами. Данная логика не распространяется на команды, добавляемые из фрагмента глобального командного интерфейса.

Для упрощения разработки различных диалогов в стандартные команды формы добавлены команды [Да](#), [Нет](#), [OK](#), [Отмена](#), [Повторить](#), [Прервать](#), [Игнорировать](#). Если такая команда добавлена в форму, то при нажатии пользователем этой кнопки будут выполнены следующие действия:

- Для формы, открытой в модальном режиме, выполняется закрытие формы и возвращается соответствующее значение типа [КодВозвратаДиалога](#).
- Для формы, открытой в немодальном режиме, выполняется только закрытие формы.

При формировании имени обработчика выполнения команды будет использоваться свойство команды [Имя](#).

[Использование](#) – подробнее см. [здесь](#).

[Функциональные опции](#) – определяет, с какими функциональными опциями связан данный реквизит формы. Подробнее про функциональные опции см. [здесь](#).

7.4.2. Особенности применения различных команд

Для расширений управляемой формы для объекта, справочника, документа, плана видов характеристик, бизнес-процесса, задачи и записи регистра сведений существует стандартная команда [Показать в списке](#), которая предназначена для отображения формы списка соответствующего объекта с позиционированием открываемого списка на объекте, из которого выполняется команда. В интерфейсе команда будет недоступна, если ключевой реквизит формы равен пустой ссылке (или пустому ключу записи), т. е. редактируется новый объект.

Если команда используется в форме документа, то для такой команды доступно свойство [Параметр](#), в котором можно выбрать форму, которая будет использоваться для отображения списка. Для выбора доступен собственно документ и все журналы, в состав которых входит используемый документ. Кнопка, связанная с командой [ПоказатьВСписке](#), отображается на форме вне зависимости от заполненности свойства [Параметр](#). Если значение свойства [Параметр](#) задано, то в качестве заголовка по умолчанию у кнопки, связанной с командой [ПоказатьВСписке](#), используется представление списка соответствующего объекта конфигурации (документа или журнала).

При автоматическом заполнении командной панели формы документа анализируются журналы, в которые входит документ, при этом на эти журналы текущий пользователь должен иметь право [Просмотр](#) (см. [здесь](#)) и они (журналы) должны быть включены функциональными опциями. Если есть хотя бы один такой журнал, то в командной панели вместо кнопки [Показать в списке](#) размещается группа кнопок с заголовком [Показать в списке](#). Подменю содержит:

- Команду [ПоказатьВСписке](#), для которой свойство [Параметр](#) заполняется ссылкой на документ, в форме которого формируется командная панель;
- Команды [ПоказатьВСписке](#) (по одной на каждый подходящий журнал), для которых свойство [Параметр](#) заполняется ссылкой на соответствующий журнал документов.

Если нет доступных журналов, то в командной панели размещается команда [ПоказатьВСписке](#), у кнопки которой свойство [Параметр](#) не задано.

Для отображения используется основная форма списка (с учетом возможности переопределения, подробнее см. [здесь](#)) соответствующего объекта конфигурации.

Если необходимо обеспечить функциональность команды [ПоказатьВСписке](#) в рамках реализации какого-то другого алгоритма, то для этого следует использовать метод `ВыполнитьПереход()`, существующий для объекта [УправляемаяФорма](#).

Пример реализации:

```
Форма = ПолучитьФорму("Справочник.Контрагенты.ФормаСписка");
Форма.ВыполнитьПереход(Объект.Ссылка);
```

[Копировать в буфер обмена](#)

Если необходимо каким-то образом обработать открытие формы с позиционированием на нужном объекте, то это можно сделать с помощью обработчика события `ОбработкаПерехода` (событие вызывается для управляемой формы). Если после исполнения обработчик события `ОбработкаПерехода` требуется выполнить стандартные действия (значение параметра `СтандартнаяОбработка` равно значению `Истина`), то у первой таблицы управляемой формы, связанной с основным реквизитом формы типа `ДинамическийСписок` (см. [здесь](#)) текущая строка устанавливается в значение, переданное в качестве параметра `Объект` обработчика события. Если основным реквизитом открываемой формы выступает не динамический список – никаких действий выполнено не будет.

7.5. Модуль формы

Модуль формы состоит из набора процедур и функций. Допускаются переменные и тело модуля.

Каждая процедура, функция или объявление переменной модуля формы должны предваряться одной из следующих директив компиляции:

- `НаКлиенте` – означает, что метод выполняется на стороне клиента, а переменная существует все время жизни клиентской части формы.

Из клиентского метода допустимыми являются вызовы любых методов.

- `НаСервере` – означает, что метод выполняется на стороне сервера, а переменная существует только во время вызова выполнения серверного вызова.

Для серверных методов допустимыми являются вызовы серверных, серверных внеконтекстных и клиент-серверных внеконтекстных методов.

- `НаСервереВнеКонтекста` – означает, что метод исполняется на сервере вне контекста формы. Переменные не могут предваряться такой директивой компиляции.

В таких методах недоступен контекст формы. При вызове этих методов не выполняется передача данных формы на сервер и обратно. Применение внеконтекстных методов позволяет существенно уменьшить объем передаваемых данных при вызове серверной процедуры из среды клиентского приложения.

Из серверных внеконтекстных методов формы допускается вызов серверных методов общих модулей.

- `НаКлиентеНаСервереВнеКонтекста` – означает, что метод исполняется как на клиенте, так и на сервере, вне контекста формы. Переменные не могут предваряться такой директивой компиляции.

Также такой метод не имеет доступа к переменным модуля формы.

Из клиент-серверного внеконтекстного метода можно вызывать методы неглобальных серверных общих модулей, а также методы неглобальных общих модулей с флажками `Сервер` и `Клиент (управляемое приложение)`.

Отсутствие директивы компиляции перед процедурой означает использование директивы по умолчанию. Директивой по умолчанию является `НаСервере`.

При передаче управления с клиента на сервер с помощью **контекстного вызова** нужно учитывать следующую особенность: перед началом вызова данные формы передаются на сторону сервера, затем происходит выполнение серверного вызова, и потом данные формы передаются обратно на клиента. Это может занимать достаточно существенное время. В то же время **внеконтекстный серверный вызов** не выполняет таких преобразований, поэтому выполняется быстрее.

В программном модуле формы (т. е. фрагменте кода, размещенном вне процедур и функций) допустимо использование инструкций препроцессора для явного выделения участков кода инициализации соответствующих переменных.

ВНИМАНИЕ! В серверной переменной формы невозможно сохранить данные между двумя вызовами серверной стороны формы.

Не допускается использование нескольких директив компиляции перед одним методом или переменной. Не допускается наличие одноименных методов или переменных, отличающихся только директивами компиляции.

Обработчики команд формы, созданных разработчиком, могут располагаться только в клиентских методах модуля формы.

В модуле формы рекомендуется использовать директивы препроцессора только внутри процедур и функций.

ПРИМЕЧАНИЕ. Для понимания результата при «пересечении» инструкциями препроцессора границ процедур следует учитывать, что обработка инструкций препроцессора выполняется до обработки директив компиляции.

Полный перечень директив компиляции и инструкций препроцессора см. [здесь](#).

Приведем пример использования директив компиляции:

```
&НаСервере
Перем СервернаяПеременная;
&НаКлиенте
Перем КлиентскаяПеременная;
&НаСервере
Процедура Серверная()
    Сообщить(СервернаяПеременная);
КонецПроцедуры
&НаКлиенте
Процедура Команда1Выполнить()
    Сообщить(КлиентскаяПеременная);
    Серверная();
КонецПроцедуры
#Если Сервер Тогда
    СервернаяПеременная = "Сервер";
#КонецЕсли
#Если НаКлиенте Тогда
    КлиентскаяПеременная = "Клиент";
#КонецЕсли
```

[Копировать в буфер обмена](#)

7.6. Элементы формы

Иерархия элементов формы определяет внешний вид и состав отображаемых на форме элементов управления. Существует несколько типов элементов формы:

- форма – собственно форма, корневой элемент дерева элементов;
- поле формы;
- декорация формы;
- таблица формы;

- кнопка формы;
- группа формы;
- дополнение элемента формы.

Элементы формы типа поле формы и таблица формы всегда связаны с данными формы. Если связанный реквизит не указан, или он недоступен на клиенте ввиду ограничения по правам, или исключен из состава (свойства [Просмотр](#) и [Редактирование](#) реквизитов формы), поля не будут видно на форме, и оно будет автоматически удалено при создании формы в режиме исполнения.

7.6.1. Общие свойства элементов формы

Данный раздел описывает общие свойства элементов формы. Свойства, специфичные для конкретных элементов, будут описаны ниже.

7.6.1.1. Группа свойств «Основные»

Свойство [Положение заголовка](#) определяет, каким образом будет выводиться заголовок элемента. Заголовком является синоним данных, связанных с выбранным элементом формы, если не указано свойство [Заголовок](#) у элемента формы. Заголовок всегда оканчивается символом ":" (добавляется системой автоматически).

При добавлении элемента в свойстве [Данные](#) необходимо указать ссылку на реквизит формы, с которым этот элемент связан. Если элемент формы не связан с реквизитом формы, то он не будет отображен на форме. Если на форму добавляется кнопка, то в свойстве [Команда](#) необходимо указать ссылку на команду, которая будет выполняться при нажатии кнопки. Если кнопка не связана с командой, то она не будет отображена на форме.

Элемент формы может находиться в режиме [Только просмотр](#) (запрещающем любые изменения), если у него установлено свойство [Только просмотр](#) (в конфигураторе или программно), или у группы, в которую он входит, установлено свойство [Только просмотр](#), или связанный с ним реквизит формы имеет признак [Сохраняемые данные](#) и у формы установлен режим [Только просмотр](#).

Для управления видимостью элементов формы существуют два свойства: [Видимость](#) и [Пользовательская видимость](#). Первое свойство можно изменять как в редакторе формы (в конфигураторе), так и программно. Свойство [Пользовательская видимость](#) настраивается только в конфигураторе и задает начальную видимость элемента формы в разрезе ролей. Результирующая видимость элемента формы образуется сложением «по И» свойств [Видимость](#) и [Пользовательская видимость](#) для конкретного пользователя. Описание редактирования свойства [Пользовательская видимость](#) см. [здесь](#). Кроме того, изменяя видимость элементов формы в диалоге [Настройка формы](#), пользователь фактически выполняет изменение свойства [Пользовательская видимость](#) конкретного элемента формы.

Описание свойств [Высота](#) и [Ширина](#) (или [Количество строк](#)) см. [здесь](#).

С помощью свойства [Пропускать при вводе](#) можно задать пропуск элемента формы при обходе формы с помощью клавиши [Enter](#). Если свойство установлено в значение [Да](#), то элемент будет пропущен при обходе по [Enter](#) (но доступен при обходе с помощью клавиши [Tab](#) или указания элемента мышью). Если свойство установлено в значение [Нет](#), то элемент не будет пропущен при обходе с помощью клавиши [Enter](#). Также существует возможность указать автоматическое определение необходимости пропуска элемента формы. Для этого необходимо установить значение этого свойства равным [Авто](#). В этом случае будет использоваться следующий алгоритм работы:

- Элемент типа [Поле](#) будет пропускаться в случае поля вида, поля надписи или поля картинки, а также в зависимости от значения свойства [Отображение предупреждения при редактировании](#) (см. [здесь](#)).
- Элемент типа [Кнопка](#) не будет пропускаться в том случае, если кнопка является кнопкой по умолчанию.
- Элемент типа [Декорация](#) будет пропускаться.
- Элемент типа [Таблица](#) не будет пропускаться.

7.6.1.1.1. Специальные режимы связи элементов с реквизитами

Как уже отмечалось, любой элемент, отображающий какие-либо данные, должен быть связан с реквизитом данных формы. Помимо связи с «обычными» реквизитами, доступны еще несколько специальных режима.

Связь с текущими данными таблиц

Элемент формы может быть связан с реквизитом, представляющим собой колонку таблицы, размещенной на форме. При этом в данном элементе отображаются данные поля текущей строки таблицы. Такая связь допускается как для полей, так и для таблиц, при этом не требуется, чтобы в таблице отображалась соответствующая колонка. Элемент, связанный с текущими данными, может находиться как в режиме [Только просмотр](#), так и в режиме редактирования.

Например, необходимо расположить на форме поле, которое отображает цену для текущей строки товара. Для этого в нужное место иерархии элементов формы добавляем поле (вид поля ввода может быть как [Поле надписи](#), так и [Поле ввода](#)), для которого в качестве данных выбираем (в диалоге [Выбор реквизита](#)) [Объект.Товары.Цена](#). Здесь [Объект](#) – это реквизит формы типа [Документ](#) конкретного вида, [Товары](#) – это табличная часть документа, а [Цена](#) – реквизит табличной части.

Связь с определенной строкой таблицы

Элемент формы может быть связан с реквизитом, представляющим собой колонку указанной строки таблицы, являющейся реквизитом формы. Такая связь допускается как для полей, так и для таблиц, при этом не требуется, чтобы в таблице отображалась соответствующая колонка. Конкретная строка задается индексом строки (например, [ТаблицаЗначений\[1\]](#)) или последовательностью индексов для иерархических таблиц (например, [ТаблицаЗначений\[2\].СписокИтогов\[4\]](#)). Элемент, связанный с текущими данными, может находиться как в режиме [Только просмотр](#), так и в режиме редактирования.

ПРИМЕЧАНИЕ. Данная связь не поддерживается для динамических списков.

Например, необходимо расположить на форме поле, которое отображает значение поля [Долг](#) из 4-й строки таблицы значений [Контрагенты](#). Для этого в нужное место иерархии элементов формы добавляем поле (вид поля ввода может быть как [Поле надписи](#), так и [Поле ввода](#)), для которого в качестве данных выбираем (в диалоге [Выбор реквизита](#)) [Контрагенты\[3\].Долг](#). Здесь [Контрагенты](#) – это реквизит формы типа [ТаблицаЗначений](#), а [Долг](#) – колонка таблицы значений.

Если данные поля ссылаются на отсутствующую строку – редактировать такое поле будет нельзя, даже если выбрано поле ввода, в противном случае изменение, выполненное в поле ввода, будет выполнено и в таблице, на которую ссылается поле ввода.

В примере выше, если пользователь изменит поле ввода, значение из этого поля попадет в колонку [Долг](#) 4-й строки таблицы значений [Контрагенты](#).

Связь с реквизитами через ссылку

Если в данных формы есть реквизиты ссылочных типов (например, [СправочникСсылка](#)), то элемент формы можно связать с реквизитом, полученным по этой ссылке. Данные для таких элементов будут получаться автоматически и обновляться при изменении ссылки. Связь с реквизитами через ссылку может быть любой глубины. Элементы, связанные с такими реквизитами, всегда находятся в режиме [Только просмотр](#).

ПРИМЕЧАНИЕ. Для реквизитов составного типа (включая типы [СправочникСсылка](#), [ДокументСсылка](#) и т. д.) недоступно получение реквизитов через ссылку.

Например, нужно расположить на форме поле, которое отображает артикул текущей строки товара. Для этого нам необходимо добавить в нужное место иерархии элементов формы поле с видом [Поле надписи](#), для которого в качестве данных выбрать (в диалоге [Выбор реквизита](#)) [Объект.Товары.Товар.Артикул](#). Здесь [Объект](#) – это реквизит формы типа [Документ](#) конкретного вида, [Товары](#) – это табличная часть документа, [Товар](#) – это реквизит табличной части, а [Артикул](#) – реквизит справочника [Товары](#).

Связь с итогами коллекций

Элемент формы может быть связан с реквизитами, представляющим собой итоговые значения некоторых коллекций: табличные части, наборы записей, список значений (только количество строк в коллекции). Такими реквизитами могут быть:

- итоги по числовым полям,
- количество строк в коллекции.

Элементы, связанные с такими реквизитами, всегда находятся в режиме [Только просмотр](#).

В качестве примера поместим на форму поле, отображающее суммой итог для табличной части **Товары**. Для этого в качестве данных поля формы укажем **Объект.Товары.ИтогСумма**, где **Объект** – это основной реквизит формы, **Товары** – табличная часть документа, а **ИтогСумма** – специальный реквизит. Связь может быть установлена не только для самого элемента. Также поддерживается установка связи для отображения данных в подвале таблицы и в заголовке закладки (группы вида [Страница](#)).

В зависимости от типа данных, которые отображает элемент формы (поле ввода, кнопка и группа), с помощью свойства **Вид** можно задать способ отображения данных.

7.6.1.2. Группа свойств «Использование»

Подробнее про использование свойства **Быстрый выбор** см. [здесь](#).

Свойства **Связи параметров выбора** см. [здесь](#) и **Параметры выбора** см. [здесь](#). Если значения свойств **Связи параметров выбора** и **Параметры выбора** заданы и в свойствах реквизита объекта метаданных, и в свойствах элемента формы, то значения свойства будут объединены. Объединение будет выполняться «по ИЛИ» (по именам параметров).

Имеется возможность указывать расширенную подсказку для следующих элементов формы:

- Поле;
- Таблица;
- Группа;
- Кнопка;
- Декорация.

Расширенная подсказка задается с помощью свойства **Расширенная подсказка** соответствующего элемента формы. Включается возможность указания данной подсказки с помощью команды **Показать расширенную подсказку** контекстного меню элемента формы. При этом сама расширенная подсказка представляет собой декорацию формы типа **Текст**. Расширенная подсказка позволяет задавать подсказку с применением различного оформления (цвет, шрифт и т. д.). Для этого необходимо выбрать в окне редактирования расширенной подсказки значение переключателя **Форматированная строка**. Сам текст расширенной подсказки задается в свойстве декорации **Заголовок**. Также у вышеперечисленных элементов формы можно задать отображение подсказки: вокруг элемента формы, кнопкой рядом с элементом формы, всплывающим окном или отображение будет выбираться автоматически. Если задан режим отображения подсказки **Кнопка**, то отображаться будет свойство **Заголовок** расширенной подсказки с форматированием. Если задан режим отображения **Всплывающий**, то будет отображаться свойство **Заголовок** расширенной подсказки, но в виде обычной строки (с убраным форматированием).

Если заданы одновременно свойства **Подсказка** и свойство **Заголовок** расширенной подсказки – отображаться будет текст из расширенной подсказки.

7.6.1.3. Группа свойств «События»

В данной группе собраны ссылки на обработчики, которые предоставляет тот или иной элемент формы.

Следует понимать, что вызов какого-либо обработчика события (не обязательно события элемента формы) из кода на встроенном языке не является аналогом вызова этого же обработчика самой платформой, в рамках выполнения последовательности событий, связанных с тем или иным действием пользователя. Так, например, программный вызов обработчика события **ПриЗаписиНаСервере()** модуля формы справочника не означает, что будет открыта транзакция и произойдет запись элемента справочника. При таком вызове будет исполнен только код обработчика, причем, возможно, что код будет исполнен некорректно или в процессе исполнения этого кода возникнут ошибки. Ошибки могут возникнуть потому, что разработчик, который создавал этот обработчик, исходил из того, что этот обработчик будет вызван только самой платформой и при этом платформа же обеспечит некоторое окружение.

7.6.1.4. Картинки элементов

Элементы формы позволяют использовать для своего оформления картинки. Задание картинки возможно двумя способами:

- из конфигуратора,
- программным способом.

В случае если картинка задается программным способом, может быть установлена либо пустая картинка, либо картинка из библиотеки картинок конфигурации.

Если картинка задается из конфигуратора, то возможен еще один вариант – задание картинки из файла на диске (внешняя картинка). Использование таких картинок рекомендуется только в тех случаях, когда разрабатываются внешние отчеты или обработки, которые могут использоваться в различных конфигурациях, и картинка является значимым элементом оформления. В остальных случаях использование таких картинок является нежелательным. Для команд формы и глобальных команд выбрать внешнюю картинку невозможно.

7.6.2. Форма

Описывает визуальные свойства формы. Элемент такого типа всегда существует в единственном числе и находится в корне иерархии элементов. Также в свойствах формы описываются обработчики событий формы.

Режим открытия формы управляется специальным свойством – **Режим открытия окна**.

Данное свойство описывает, каким образом будет открыто окно:

- **Независимый** – форма открывается:
 - Режим **Формы в отдельных окнах**. Форма открывается в немодальном окне, которое является вспомогательным окном приложения.
 - Режим **Формы в закладках**. Форма открывается в отдельной закладке.
 - Режим **Такси**. Форма открывается в рабочей области основного окна приложения.
- **Блокировать окно владельца** – форма открывается в немодальном окне, в режиме, когда работа с формой, из которой инициировано открытие текущей формы, блокируется. Данный режим предназначен для форм, в которых вводится мало информации и работа с которыми не требует длительного времени, например, для ввода элементов справочников, содержащих небольшое количество реквизитов. Этот режим внешне аналогичен модальному открытию формы, однако при открытии из встроенного языка работа модуля не останавливается на время работы открываемой формы. При этом все остальное взаимодействие с формой выполняется как с другими немодальными формами. Такая форма также открывается во вспомогательном окне.

Форма, открытая в таком режиме, не участвует в поиске уже открытых форм. Если попытаться открыть точно такую же форму (даже с параметром уникальности, установленным в значение **Ложь**), то форма, открытая в режиме **Блокировать окно владельца**, не будет найдена и будет открыта новая форма.

Этот режим по умолчанию установлен у следующих форм:

- элемент и группа справочника,

- узел плана обмена,
 - элемент и группа плана видов характеристик,
 - счет,
 - вид расчета,
 - задача,
 - запись независимого регистра сведений.
- **Блокировать весь интерфейс** – аналогично форме, которая открывается в режиме **Блокировать окно владельца**, но блокируется работа не только с родительской формой, но со всем интерфейсом прикладного решения. Такой режим открытия не используется в качестве установки «по умолчанию».

При открытии формы в блокирующем режиме используется следующий алгоритм определения блокируемого окна:

- Если в свойстве **ВладелецФормы** задана форма и эта форма не закрыта, блокируется окно этой формы.
- Если в свойстве **ВладелецФормы** задан элемент формы и эта форма не закрыта, то блокируется окно формы, которой принадлежит этот элемент.
- В остальных случаях (если в свойстве **ВладелецФормы** задано значение **Неопределено** или форма владельца закрыта) блокируется окно, которое является текущим окном клиентского приложения на момент открытия формы.

Свойство **Режим открытия окна** не оказывает влияния на открытие формы, если она открывается в модальном режиме, а также при открытии в уже существующем основном или вспомогательном окне.

Описание правила формирования автоматического заголовка см. [здесь](#).

Свойство формы **Вертикальная прокрутка** определяет, как будут вести себя элементы формы в том случае, если форма будет сжиматься по вертикали. Если для свойства установлено значение **Использовать** или **Авто** (которое для конкретной формы определяется как **Использовать**), то полоса вертикальной прокрутки появится после того, как вертикальный размер элементов формы достигнет размера, который был задан при редактировании формы в конфигураторе (это относится и к элементам с автоматически определенным высотой элемента). Если для свойства установлено значение **ИспользоватьПриНеобходимости**, то перед появлением полосы прокрутки элементы формы будут сжиматься по вертикали до своего минимального размера.

Рассмотрим пример: имеется форма с реквизитом **Список** типа **ДинамическийСписок**. Этот реквизит отображается на форме таблицей **Список**, для которой высота указана в 8 строк таблицы. Если свойство формы **Вертикальная прокрутка** установить в значение **Использовать**, то форма практически не будет сжиматься:

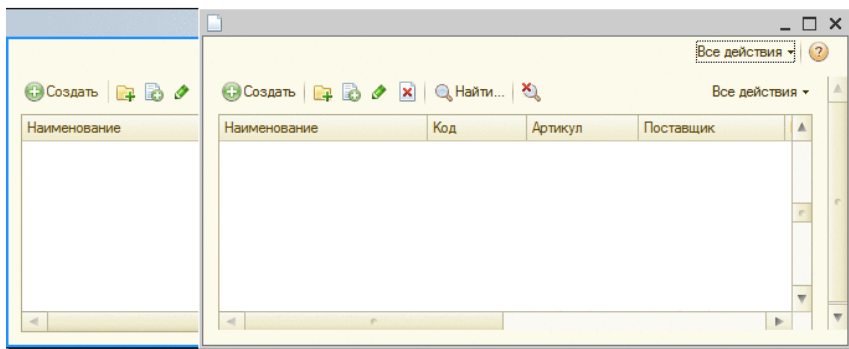


Рис. 177. Вертикальная прокрутка формы – использовать

В этом случае полоса вертикальной прокрутки появилась сразу, как только форма попыталась сжать по вертикали.

Если свойство формы **Вертикальная прокрутка** установить в значение **ИспользоватьПриНеобходимости**, то форма вначале максимально уменьшит свой размер по вертикали, и лишь потом появится вертикальная полоса прокрутки:

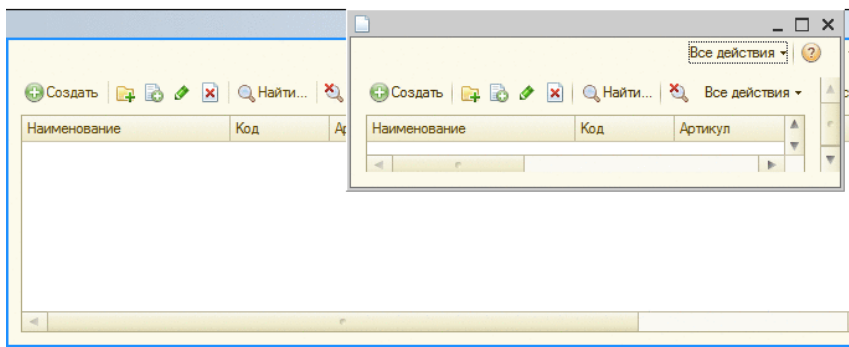


Рис. 178. Вертикальная прокрутка формы – использовать по необходимости

При установке свойства в значение **Авто**, поведение формы определяется системой на основании типа основного реквизита формы:

- Если основной реквизит формы имеет тип **ДинамическийСписок**, или основной реквизит формы имеет тип **ОтчетОбъект**, и для расширения формы отчета заполнено свойство **РезультатОтчета**, то значение **Авто** трактуется как значение **ИспользоватьПриНеобходимости**;
- В остальных случаях значение **Авто** трактуется как **Использовать**.

Свойство **Вариант масштаба** определяет, каким образом управляемая форма будет отображаться в интерфейсе **Такси**:

- **Авто** – система автоматически определяет режим отображения формы. В этом случае анализируется свойство **ВариантМасштабаФормКлиентскогоПриложения** объекта **НастройкиКлиентскогоПриложения**. Если в данном свойстве не указано конкретное значение варианта масштаба, то в интерфейсе **Такси** значение **Авто** будет трактоваться как **Обычный**.
- **Обычный** – форма выглядит стандартно для интерфейса **Такси**;

Рис. 179. Обычный вариант масштаба

- **Компактный** – в этом случае «почти» как в интерфейсе версии 8.2.

Рис. 180. Компактный вариант масштаба

С технической точки зрения, форма, для которой значение свойства **Вариант масштаба** установлено в значение **Компактный**, имеет следующие особенности:

- В качестве шрифта формы используется системный шрифт **Шрифт диалогов и меню**;
- Элементы формы имеют размеры, максимально соответствующие таковым в интерфейсе версии 8.2;
- Расстояния между элементами формы максимально соответствуют расстояниям в интерфейсе версии 8.2;
- Табличный документ, диаграмма, диаграмма Ганта, дендрограмма, графическая схема и географическая схема отображаются без увеличения масштаба.

Данное свойство предназначено для того, чтобы упростить использование сложных (насыщенных) форм в интерфейсе **Такси** без переработки этих форм. Данная возможность не является основной, т. к. получающийся в результате интерфейс не вполне соответствует интерфейсу **Такси**.

Также можно управлять масштабом сразу всех форм прикладного решения. Для этого следует соответствующим образом установить значение свойства

ВариантМасштабаФормКлиентскогоПриложения объекта НастройкиКлиентскогоПриложения. Значение этого свойства будет использоваться для всех форм, у которых значение свойства **Вариант масштаба** установлено в значение **Авто**.

Для формы можно отобразить командную панель (ее расположением на форме можно управлять с помощью свойства **Положение командной панели**). Состав стандартных команд формы можно регулировать через свойство формы **Состав команд**.

Если свойство **Разрешить изменять форму** снято, то пользователь не может изменить состав и относительное положение элементов формы в режиме 1С:Предприятие.

Свойство **Проверять заполнение автоматически** отвечает за проверку заполнения данных формы. Подробнее о проверке заполнения см. [здесь](#).

Если свойство формы **Сохранение данных формы в настройках** установлено в значение **Использовать список**, то в списке реквизитов становится доступной колонка **Сохранение**. Те реквизиты, для которых свойство **Сохранение** будет установлено в значение **Истина**, будут сохраняться в хранилище данных формы (подробнее см. [здесь](#)). Для сохранения будет использовано либо хранилище, указанное в свойстве формы **Хранилище настроек**, либо хранилище, указанное в свойстве **Хранилище настроек данных форм** конфигурации.

ПРИМЕЧАНИЕ. Недоступно указание флажка **Сохранение** для табличных частей объектов.

Для установки режима **Только просмотр** у формы существует свойство **ТолькоПросмотр**, доступное только для программного изменения. В случае установки этого свойства в состояние **Истина** следующие стандартные команды станут недоступны (в том числе и при попытке использовать для вызова этих команд сочетания клавиш):

- Для всех форм:
 - Восстановить параметры.
- Для табличного поля:
 - Добавить,
 - Скопировать,
 - Удалить.
- Команды **Создать на основании**.
- Для расширения формы объекта, записи регистра сведений, константы:
 - Записать,
 - Записать и закрыть,
 - Провести,
 - Провести и закрыть,
 - Отмена проведения,
 - Старт,
 - Выполнено.
- Для расширения формы выбора/настроек:
 - Завершить редактирование,
 - Выбрать настройки.
- Для расширения табличного поля для динамического списка:
 - Пометить на удаление,
 - Провести,
 - Отмена проведения,
 - Создать группу,
 - Переместить в группу.
- Для расширения табличного поля для списка значений:
 - Изменить,
 - Переместить вверх,
 - Переместить вниз,
 - Сортировать по убыванию,
 - Сортировать по возрастанию,
 - Подбор.
- Все команды изменения следующих коллекций системы компоновки данных:
 - коллекций настроек,
 - коллекций доступных полей.
- Для расширения табличного поля для **ДанныеФормыКоллекция**, **ДанныеФормыДерево**, **ДанныеФормыСтруктураИКоллекция**:
 - Изменить,
 - Переместить вверх,
 - Переместить вниз,
 - Сортировать по убыванию,
 - Сортировать по возрастанию.

Также будут недоступны элементы формы, связанные с этими командами.

Если для формы установлено свойство [Только просмотр](#) и основным реквизитом формы является динамический список, то все таблицы, связанные с этим списком, также перейдут в режим просмотра. Кроме того, если заполнено свойство [Список групп](#), то элемент формы, указанный в этом свойстве, также перейдет в режим просмотра. При этом свойство элементов формы [Только просмотр](#) не будет изменено.

7.6.3. Поле

Элемент формы [Поле](#) предназначен для отображения и редактирования какого-либо реквизита формы. Поле формы может быть нескольких видов:

- поле ввода,
- поле надписи,
- поле флажка,
- поле картинки,
- поле переключателя,
- поле текстового документа,
- поле табличного документа,
- поле календаря,
- поле индикатора,
- поле полосы регулирования,
- поле диаграммы,
- поле диаграммы Ганта,
- поле дендрограммы,
- поле графической схемы,
- поле географической схемы,
- поле HTML-документа,
- поле форматированного документа;
- поле планировщика.

Если поле является подчиненным элементом для элемента формы [Таблица](#), то оно может принимать следующие типы:

- поле ввода,
- поле картинки,
- поле надписи,
- поле флажка.

Если вставляется поле вида [Поле переключателя](#), то для него необходимо задать свойство элемента формы [Список выбора](#), который определяет количество и значения переключателей. По умолчанию используется горизонтальное расположение переключателей. Для того чтобы расположить их по вертикали, необходимо изменить значение свойства [Количество колонок](#).

Для полей вида [Поле флажка](#) и [Поле переключателя](#) существует возможность указывать вид отображения. Это делается с помощью свойства [Вид флажка](#) у соответствующего элемента формы. Для поля флажка можно выбрать представление в виде флажка или тумблера.



Полнотекстовый поиск:

Рис. 181. Флажок или тумблер

На [рис. 181](#) представлены различные варианты отображения поля флажка. В верхней части рисунка поле флажка представлено тумблером, а в нижней – флажком. Для поля переключателя можно выбрать представление в виде переключателя или тумблера.

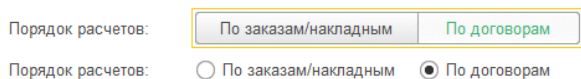


Рис. 182. Переключатель или тумблер

На [рис. 182](#) представлены различные варианты отображения поля переключателя. В верхней части рисунка поле переключателя представлено в виде тумблера, а в нижней – в виде переключателя.

Для того чтобы разместить на форме поле ввода с видами: поле табличного документа, поле диаграммы, поле диаграммы Ганта, поле дендрограммы, поле графической схемы, поле географической схемы, необходимо создать реквизит формы соответствующего типа и указать этот реквизит в качестве данных элемента формы.

Для того чтобы разместить на форме поле HTML-документа, необходимо создать реквизит формы типа [Строка](#), затем создать поле ввода вида [Поле HTML документа](#) и указать созданный реквизит в качестве данных для этого поля. В качестве значения реквизита формы может выступать навигационная ссылка или текст HTML-документа.

ПРИМЕЧАНИЕ. Следует учитывать, что при использовании поля HTML-документа может не обеспечиваться поддержка всей функциональности HTML. В частности, не поддерживаются скрипты с состоянием (переменные, хранящие значения, необходимые в различных обработчиках событий), свойство HTML-документа `parentWindow`, работа с языком разметки SVG. Ограничения могут быть обусловлены специфическими особенностями функционирования HTML (в том числе и неидентичное функционирование) в формах «1С:Предприятия» в клиентских приложениях и различных веб-браузерах.

Чтобы разместить на форме поле форматированного документа, необходимо создать реквизит формы типа [ФорматированныйДокумент](#), затем создать поле ввода вида [Поле форматированного документа](#) и указать созданный реквизит в качестве данных для этого поля.

ПРИМЕЧАНИЕ. Работа с реквизитом типа [ГеографическаяСхема](#) возможна только на сервере.

Если в поле ввода формы редактируется реквизит, который заполняется автоматически, но, тем не менее, может заполняться вручную (в крайне редких случаях), то такое поведение можно реализовать с помощью свойств поля ввода **Отображение предупреждения при редактировании** и **Предупреждение при редактировании**. Если значение свойства **Отображение предупреждения при редактировании** равно **Отображать**, то при попытке начать редактирование поля «1С:Предприятие» выдаст предупреждение, которое состоит либо из строки, указанной в свойстве **Предупреждение при редактировании**, либо формируется автоматически.

Для стандартных полей **Код** и **Номер** (для объектов с автоматической нумерацией) автоматическая строка предупреждения будет выглядеть как **Номер заполняется при записи автоматически**, а для остальных полей автоматическая строка будет выглядеть как **Редактирование поля "Имя поля" не рекомендуется**.

Если значение свойства **Отображение предупреждения при редактировании** равно **Авто**, то для стандартных реквизитов **Код** и **Номер** будет использоваться значение **Отображать**, а для остальных полей – **Не отображать**.

Свойство **Отображение предупреждения при редактировании** также оказывает влияние на свойство поля ввода **Пропускать при вводе**. Если значение свойства **Пропускать при вводе** равно **Авто**, то поле будет пропускаться при вводе в том случае, если значение свойства **Отображение предупреждения при редактировании** равно **Отображать** (или **Авто** для стандартных полей **Код** и **Номер**).

Если свойство **АвтоОтметкаНезаполненного** поля ввода установлено в значение **Истина**, то свойство этого поля **ОтметкаНезаполненного** будет автоматически актуализироваться при передаче значения из реквизита в поле ввода.

Для реквизита формы типа **СтандартныйПериод** или **СтандартнаяДатаНачала** существуют два способа редактирования:

- С использованием одного поля формы. В этом случае можно разместить на форме поле, связанное непосредственно с редактируемым реквизитом. Тогда все редактирование будет выполняться именно в этом поле.
- С использованием нескольких полей. При этом можно разместить на форме поля, связанные со свойствами реквизита. В этом случае разработчик может сам реализовывать необходимую логику взаимодействия варианта стандартного периода и значений дат.

Свойство поля ввода **Подсказка ввода** позволяет указывать подсказку, которая будет формироваться непосредственно «внутри» поля ввода, если выполняются следующие условия:

- Тип связанного реквизита не является типом **Число** или **Дата**.
- Значение связанного реквизита не содержит значение по умолчанию для данного типа.

7.6.4. Декорация

Представляет собой оформительский элемент формы. Декорация не связана с данными (реквизитами формы). Декорация может представлять собой надпись или картинку.

СОВЕТ. Рекомендуется использовать декорацию только для неизменяемых текстовых пояснений. Текст, который может изменяться программно, рекомендуется отображать полем вида поле надписи.

7.6.5. Таблица

Элемент формы **Таблица** предназначен для визуализации табличных данных. Это может быть динамический список, табличная часть, список значений и т. д.

Положение командной панели таблицы регулируется свойством **Положение командной панели** элемента формы. Состав стандартных команд, располагающихся в командной панели таблицы, регулируется с помощью свойства **Состав команд**. Команды, отключенные при редактировании данного свойства, становятся недоступными.

Имеется возможность группировать колонки с помощью элемента формы **Группа**. Подробнее об этом см. [здесь](#).

Для реквизита формы типа **СписокЗначений** существует выбор, каким образом реквизит будет отображаться. Если выбирается вариант **Таблица**, то список значений будет выглядеть как таблица с доступными колонками: **Значение**, **Представление**, **Пометка** и **Картинка**. При этом следует помнить об особенности отображения значений в колонке **Значение**. В этой колонке будет отображено значение колонки **Представление**. Само значение при этом остается неизменным. Если будет выбран тип **Поле ввода**, то для редактирования списка будет открыто специальное окно.

Также для таблицы, связанной с реквизитом типа **СписокЗначений**, предоставляется стандартный набор команд редактирования, включая команды установки и снятия пометок.

Если таблица связана с реквизитом типа **ДинамическийСписок**, у которого в настройках списка (свойство реквизита формы **Настройка списка**) задана группировка, то список всегда будет отображаться в режиме **Дерево** (свойство элемента формы **Отображение**), вне зависимости от того, какой режим отображения задан разработчиком. Если для табличного поля, связанного с динамическим списком, установлено свойство **Только просмотр**, то при открытии из такого списка форм объектов им будет автоматически установлен параметр формы **ТолькоПросмотр**.

Также для таблицы, связанной с динамическим списком, имеется возможность управлять обновлением списка при изменении данных с помощью свойства **Обновление при изменении данных**. Если свойство установлено в значение **Авто**, то список будет автоматически обновляться при любых изменениях отображаемых данных. Если свойство имеет значение **Не обновлять**, то список не будет выполнять автоматическое обновление и для обновления списка необходимо явно выполнить команду **Обновить ()**. Если для динамического списка задано автоматическое обновление с указанным интервалом и свойство **Обновление при изменении данных** установлено в значение **Авто**, интервал отсчитывается от ближайшего изменения данных, а если в значении **Не обновлять** – от последнего вызова автоматического обновления или выполнения команды **Обновить ()**.

Для таблицы, связанной с основным реквизитом формы типа динамический список, имеется возможность управлять режимом множественного выбора с помощью соответствующего параметра формы. Значение параметра формы **МножественныйВыбор** будет установлено в одноименное свойство таблицы. При этом свойство **Режим выделения** будет установлено в значение **Множественный**, если параметр формы **МножественныйВыбор** имеет значение **Истина**.

При открытии формы выполняется чтение данных только для видимых таблиц, связанных с динамическими списками. Для таблиц, которые не видны при открытии формы, чтение данных выполняется в момент их (таблиц) отображения пользователю. В частности, для невидимых таблиц, значение свойства **ТекущаяСтрока** будет равно **Неопределено**. Эту особенность следует учитывать при разработке логики формы.

Для динамического списка, находящегося в режиме выбора, при добавлении нового объекта будет выполнена проверка на соответствие добавленного объекта установленным критериям отбора, и если новый объект удовлетворяет критериям отбора, на него будет установлен курсор в таблице, отображающей динамический список.

Если форма списка плана счетов или справочника с иерархией элементов открывается с установленными параметрами **РазрешитьВыборКорня**, равным **Истина**, и **РежимВыбора**, равным **Истина**, то для всех таблиц формы, связанных с основным реквизитом формы (типа **ДинамическийСписок**), будут автоматически изменены значения следующих свойств:

- свойство **ОтображатьКорень** будет установлено в значение **Истина**;
- свойство **РазрешитьВыборКорня** будет установлено в значение **Истина**;
- свойство **Отображение** будет установлено в значение **Дерево**.

Если таблица отображает иерархический список, для которого установлена картинка строк, то для строк таблицы, отображающих группы, установленная картинка будет игнорироваться и всегда будет выводиться стандартная картинка группы.

ПРИМЕЧАНИЕ. Если таблица отображает динамический список (см. [здесь](#)) и свойство **Начальное отображение дерева** установлено в значение **Раскрывать все уровни**, то при создании формы на сервере будет получен только список строк, отображаемых на первом уровне, а затем, при открытии формы, будут получены остальные отображаемые строки. Причем для каждой отображаемой группы будет выполнен отдельный запрос к серверу.

При разработке форм возникают ситуации, когда необходимо в таблице предотвратить прокрутку некоторых колонок. Например, в табличной части нужно зафиксировать колонки **Номер**, **Номенклатура** и **Сумма**. При этом колонки **Номер** и **Номенклатура** требуются прижать к левому краю таблицы, а **Сумма** – к правому.

Для выполнения такой операции предназначено свойство **ФиксацияВТаблице**. Данное свойство доступно у поля, размещенного в таблице, и у группы вида **Группа колонок** (см. [здесь](#)). В нашем примере необходимо для колонок **Номер** и **Номенклатура** установить свойство **ФиксацияВТаблице** в значение **Лево**, а для колонки **Сумма** – в значение **Право**. Колонки будут прижаты к соответствующему краю таблицы вне зависимости от того, где эти колонки были расположены в редакторе формы, однако зафиксированные колонки располагаются в том порядке, в котором они расположены в редакторе формы.

Если признак фиксации указан у группы, то прижата будет вся группа (вместе со всеми подчиненными элементами формы), при этом значение свойства **ФиксацияВТаблице** для подчиненных элементов игнорируется. Если признак фиксации указан у колонки, входящей в группу, для которой свойство **ФиксацияВТаблице** установлено в значение **Нет**, то колонка будет обрабатываться как отдельная колонка, не входящая в группу.

Свойство **Поиск при вводе символов** отвечает за то, каким образом система будет обрабатывать набор символов с клавиатуры, если активным элементом является таблица и в колонку невозможен непосредственный ввод данных:

- **Авто** – в этом случае окно поиска открывается для таблицы, связанной с динамическим списком, для остальных таблиц окно поиска не открывается.
- **Использовать** – окно поиска открывается для любых типов реквизитов, связанных с таблицей, если у колонки, в которой осуществляется ввод символов, свойство **Режим редактирования** установлено в значение **Вход**.
- **Не использовать** – окно поиска не открывается для любых типов реквизитов, связанных с таблицей.

7.6.6. Кнопка

На форме команда отображается элементом формы **Кнопка**. Кнопка может отображаться в командной панели и просто на форме. В случае своего отображения на форме кнопка может также иметь вид гиперссылки (для этого нужно изменить свойство **Вид**).

Команда, к которой будет происходить обращение по нажатию кнопки, устанавливается с помощью свойства **Команда**.

Установленное свойство **Только во "Все действия"** означает, что по умолчанию кнопка будет находиться в меню **Еще (Все действия)**, однако пользователь может сделать кнопку доступной на панели, воспользовавшись специальным редактором.

Для кнопки, связанной с командой **Создать по параметру**, имеется возможность указать тип документа, который будет создаваться при нажатии на кнопку. Это можно сделать с помощью свойства кнопки **Параметр**.

Если для команды формы не задана свойство **Подсказка**, то для кнопки, связанной с этой командой, не будет отображаться всплывающая подсказка. Если заголовок кнопки не может быть целиком отображен на кнопке, то полный текст заголовка отображается во всплывающей подсказке первой (или единственной, если для кнопки не задана подсказка) строкой.

7.6.7. Группа

При разработке формы имеется возможность различного объединения элементов. Можно объединять поля, страницы формы, команды, колонки.

На форме могут находиться группы элементов управления. Это могут быть группы полей, страницы, группы команд. Также для элементов типа **Таблица** можно создавать группы колонок.

В редакторе можно создать группы следующих видов:

- **Обычная группа** – предназначена для объединения элементов формы. Такая группа может обладать различными вариантами оформления:
 - **Нет** – группа специально никак не выделяется;
 - **Слабое выделение** – заголовок группы отображается большим шрифтом зеленого цвета;
 - **Обычное выделение** – заголовок группы отображается большим шрифтом зеленого цвета. Вокруг элементов группы (с каждой стороны) формируются отступы;
 - **Сильное выделение** – заголовок группы отображается большим шрифтом зеленого цвета. С левой стороны (на всю высоту группы) формируется полоса зеленого цвета. Снизу формируется отступ.

Обычную группу можно настроить таким образом, что пользователь (в режиме «1С:Предприятия») сможет сворачивать или разворачивать такую группу. Например, можно разместить в сворачиваемой группе информацию, которая не важна при регулярной работе с формой, но иногда может потребоваться для просмотра и анализа.

Поведение группы (обычная или сворачиваемая) определяется свойством **Поведение**. Начальное состояние группы определяется флажком **Свернута**. Группа не может быть свернута или развернута, если у нее не отображается заголовок или заголовок не заполнен. Прикладной разработчик не может определить текущее состояние группы из встроеного языка. Также он не может принудительно свернуть или развернуть группу.

Управлять состоянием группы (свернутая или развернутая) можно с помощью специальной картинкой или гиперссылки. Для задания элемента, управляющего состоянием, служит свойство **Отображение управления**. Также можно задать заголовок группы (свойство **Свернутый заголовок**), который будет отображаться в том случае, если группа находится в свернутом состоянии. Если свойство **Свернутый заголовок** не заполнено, то в свернутом состоянии будет отображаться обычный заголовок группы.

- **Командная панель** – элемент формы, предназначенный для размещения кнопок и групп. Свойство командной панели **Источник команд** определяет элемент формы (**Форма** или элементы формы типа **Таблица**), который предоставит «свои» команды для отображения в командной панели. Состав команд для отображения в командной панели регулируется свойством **Состав** команд элемента формы, являющегося источником команд.

В том случае, когда в командной панели у всех кнопок свойство **ТолькоВоВсехДействиях** имеет значение **Ложь** и не все кнопки могут быть размещены на командной панели (по ширине), в командной панели автоматически формируется кнопка **Еще** (с подменю). В это подменю размещаются все команды командной панели.

- Группа, добавляемая в командную панель, может быть одного из следующих видов:
 - **Подменю** – группа, которая позволяет делать выпадающие меню.
 - **Группа кнопок** – позволяет создать группу кнопок, которые будут обладать следующими свойствами:
 - группа кнопок отделяется разделителями слева и справа;
 - каждая группа кнопок может обладать собственным источником команд.
 - **Страницы** – данная группа предназначена для организации панели с закладками. Закладки могут располагаться как по вертикали, так и по горизонтали. Кнопки прокрутки появляются в том случае, если закладки не помещаются в отведенное им место на форме (по ширине или высоте группы **Страницы**). Для того чтобы добавить на такую панель страницы, необходимо добавить столько вложенных групп, сколько страниц должно быть у панели. Внутри группы вида **Страницы** могут располагаться только группы вида **Страница**:
 - **Страница** – группа специального вида, которая предназначена для формирования страниц панели. Данная группа может содержать другие вложенные элементы.
 - **Группа колонок** – позволяет объединять колонки в таблице. С помощью группы этого вида можно изменять правило группировки колонок (вертикальная или горизонтальная группировка).

Группы могут быть вложенными.

Элементы формы можно перемещать между группами. При этом автоматически определяется допустимость такого переноса. В случае, если перенос требует изменения каких-либо свойств элемента (например, Вид), изменение осуществляется автоматически. Если в результате изменились требования к подчиненным элементам, они либо автоматически изменяются, либо удаляются.

Свойство **Только просмотр** группы влияет на все элементы, подчиненные группе.

Свойство группы **Разрешить изменение состава** управляет возможностью изменять состав группы в процессе настройки формы пользователем. Если свойство выключено, то пользователь не может изменять состав и менять порядок элементов внутри группы. Однако свойство не влияет на возможность пользователя изменить видимость элементов, входящих в состав группы.

Если у группы указано свойство **Данные заголовка**, то эти данные будут автоматически отображаться в заголовке группы. Если у группы установлено свойство **Заголовок**, то данные **заголовка** отображаются в скобках после заголовка: **Заголовок (Данные заголовка)**.

Любая группа кнопок (командная панель, подменю, группа кнопок, контекстное меню) заполняется по одинаковым правилам. Кнопки будут отображены в следующем порядке:

1. Кнопки, предоставляемые источником команд, с которым связана данная группа.
2. Кнопки командного интерфейса, если источник команд подразумевает его включение.
3. Пользовательские команды, добавленные в данную группу.
4. Команды меню **Еще (Все действия)** (только для командной панели).
5. Последней будет размещена **Справка** (в том случае если она была предоставлена источником команд).

Если вышеприведенный автоматический порядок расположения кнопок в группе не устраивает, то можно воспользоваться следующими возможностями:

- Снять флажок **Автозаполнение** для командной панели формы, контекстного меню или командной панели элемента и добавить нужные команды (в нужном порядке) вручную.
- Отключить источник команд командной панели, затем добавить нужные команды (в нужном порядке) вручную, затем добавить группу типа **Группа кнопок** и установить для добавленной группы источник команд.
- Можно вручную поместить кнопку, связанную со стандартной командой источника команд, в необходимое место командной панели. В этом случае соответствующая команда не будет автоматически размещена в командной панели.

7.6.8. Дополнение элемента формы

Дополнение элемента формы предназначено для отображения дополнительных свойств элемента формы (дополнения элемента формы) и управления этими элементами. Существуют следующие дополнения:

- Отображение строки поиска. Позволяет настроить отображение строки поиска (см. [здесь](#)).
- Состояние просмотра. Позволяет настроить отображение перечня примененных поисковых запросов (см. [здесь](#)). Не может располагаться в командной панели.
- Управление поиском. Позволяет настроить внешний вид кнопки управления поиском (см. [здесь](#)).

7.6.9. Специальные командные панели

Кроме создания собственных командных панелей (элемент формы **Группа** вида **Командная панель**) редактор форм предоставляет возможность работать со специализированными командными панелями:

- **Командная панель формы** – система предоставляет командную панель формы. Ее видимостью, положением и наполнением можно управлять. Видимость и положение панели управляются с помощью свойства формы **Положение командной панели**, а содержимое управляется свойством **Автозаполнение** специальной группы **Командная панель**. Кроме стандартных команд, которые могут быть добавлены на командную панель формы, туда автоматически добавляются команды из раздела **Командная панель формы** глобального командного интерфейса.

Если основным реквизитом формы является динамический список и у командной панели формы установлено свойство **Автозаполнение**, то в нее автоматически будут добавлены команды, предоставляемые этим динамическим списком. Если на форме также присутствует элемент формы, отображающий дерево, то командная панель формы будет дополнена командами, которые предоставлены реквизитом, связанным с таким элементом.

- **Командная панель таблицы** – системой автоматически предоставляется командная панель для элемента формы вида **Таблица**. Ее видимостью, положением и наполнением можно управлять. Видимость и положение панели управляются с помощью свойства формы **Положение командной панели**, а содержимое управляется свойством **Автозаполнение** специальной группы **Командная панель**.
- **Контекстное меню элемента** – имеется возможность модифицировать стандартное контекстное меню элемента формы. Для этого контекстное меню необходимо отобразить в дереве элементов формы с помощью команды контекстного меню **Показать контекстное меню** элемента формы.

Если в контекстное меню элемента добавить элемент вида **Группа** и указать у этой группы в качестве источника команд другой элемент формы, то добавленная группа будет заполнена командами (в режиме 1С:Предприятие) по тем же правилам, что и контекстное меню элемента, выбранного в качестве источника команд.

Во все вышеперечисленные командные панели можно добавлять собственные команды, которые создаются с помощью редактора команд.

Если есть необходимость поместить команды, расположенные в командной панели формы (см. [здесь](#)), в какую-либо командную панель, то необходимо выполнить следующее:

- Разместить в командной панели элемент **Группа – Командная панель**.
- Выбрать эту группу в качестве источника команд **Глобальные команды командной панели формы**.

Если в командной панели или контекстном меню, содержащем группу с источником команд **Глобальные команды командной панели формы**, есть группа, для которой установлен источник команд **Форма**, то при заполнении этой группы команды, относящиеся к командному интерфейсу формы, не добавляются.

Если в процессе контекстного серверного вызова в форме выполнены следующие изменения: изменена структура данных; добавлены или удалены команды формы; добавлены, удалены или перемещены элементы формы; изменены параметры таблиц управляемой формы, то будет выполнено перезаполнение стандартных командных панелей и групп. При этом все пользовательские команды, вставленные в такие панели и группы, будут удалены, а измененные параметры стандартных команд будут установлены в значения по умолчанию.

7.6.10. Особенности поведения элементов формы

В режиме **ТолькоПросмотр**, который установлен как для конкретного элемента, так и для формы в целом, поля формы вида надпись, картинка, декорация-надпись, декорация-картинка отображаются незапрещенными.

Если для поля вида надпись или картинка свойства **Гиперссылка** установлено в значение **Истина**, то при выборе гиперссылки (с помощью клавиши **Enter** или двойного щелчка мыши) будет срабатывать обработчик события элемента формы **Нажатие**. Разработчику необходимо самому выполнять обработку свойства **ТолькоПросмотр** у формы или соответствующего элемента и выполнять соответствующие действия (например, по блокировке данных для редактирования).

Если у реквизита формы типа **Число** свойства **Длина** и **Точность** заданы одинаковыми значениями, то поле ввода позволяет ввести любое число, но при покидании поля ввода будет выдано сообщение о некорректном вводе, если введенное значение не находится в диапазоне от **-1** до **1** (исключая **-1** и **1**).

Элемент формы **Кнопка**, связанный с командой, у которой свойство **Изменяет сохраняемые данные** имеет значение **Истина**, будет отображаться запрещенным, если для него (или формы в целом) установлено свойство **ТолькоПросмотр** равным значению **Истина**.

Если в обработчике **ПриИзменении** поля, расположенного в таблице формы, выполняются действия, которые ведут к вызову сервера и перестроению формы, то обработчики событий этой таблицы **ПриОкончанииРедактирования** и **ПередОкончаниемРедактирования** не будут вызваны.

Табличный документ, расположенный на форме, по вертикали прокручивается построчно.

Если для поля ввода или поля надписи свойство **Высота** установлено в значение большим 1 и свойство **Растягивать по вертикали** установлено в значение **Авто**, то это значение будет трактоваться как **Истина**, т. е. указанные элементы формы будут растягиваться по вертикали.

7.6.11. Правила размещения элементов формы

В общем случае можно определить правила размещения элементов формы следующим образом:

- Элементы формы размещаются в горизонтальном (каждый новый элемент формы располагается правее предыдущего) или вертикальном (каждый новый элемент формы располагается ниже предыдущего) направлении (в зависимости от свойства формы **Группировка**), без ограничения по количеству отображаемых элементов.
- Порядок отображения элементов определяется порядком их расположения на закладке **Элементы** редактора формы.
- Размер формы (и элементов формы) может быть ограничен свойствами **Ширина** и **Высота** (или **Высота в строках таблицы**).
- В случае, если при размещении элементов заданные размеры формы будут превышены, в форме появятся линейки прокрутки.
- Если отображаемый элемент является группой, то правила размещения элементов формы внутри группы эквивалентны правилам размещения внутри формы.

Несмотря на простоту алгоритма, у разработчика есть несколько способов существенно вмешаться в этот алгоритм и повысить удобочитаемость формы:

- группы элементов формы,
- размеры элементов,
- ширина подчиненных элементов.

Далее мы рассмотрим подробнее каждую из перечисленных возможностей.

ВНИМАНИЕ! При размещении элементов одновременно работают и основные правила размещения, и дополнительные правила, которые рассмотрены ниже.

7.6.11.1. Группы и объединение элементов формы

Для объединения элементов формы доступны несколько видов элемента формы **Группа**:

- **Командная панель** – не позволяет изменять правила размещения, установленные для родительского элемента формы (другой группы или формы).
- **Страница** – позволяет изменять правила размещения элементов.
- **Обычная группа** – позволяет изменять правила размещения элементов.
- **Группа колонок** – позволяет изменять правила размещения элементов (колонок таблицы формы).

Для группы вида **Обычная группа** или **Страница** можно задавать группировку (свойство **Группировка**) элементов и ширину подчиненных элементов (свойство **Ширина подчиненных элементов**). Опишем пример использования такой группировки.

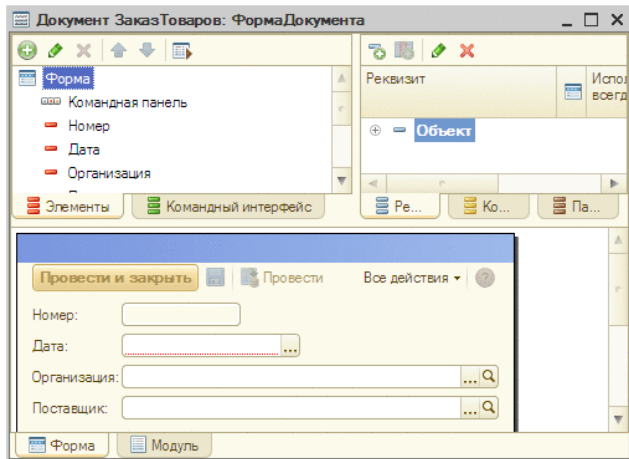


Рис. 183. Вертикальная группировка

Рассмотрим создание формы документа. При этом группировка элементов формы выбрана как **Вертикальная**, но мы хотим, чтобы реквизиты документа **Дата** и **Номер** располагались не в двух строках, а в одной.

Для этого мы создадим группу с именем **ДатаИНомер**, установим свойство **Группировка** в значение **Горизонтальная** и поместим туда элементы формы **Номер** и **Дата**.

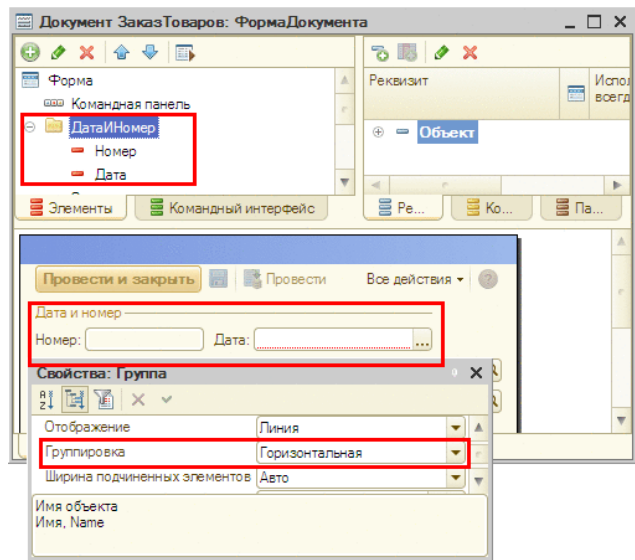


Рис. 184. Горизонтальная группировка

ПРИМЕЧАНИЕ. Следует отметить, что группы могут иметь неограниченную вложенность. При этом свойство [Группировка](#) может быть разным для каждой вложенной группы.

7.6.11.2. Размеры элементов формы

Для каждого элемента формы можно задать размеры – высоту и ширину. Размеры задаются в абстрактных единицах измерения. Реальный размер элемента формы в режиме 1С:Предприятие может не совпадать с указанными размерами. При определении размеров элементов система учитывает шрифт (и его размеры), указанный в свойствах элементов формы.

ПРИМЕЧАНИЕ. Изменение шрифта элемента формы, выполненное с помощью условного оформления, не учитывается при определении размера элементов.

Если размеры заданы равными 0, то платформа будет вычислять размеры автоматически, стремясь к наилучшему отображению формы на экране.

Если размеры заданы отличными от 0, то действуют следующие правила:

- размер, заданный для родительского элемента, ограничивает размеры подчиненных элементов;
- размер, заданный для подчиненного элемента, изменяет размер родительского элемента до тех пор, пока не нарушается предыдущее условие.

Если размер родительского элемента задан таким образом, что подчиненные элементы не помещаются в отведенное место, то действуют следующие правила:

- Если они не помещаются по высоте, добавляется вертикальная полоса прокрутки.
- Если они не помещаются по ширине, уменьшается их размер так, чтобы они отображались без горизонтальной прокрутки. Тем не менее, горизонтальная прокрутка появится в форме, если уменьшить ее ширину так, что элементы больше не смогут уменьшаться по ширине.

7.6.11.3. Ширина подчиненных элементов

При проектировании форм возникают ситуации, когда нужно расположить элементы формы в две колонки, шириной колонок подчеркнув их значимость.

Для решения первой задачи (размещение в две колонки) необходимо использовать группы, а для решения второй задачи – использовать свойство [Ширина подчиненных элементов](#) (для групп, имеющих горизонтальную группировку элементов).

ВНИМАНИЕ! Следует помнить, что свойство [Ширина подчиненных элементов](#) оказывает влияние на размещение элементов формы только тогда, когда у родительской группы установлена горизонтальная группировка и существуют только две подчиненные группы.

Это свойство может иметь следующие значения:

- **Авто** – ширина элементов подбирается системой автоматически.
- **Одинаковая** – выбирается одинаковая ширина элементов.
- **Левый широкий** – ширина левого элемента относится к ширине правого элемента как 3:2.
- **Левый очень широкий** – ширина левого элемента относится к ширине правого элемента как 2:1.
- **Левый узкий** – ширина левого элемента относится к ширине правого элемента как 2:3.
- **Левый очень узкий** – ширина левого элемента относится к ширине правого элемента как 1:2.

Рассмотрим это на примере.

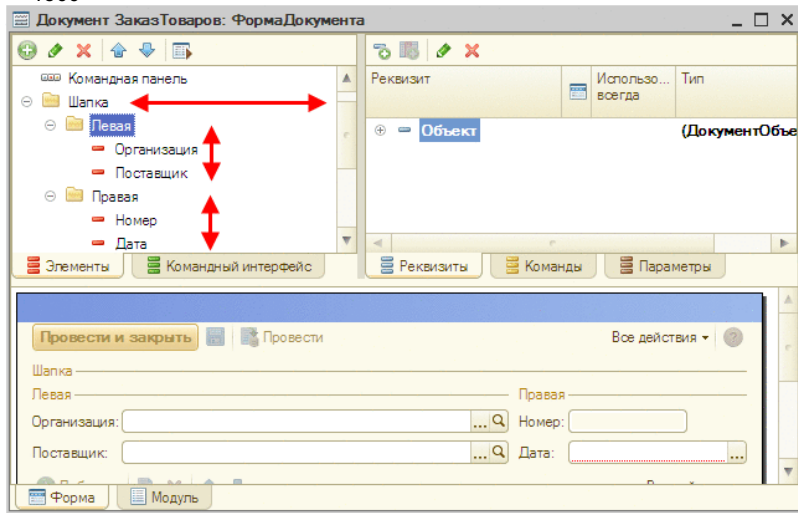


Рис. 185. Ширина подчиненных элементов

Допустим, нам необходимо, чтобы в шапке документа было две колонки. В одной (левой) должны быть реквизиты **Покупатель** и **Склад**, а в другой (правой) – реквизиты **Вид цен** и **Валюта взаиморасчетов**. При этом мы считаем, что левая колонка является более значимой, и мы хотим увеличить ее ширину.

Для этого создаем группу **Шапка** и задаем в ней тип группировки **Горизонтальная**. В этой группе создаем еще две группы: **Левая** и **Правая**. Каждая группа имеет тип группировки **Вертикальная**. В группу **Левая** поместим реквизиты **Покупатель** и **Склад**, а в группу **Правая** – реквизиты **ВидЦен** и **Валюта**.

Затем у группы **Шапка** установим свойство **Ширина подчиненных элементов** в значение **Левый очень широкий**.

ПРИМЕЧАНИЕ. Основное предназначение свойства **Ширина подчиненных элементов** состоит в том, чтобы указать платформе, какие пропорции следует соблюдать при формировании формы.

7.6.12. Выравнивание границ элементов формы

При размещении элементов в форме система «1С:Предприятие» выполняет автоматическое выравнивание левой границы областей данных элементов формы в зависимости от типа элемента формы и некоторых правил. В данном разделе описываются правила выравнивания границ элементов формы.

Элемент формы состоит из **области данных** – области элемента формы, в которой отображаются данные, и **области заголовка (заголовок)** – области элемента формы, которая отображает заголовок (см. [рис. 186](#)). **Стандартная ширина области данных** определяет ширину области данных и обладает следующими свойствами:

- определяется типом данных реквизита, связанных с элементом формы или свойством **Ширина** элемента формы;
- не зависит от свойства элемента формы **Растягивать по горизонтали**.

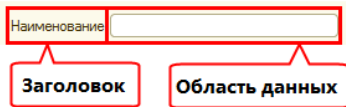


Рис. 186. Структура элемента формы

7.6.12.1. Выравнивание левых границ областей данных

7.6.12.1.1. Общие правила

Если в форму или группу вида **Обычная группа** или **Страница** с вертикальной группировкой (свойство **Группировка** установлено в значение **Вертикальная**) поместить несколько элементов, имеющих заголовки, отображаемые слева от области данных, то будет выполнено выравнивание левых границ областей данных элементов формы.

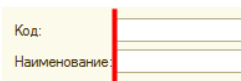


Рис. 187. Выравнивание левой границы

Однако для выполнения такого выравнивания необходимо соблюдение следующих правил:

- Расширение заголовка не приведет к тому, что ширина области данных станет меньше стандартной ширины данных:

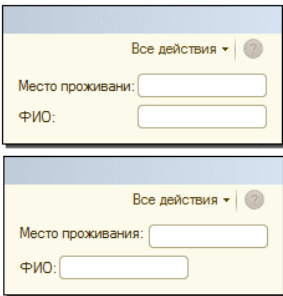


Рис. 188. Ширина области данных стала меньше стандартной ширины области данных

В примере, показанном на [рис. 188](#), выравнивание перестает работать после того, как строка заголовка первого элемента расширится. При этом ширина области данных поля **ФИО** станет меньше, чем стандартная ширина области данных этого поля, в том случае, если бы выравнивание произошло. Это соотношение изменяет добавление к заголовку одной буквы «я».

- Ширина заголовка не станет существенно превышать максимальное значение из стандартной ширины области данных поля и ширины заголовка этого поля.

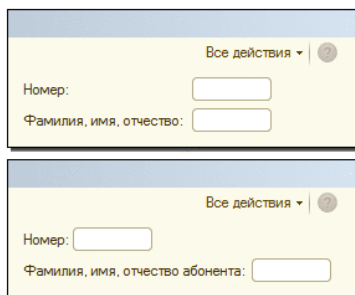


Рис. 189. Заголовок больше стандартной ширины области данных

В примере, показанном на рис. 189, выравнивание перестает работать после того, как строка заголовка второго элемента расширится так, что ширина заголовка поля «**Фамилия, имя, отчество...**» станет существенно больше, чем стандартная ширина области данных поля. Это соотношение изменяет добавление к заголовку слова «**абонента**».

Если для нескольких элементов, расположенных вертикально, невозможно определить единую ширину заголовка по вышеописанным правилам, то система будет определять наборы элементов, внутри которых условия выполняются. Объединение в наборы не зависит от порядка следования элементов по вертикали. Объединение элементов начинается с элемента, заголовок которого имеет минимальную ширину. При этом элементы, расположенные в одном наборе, не обязательно будут следовать подряд.

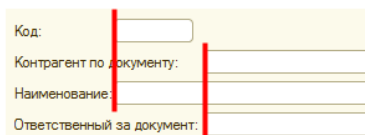


Рис. 190. Выравнивание наборов элементов

При выравнивании следует учитывать некоторые особенности:

- Если у элемента формы нет заголовка или заголовок расположен не слева, то такой элемент не выравнивается.
- Таблицы формы не выравниваются, даже если у нее имеется заголовок, расположенный слева.

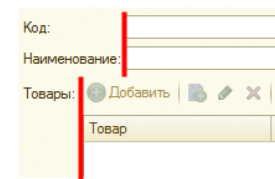


Рис. 191. Таблицы не выравниваются

- Если между элементами есть элемент, для которого выравнивание неприменимо, то он не влияет на выравнивание.

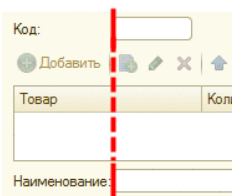


Рис. 192. Выравнивание неприменимо к элементу формы

- Для флажка отображаемыми данными считается прямоугольная область, в которой отображается сам флажок.

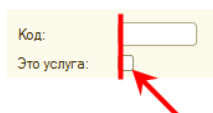


Рис. 193. Выравнивание флажка

7.6.12.1.2. Вложенные группы

Если элементы формы (с отображаемыми заголовками слева) расположены в группе типа **Обычная группа**, то выравнивание элементов будет происходить вместе с обрамляющими элементами только в том случае, если свойство **Отображение** у группы установлено в значение **Нет**. В противном случае элементы, помещенные в группу, при выравнивании не будут учитывать элементы, расположенные вокруг группы.



Рис. 194. Выравнивание с учетом групп элементов

7.6.12.2. Выравнивание правых границ элементов формы

Выравнивание правых границ элементов формы зависит от свойства элемента формы [Растягивать по горизонтали](#). Если свойство [Растягивать по горизонтали](#) установлено в значение [Авто](#), то выравниванием правой границы элементов управляет система.

Если свойство установлено в значение [Нет](#), то данный элемент не изменяет размер по горизонтали, а если у него есть вложенные элементы, препятствует изменению горизонтального размера этих элементов.

Если свойство установлено в значение [Да](#), то:

1. Правая граница элемента формы изменяет свое положение до тех пор, пока не «упрется» в другой элемент (или границу формы), препятствующий дальнейшему изменению положения.

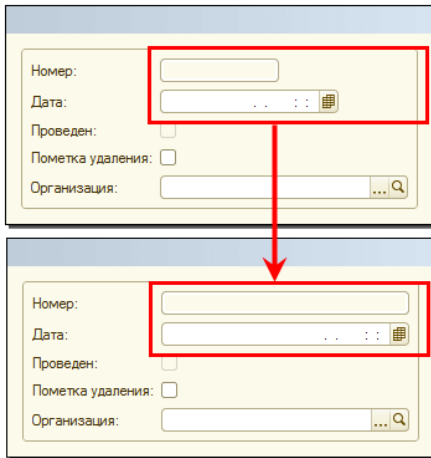


Рис. 195. Расширение элемента вправо до ограничителя

2. Вложенные элементы формы, у которых свойство [Растягивать по горизонтали](#) установлено в значение [Да](#) (или [Авто](#) – для некоторых элементов), также изменяют положение своей правой границы при изменении размера родительского элемента формы.

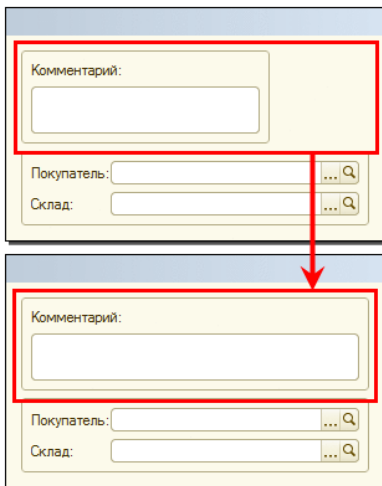


Рис. 196. Расширение подчиненных элементов

3. Если правее элемента формы есть другие элементы формы, которые могут быть перемещены вправо, такое перемещение выполняется до тех пор, пока для самого правого элемента формы не сработает ограничение из п. 1 (элемент формы «толкает» своих соседей справа).

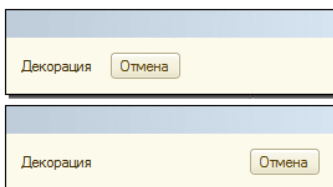


Рис. 197. Перемещение элемента справа

4. Если свойство [Растягивать по горизонтали](#) установлено в значение [Да](#) для двух (и более) рядом расположенных элементов, то при изменении горизонтального размера родительского элемента такие элементы изменяют свою ширину одновременно и пропорционально.

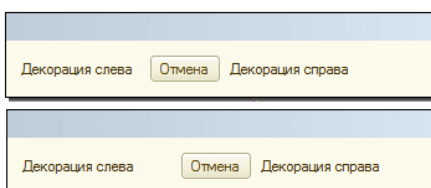


Рис. 198. Пропорциональное изменение

В примере на [рис. 198](#) кнопка [Отмена](#), расположенная между декорациями, не изменяет свой размер, т. к. у нее нет свойства [Растягивать по горизонтали](#).

7.6.12.3. Выравнивание нижних границ элементов формы

Выравнивание нижних границ элементов формы зависит от свойства элемента формы **Растягивать по вертикали**. Если свойство **Растягивать по вертикали** установлено в значение **Авто**, то выравниванием нижних границ элементов управляет система.

Если свойство установлено в значение **Нет**, то данный элемент не изменяет размер по вертикали, а если у него есть вложенные элементы, препятствует изменению вертикального размера этих элементов.

Если свойство установлено в значение **Да**, то:

1. Нижняя граница элемента формы изменяет свое положение до тех пор, пока не «упрется» в другой элемент (или границу формы), препятствующий дальнейшему изменению положения.

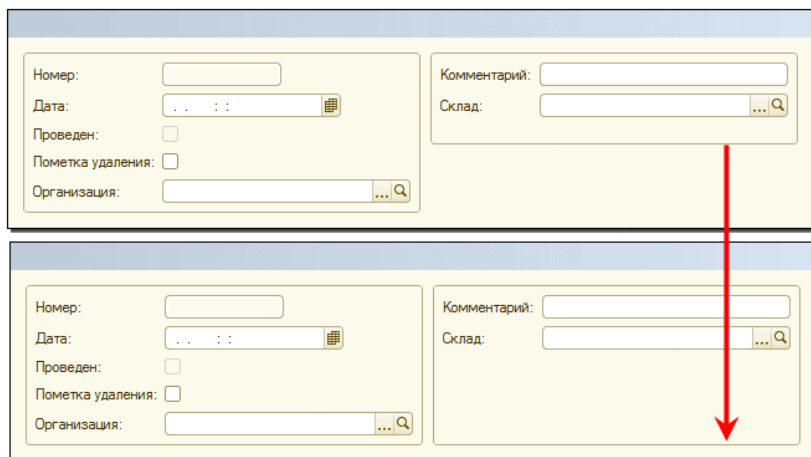


Рис. 199. Расширение элемента

2. Вложенные элементы формы, у которых свойство **Растягивать по вертикали** установлено в значение **Да** (или **Авто** – для некоторых элементов), также изменяют положение своей нижней границы при изменении размера родительского элемента формы.

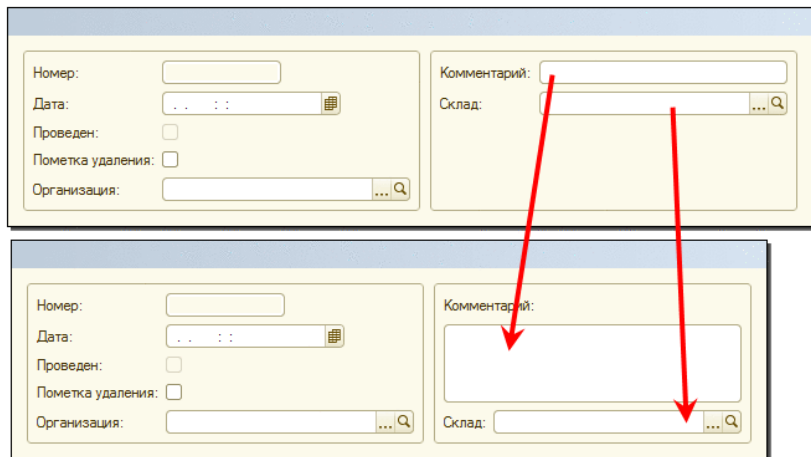


Рис. 200. Изменение вложенного элемента

3. Если ниже элемента формы есть другие элементы формы, которые могут быть перемещены вниз, такое перемещение выполняется до тех пор, пока для самого нижнего элемента формы не срабатывает ограничение из п. 1 (элемент формы «толкает» своих соседей снизу). Эта ситуация изображена на [рис. 200](#) (элемент **Склад**).

7.6.13. Механизм перетаскивания

В «1С:Предприятии» поддерживается операция перетаскивания. С ее помощью можно осуществлять перенос данных между разными элементами формы. Например, можно перенести элементы списка справочника из одной группы в другую, перенести данные из табличного поля в поле табличного документа или перенести список выделенных файлов из проводника Microsoft Windows в какой-либо элемент формы.

Операции перетаскивания поддерживают следующие элементы формы:

- таблица формы,
- поле табличного документа,
- поле календаря,
- поле картинки,
- декорация-картинка.

При операциях перетаскивания используются следующие термины:

- **источник данных** – элемент формы, из которого можно перетаскивать данные;
- **приемник данных** – элемент формы, в который можно перетаскивать данные.

Существует возможность разрешать или запрещать элементам управления предоставлять или принимать данные, т. е. быть источником или приемником данных. Для этого у элементов формы существуют следующие свойства:

- **Разрешить начало перетаскивания** – разрешает элементу формы предоставлять данные.
- **Разрешить перетаскивание** – разрешает элементу формы принимать данные. Эти свойства можно устанавливать из палитры свойств или из встроенного языка.

При нажатии кнопки мыши на выделенной области элемента управления у элемента управления – источника данных вызывается обработчик события

НачалоПеретаскивания. В качестве параметров данного события передаются объекты типа **ПараметрыПеретаскивания** и **Выполнение**. Параметр **ПараметрыПеретаскивания** имеет следующие свойства:

- **Значение** – содержит перетаскиваемое значение, например, для таблицы управляемой формы это может быть ссылка на объект, для табличного документа – область табличного документа, для календаря – дата. Можно присвоить этому свойству свое значение (например, какую-нибудь структуру), тогда это значение будет являться перетаскиваемым объектом. Для таблицы управляемой формы, связанной с реквизитом формы типа **ДанныеФормыКоллекция**, **ДанныеФормыДерево** или **ДанныеФормыСтруктураСКоллекцией**, в качестве значения перетаскивания передается:
- идентификатор строки (или массив идентификаторов), если перетаскивание осуществляется в рамках одной таблицы;
- элемент коллекции (или массив элементов), если перетаскивание осуществляется между разными таблицами.
- **Действие** – указывает действие перетаскивания и является значением типа **ДействиеПеретаскивания**.
- **ДопустимыеДействия** – указывает допустимые действия перетаскивания и является значением типа **ДопустимыеДействияПеретаскивания**. При помощи этого свойства можно указать, какие операции возможны с данными источника данных (например, только копирование).

Параметр **Выполнение** позволяет разрешить или запретить стандартную обработку операции перетаскивания из данного элемента управления. Для события **НачалоПеретаскивания** стандартной обработкой является начало перетаскивания данных.

Далее у элемента формы – приемника данных вызывается обработчик события **ПроверкаПеретаскивания**. Данный обработчик вызывается всякий раз, когда курсор попадает на новый объект в элементе формы – приемнике данных (например, в новую ячейку таблицы или поля табличного документа, при попадании в новую дату в поле календаря). Набор параметров данного события зависит от элемента формы – приемника данных, но первые два параметра у всех одинаковы. Это объекты типа **ПараметрыПеретаскивания** и **СтандартнаяОбработка**. Остальные параметры описывают объект под курсором. При обработке данного события можно управлять видом курсора, т. е. например, указывать, что перетаскивание в данный элемент управления запрещено или возможно только копирование. Для этого нужно установить необходимое действие в свойстве **Действие** параметра **ПараметрыПеретаскивания**. Необходимо учитывать, что устанавливаемое действие перетаскивания должно быть разрешенным, т. е. не вступать в противоречие со значением свойства **ДопустимыеДействия**. Например, действие **Копирование** не вступает в противоречие со значением допустимых действий **КопированиеИПеремещение**, а значение **Перемещение** вступает в противоречие со значением допустимых действий **Копирование**. Параметр **СтандартнаяОбработка** используется для указания возможности стандартной обработки элементом формы данного события. Стандартная обработка перетаскивания зависит от типа элемента управления:

- Для таблицы проверяется возможность вставки значения, т. е. проверяется тип значения, и, если он совпадает с типом отображаемых данных, производятся стандартные действия. Стандартные действия для иерархических динамических списков – перемещение в группу, для таблиц, отображающих наборы записей или табличные части, – изменение порядка строк и копирование.

Если у таблицы свойство **ИзменитьПорядокСтрок** установлено в значение **Ложь**, то при перетаскивании внутри таблицы стандартная обработка отменяет перетаскивание и событие **Перетаскивание** вызвано не будет.

Если свойство **ИзменитьПорядокСтрок** установлено в значение **Истина** или перетаскивание осуществляется извне таблицы, стандартная обработка не отменяет перетаскивание.

Стандартная обработка не отменяет перетаскивание, если у таблицы свойство **РежимВыделения** установлено в значение **Множественный**, свойство **ИзменитьПорядокСтрок** установлено в значение **Истина** или перетаскивание осуществляется извне таблицы, и перетаскивается более одного элемента.

- Для поля табличного документа – проверка возможности вставить передаваемое значение.
- Для поля картинки и календаря стандартной обработки нет.

Если отпустить клавишу мыши в элементе формы – приемнике данных, вызывается обработчик события **Перетаскивание**. Набор параметров этого события тот же, что и у события **ПроверкаПеретаскивания**. Параметр **СтандартнаяОбработка** позволяет разрешить или запретить стандартную обработку события элементом формы. Стандартные действия перетаскивания описаны выше.

Затем в элементе управления – источнике данных вызывается обработчик события **ОкончаниеПеретаскивания**. При обработке этого события элемент формы – приемник данных может, например, удалить перемещенные данные или очистить какие-либо переменные.

7.7. Командный интерфейс формы

Командный интерфейс формы состоит:

- из панели навигации формы,
- командной панели самой формы.

Имеется возможность редактирования командного интерфейса формы: включение новых, использование имеющихся команд и т. д.

Заметим, что командный интерфейс формы редактируется на отдельной закладке в редакторе формы и описывает состав панели навигации вспомогательного окна, в котором отображается форма, и частично состав командной панели формы. В основном этот механизм нужен для настройки команд глобального интерфейса, которые необходимо отображать в форме в этих двух панелях. Состав командной панели формы определяется и непосредственно в структуре элементов, и в редакторе командного интерфейса.

Команда автоматически попадает в командный интерфейс формы, в случае если тип параметра параметризованной команды совпадает с типом основного реквизита формы.

Кроме того, можно принудительно добавить команду в нужную панель командного интерфейса. Для этого достаточно просто перетащить ее в нужную группу нужной командной панели.

Установленный признак **Автоположение** показывает, что будет использована последовательность команд, формируемая системой по умолчанию. Если его отключить, то можно редактировать порядок команд.

Для того чтобы настроить видимость команд, которые расположены в панелях командного интерфейса, можно снять флажок в колонке **Автовидимость** и изменить значение в колонке **Видимость**.

7.8. Различные подходы к модальности

7.8.1. Общая информация

В прикладных решениях, разрабатываемых с помощью системы «1С:Предприятие», часто возникает необходимость открыть форму, которая будет предоставлять одновременно несколько возможностей:

- Блокировать работу с другими фрагментами интерфейса прикладного решения;
- Позволять ввести какую-либо информацию (включая ответ на вопрос);
- Останавливать исполнение кода прикладного решения до момента, пока не будет завершена работа с открытой формой.

Причем рассматриваемые формы могут быть созданы прикладным разработчиком на встроенном языке «1С:Предприятие» и являться частью самой платформы (например формы, которые открывают методы **Предупреждение()**, **Вопрос()** и т. д.).

В рамках системы «1С:Предприятие» существуют две схемы использования таких форм:

- Модальные окна;
- Блокирующие окна.

В рамках одного прикладного решения может использоваться или какой-либо один подход или смешанная модель. Для управления такой возможностью существует специальное свойство конфигурации – [Режим использования модальности](#) (см. [здесь](#)).

Первая схема использования модальности (модальные окна) предполагает использование особых типов окон операционной системы – модальные окна. В случае использования этой схемы исполнение кода на встроенном языке останавливается до тех пор, пока модальная форма не будет закрыта. Значение возврата будет получено непосредственно в месте открытия модальной формы и сразу после этого (в том же методе) можно выполнять какие-либо действия с этим значением: проанализировать, использовать в алгоритме и т. д. Модальные окна можно открыть с помощью методов `ОткрытьФормуМодально()` или `Предупреждение()`. Однако, модальные окна не поддерживаются при работе на устройствах iPad и плохо поддерживаются в веб-клиенте.

В этих случаях рекомендуется использовать схему использования модальности с помощью блокирующих окон. Основное отличие блокирующего окна от модального заключается в том, что открытие блокирующего окна не останавливает исполнение кода на встроенном языке. Получение результата работы блокирующего окна выполняется с помощью специальной процедуры обратного вызова. Следствием этой особенности является то, что алгоритм, использующий модальные вызовы для управления своей работой, будет необходимо полностью переработать для использования с блокирующими окнами.

С точки зрения прикладного разработчика, разница между двумя схемами использования модальных окон будет продемонстрирована на следующем примере. Допустим, что есть некоторый клиентский обработчик, который задает пользователю вопрос и в зависимости от данного ответа выполняет одну или другую ветку прикладного кода.

Пример с модальными окнами:

[Копировать в буфер обмена](#)

```
&НаКлиенте
Процедура ВыполнитьАлгоритм()
    Результат = Вопрос("Выполнить быстро или не медленно?", РежимДиалогаВопрос.ДаНет);
    Если Результат = КодВозвратаДиалога.Да Тогда
        // Первый вариант алгоритма
    Иначе
        // Второй вариант алгоритма
    КонецЕсли;
КонецПроцедуры
```

В этом примере (модальные окна) работа встроенного языка остановится на строке `Результат = Вопрос()` и продолжится только после того, как пользователь ответит на заданный вопрос.

Пример с блокирующими окнами:

[Копировать в буфер обмена](#)

```
&НаКлиенте
Процедура ВыполнитьАлгоритм(Команда)
    ОбратныйВызов = Новый ОписаниеОповещения("ВыполнитьАлгоритмОбратныйВызов", ЭтотОбъект);
    ПоказатьВопрос(ОбратныйВызов, "Выполнить быстро или не медленно?", РежимДиалогаВопрос.ДаНет);
КонецПроцедуры
&НаКлиенте
Процедура ВыполнитьАлгоритмОбратныйВызов(Результат, ДополнительныеПараметры) Экспорт
    Если Результат = КодВозвратаДиалога.Да Тогда
        // Первый вариант алгоритма
    Иначе
        // Второй вариант алгоритма
    КонецЕсли;
КонецПроцедуры
```

В этом примере (с использованием блокирующих окон) алгоритм оказывается разделенным на две части:

- Место, где отображается вопрос пользователю;
- Место, где обрабатывается ответ пользователя.

Но это разные фрагменты прикладного кода. После вызова метода `ПоказатьВопрос()` исполнение кода на встроенном языке не завершится. Однако, пользователь сможет ответить на заданный вопрос только в том случае, когда исполнение кода на встроенном языке будет завершено и управление будет передано системе «1С:Предприятие». Следовательно, в месте вызова метода, отображающего форму с вопросом, невозможно получить результат работы пользователя с открытой формой.

После того, как пользователь ответит на вопрос, будет вызвана экспортная процедура, описание которой прикладной разработчик должен передать в метод открытия блокирующего окна. Описание метода обратного вызова создается с помощью объекта `ОписаниеОповещения`. Метод обратного вызова может располагаться в модуле формы, модуле команды и общем модуле. Если требуется указать, что метод обратного вызова находится в том же модуле, что и вызов блокирующего окна, следует использовать значение `ЭтотОбъект`. Также имеется возможность передать в обработчик обратного вызова различные вспомогательные данные, которые в самом обработчике будут доступны через формальный параметр `ДополнительныеПараметры`. Это могут быть входные данные для работы алгоритма, различные параметры и т. д.

В общем случае, после вызова любого метода работы с блокирующими окнами, метод, содержащий такой вызов, должен сразу завершаться, а весь дальнейший код должен быть расположен в обработчике оповещения.

В том случае, если требуется организовать каскадный вызов методов обратного вызова, следует использовать метод `ВыполнитьОбработкуОповещения()`. Также этот метод можно использовать в тех случаях, когда реализация какого-либо действия может произойти либо сразу, либо после задания вопроса о необходимости действия. Например, при попытке перечитать файл с данными, проверяется текущее состояние редактора, и если данные в редакторе изменены – задается вопрос о возможности перечитать данные.

Пример:

[Копировать в буфер обмена](#)

```
&НаКлиенте
Процедура ПеречитатьДанные(Команда)
    ОбратныйВызов = Новый ОписаниеОповещения("ПеречитатьДанныеЗавершение", ЭтотОбъект, ИмяФайлаСДанными);
    Если Модифицированность Тогда
        ПоказатьВопрос(ОбратныйВызов, "Данные изменены. Перечитать?", РежимДиалогаВопрос.ДаНет);
    Иначе
        ВыполнитьОбработкуОповещения(ОбратныйВызов, КодВозвратаДиалога.Да);
    КонецЕсли;
КонецПроцедуры
&НаКлиенте
Процедура ПеречитатьДанныеЗавершение(Результат, ИмяФайла) Экспорт
    Если Результат = КодВозвратаДиалога.Да Тогда
        // выполним чтение данных
        Модифицированность = Ложь;
    КонецЕсли;
КонецПроцедуры
```

7.8.2. Особенности использования блокирующих окон

В тонком и толстом клиенте, в отличие от веб-клиента, метод `НачатьПомещениеФайла()` останавливает исполнение кода на встроенном языке. Тем не менее, после завершения операции вызов процедуры обратного вызова, переданной в метод `НачатьПомещениеФайла()`, все равно будет выполнен. Для того чтобы избежать различий в поведении при работе в тонком клиенте и веб-клиенте, следует весь код, который должен быть выполнен после помещения файла, размещать строго в обработчике обратного вызова.

Если логика построения прикладного решения требует, чтобы при закрытии формы пользователю был задан вопрос, то следует это делать особым образом, который отражен в примере:

[Копировать в буфер обмена](#)

```
&НаКлиенте
Перем ОтветПередЗакрытием;
&НаКлиенте
Процедура ПередЗакрытием(Отказ, СтандартнаяОбработка)
    Если ОтветПередЗакрытием <> Истина Тогда
        Отказ = Истина;
        ОбратныйВызов = Новый ОписаниеОповещения("ПередЗакрытиемЗавершение", ЭтотОбъект);
        ПоказатьВопрос(ОбратныйВызов, "ЗакрытьФорму?", РежимДиалогаВопрос.ДаНет);
    КонецЕсли;
КонецПроцедуры
&НаКлиенте
Процедура ПередЗакрытиемЗавершение(Результат, ДополнительныеПараметры) Экспорт
    Если Результат = КодВозвратаДиалога.Да Тогда
        ОтветПередЗакрытием = Истина;
        Закрыть();
    Иначе
        ОтветПередЗакрытием = Неопределено;
    КонецЕсли;
КонецПроцедуры
```

Фактически, закрытие формы выполняется в два этапа:

- Вначале задается вопрос о необходимости (возможности) закрытия формы и собственно закрытие формы отменяется.
- После того, как пользователь ответил на вопрос, в специальной клиентской переменной отмечается, что сейчас будет выполнено «настоящее» закрытие формы и форма закрывается повторно.

Для объектов `ДиалогРедактированияСтандартногоПериода`, `КонструкторФорматнойСтроки`, `ДиалогВыбораШрифта`, `ДиалогВыбораЦвета`, `ДиалогРасписанияРегламентногоЗадания` в обработчик обратного вызова, который передается при вызове метода `Показать()`, в качестве первого параметра передается значение основного свойства закрываемого диалога, если в диалоге нажата кнопка `ОК` или `Неопределено` в остальных случаях. Например, при редактировании стандартного периода, в качестве первого параметра обработчика обратного вызова будет выступать объект `СтандартныйПериод` (или `Неопределено`).

7.9. Работа с формой из встроенного языка

7.9.1. Открытие формы

Для того чтобы открыть форму, существуют два способа:

- Воспользоваться методом `ОткрытьФорму()/ОткрытьФормуМодально()`.
- Воспользоваться комбинацией метода `ПолучитьФорму()` и метода `Открыть()` или `ОткрытьМодально()` объекта `УправляемаяФорма`.

В любом из перечисленных случаев можно передать в открываемую форму параметры формы.

Использовать метод `ОткрытьФорму()` рекомендуется во всех случаях, кроме необходимости открыть форму в модальном режиме и затем получить результат работы формы через реквизиты открываемой формы.

Такая особенность связана с тем, что в качестве значения возврата модальной формы будут выступать те данные, которые возвращает сама форма, и у разработчика не будет доступа к объекту `УправляемаяФорма`, чтобы получить реквизиты формы. Если предварительно получить форму методом `ПолучитьФорму()`, доступ к реквизитам можно будет получить после завершения работы метода `ОткрытьМодально()`.

Пример 1:

[Копировать в буфер обмена](#)

```
// Откроем форму списка справочника товары
// в режиме "только просмотр"
Параметры = Новый Структура("ТолькоПросмотр", Истина);
ОткрытьФорму("Справочник.Товары.ФормаСписка", Параметры);
```

Пример 2:

[Копировать в буфер обмена](#)

```
// Открыть модальную форму и после закрытия
// получить доступ к реквизитам формы
Форма = ПолучитьФорму("ОбщаяФорма.ВыборПериода");
Результат = Форма.ОткрытьМодально();
Если Результат = КодВозвратаДиалога.Да Тогда
    ДатаНачала = Форма.ДатаНачала;
    ДатаОкончания = Форма.ДатаОкончания;
КонецЕсли;
```

Описание работы с параметрами формы см. [здесь](#).

ПРИМЕЧАНИЕ. Не следует выполнять программное открытие и закрытие одной и той же формы в одном вызове встроенного языка.

7.9.2. Переопределение открываемой формы

В прикладных системах может возникать задача открытия различных форм при различных параметрах объекта (или текущего окружения). Например, для элемента справочника `Номенклатура` нужно открывать разные формы для собственно товара и для услуги. Или для некоторого автоматизированного рабочего места нужно открывать разные формы в зависимости от текущих настроек пользователя.

Для выполнения такой задачи в модуле менеджера объекта следует создать обработчик события `ОбработкаПолученияФормы`. В этом обработчике нужно выполнить все необходимые проверки и принять решение о том, какую форму следует открывать. Данный обработчик вызывается только в том случае, если происходит попытка открыть стандартную форму объекта с помощью имен форм по умолчанию.

Так, при открытии формы элемента справочника `Номенклатура` в следующем случае обработчик будет вызван.

[Копировать в буфер обмена](#)

```
ОткрытьФорму("Справочник.Номенклатура.ФормаОбъекта");
```

Однако обработчик не будет вызван, если попытаться открыть форму следующим образом.

[Копировать в буфер обмена](#)

```
ОткрытьФорму("Справочник.Товары.Форма.ФормаЭлемента");
```

Рассмотрим пример с разными формами формы элемента справочника более подробно.

При открытии формы будет выполняться анализ реквизита `Вид справочника Номенклатура`, и в том случае, если значение этого свойства равно значению `Перечисления.ВидыТоваров.Услуга`, открывать форму `ФормаУслуги справочника Номенклатура`.

```
Процедура ОбработкаПолученияФормы(ВидФормы, Параметры, ВыбраннаяФорма, ДополнительнаяИнформация, СтандартнаяОбработка)
Если ВидФормы = "ФормаОбъекта" И Параметры.Ключ.Вид = Перечисления.ВидыТоваров.Услуга Тогда
    ВыбраннаяФорма = Метаданные.Справочники.Товары.Формы.ФормаУслуги;
    СтандартнаяОбработка = Ложь;
КонецЕсли;
КонецПроцедуры
```

Данный обработчик следует разместить в модуле менеджера справочника [Номенклатура](#).

ПРИМЕЧАНИЕ. Обработчик [ОбработкаПолученияФормы](#) не вызывается при попытке открытия общих форм, форм констант, форм хранилищ настроек, внешних отчетов и внешних обработок.

7.9.3. Модификация свойств элементов формы

В процессе работы с формой возникают ситуации, когда необходимо изменять какие-либо свойства элементов формы, например, доступность элементов.

Для выполнения этих действий следует воспользоваться коллекцией [Элементы](#). Данная коллекция предоставляет доступ к списку всех элементов формы (без учета иерархии).

Для доступа к иерархии элементов предназначены свойства [Родитель](#) и [ПодчиненныеЭлементы](#) (для элементов группа, таблица и для собственно формы).

Так, отключение доступности элемента формы [ВидЦены](#) выполняется следующим образом:

```
Элементы.ВидЦены.Доступность = Ложь;
```

Копировать в буфер обмена

Следует обратить внимание на особенность установки свойств [ТолькоПросмотр](#), [Доступность](#) и [Видимость](#) для элементов, содержащих подчиненные элементы. Установка свойства для родительского элемента оказывает влияние также на все подчиненные элементы формы. При этом значение изменяемого свойства самого элемента не изменяется. Другими словами, фактическое значение свойств [ТолькоПросмотр](#), [Доступность](#) и [Видимость](#) конкретного элемента формы определяется как сложение «по И» значений этих свойств всех родителей данного элемента.

Например, мы имеем группу [ВалютаДокумента](#), которая состоит из полей ввода [Валюта](#) и [КурсВалюты](#). Если необходимо сделать всю группу недоступной, достаточно выполнить следующее:

```
Элементы.ВалютаДокумента.Доступность = Ложь;
```

Копировать в буфер обмена

СОВЕТ. При программной модификации свойств элементов формы следует избегать неоправданной модификации свойств, для которых в синтакс-помощнике указано [Изменение свойства на клиенте требует обращения к серверу](#). Это замедляет работу формы и требует лишних обращений к серверу.

ПРИМЕЧАНИЕ. Если модификация элементов формы приводит к изменению расположения элементов на форме (например, при изменении видимости элементов формы), то возможны вызовы различных обработчиков событий, связанных с активизацией элементов (например, будет вызвано событие [ПриАктивизацииСтроки](#) у таблицы).

Программное изменение свойства [Вид](#) у элемента формы, обладающего расширением (поле, таблица, группа, декорация), приводит к созданию нового расширения элемента формы. Свойства элемента формы, относящиеся к расширению, устанавливаются в значения по умолчанию и не переносятся из расширения элемента, которое было до изменения свойства [Вид](#).

ПРИМЕЧАНИЕ. Следует помнить, что свойства элементов формы, относящиеся к отображаемым данным (например, [ПутьКДанным](#), [ПутьКДаннымЗаголовка](#) и т. д.), можно изменять только для новых элементов формы или после смены вида элемента формы.

7.9.4. Модификация формы

Существует возможность программной модификации формы. Для модификации (создания, изменения и удаления) доступны:

- реквизиты формы,
- локальные команды формы,
- элементы формы.

ПРИМЕЧАНИЕ. Удалять можно только те объекты, которые созданы программно.

Общую схему программной работы с формой можно представить следующим образом:

- изменение состава реквизитов формы,
- изменение состава команд формы,
- изменение состава элементов формы.

Отдельно следует заметить, что работа по добавлению, изменению и удалению составных частей формы возможна только на сервере. Также стоит обратить внимание на тот факт, что на программную модель формы не оказывают влияния пользовательские настройки и функциональные опции формы.

При выполнении пользовательских настроек пользователь будет настраивать именно ту форму, которую он видит (с учетом программных модификаций формы). Если пользователь добавляет какие-либо элементы формы, то обращение к таким элементам из встроенного языка невозможно. Об этой особенности следует помнить и при разработке механизмов взаимодействия с пользователем. Так, например, если в элемент формы [Группа](#) вида [Страницы](#) пользователь добавит страницу, то при переходе на эту страницу свойство группы [ТекущаяСтраница](#) станет равно [Неопределено](#). Свойство формы [ТекущийЭлемент](#) также будет содержать значение [Неопределено](#) в том случае, когда активным элементом формы является элемент, добавленный пользователем.

Более подробно рассмотрим все этапы модификации формы.

7.9.4.1. Изменение состава реквизитов

Изменение (добавление и удаление) реквизитов выполняется методом [ИзменитьРеквизиты\(\)](#) объекта [УправляемаяФорма](#). При этом действия удаления и добавления выполняются за один вызов. Это позволяет выполнять операцию изменения свойств реквизитов формы. При этом следует понимать, что операция изменения состава реквизитов является ресурсоемкой операцией (фактически происходит полное создание формы), именно поэтому операции изменения состава реквизитов формы выполняются пакетным образом.

Рассмотрим подробнее методику создания реквизитов.

Вначале нам необходимо создать нужное количество объектов типа [РеквизитФормы](#). При создании реквизита мы указываем его имя, тип реквизита и то место в иерархии реквизитов формы (см. [здесь](#)), которое будет занимать создаваемый реквизит.

Так, если мы создаем таблицу значений из двух колонок, то программный код для их создания будет выглядеть следующим образом:

```
МоиРеквизиты = Новый Массив;
МоиРеквизиты.Добавить(Новый РеквизитФормы("ТаблицаДанных", Новый ОписаниеТипов("ТаблицаЗначений"), , "Таблица значений", Ложь));
```

Копировать в буфер обмена

МоиРеквизиты.Добавить(Новый РеквизитФормы("Поставщик", Новый ОписаниеТипов("СправочникСсылка.Контрагенты"), "ТаблицаДанных", "Кто поставляет", Ложь));
 МоиРеквизиты.Добавить(Новый РеквизитФормы("Товар", Новый ОписаниеТипов("СправочникСсылка.Товары"), "ТаблицаДанных", "Имя товара", Ложь));

Следует обратить внимание, что для двух последних реквизитов предпоследний параметр указывает, для какого реквизита будут создаваться колонки. Другими словами, существует возможность добавлять колонки для реквизитов тех типов, которые это позволяют.

На [рис. 201](#) изображены реквизиты, созданные в редакторе формы, аналог которых создает код, приведенный выше.

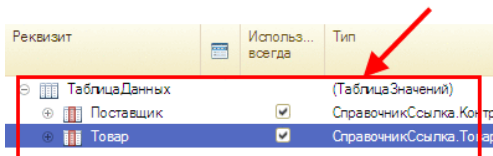


Рис. 201. Эквивалент программному коду

После того как были созданы все реквизиты, которые планировалось создать в форме, следует выполнить изменение списка реквизитов:

```
ИзменитьРеквизиты(ДобавляемыеРеквизиты, УдаляемыеРеквизиты);
```

[Копировать в буфер обмена](#)

Выполнение данного кода приведет к тому, что система вначале удалит реквизиты, перечисленные в массиве `УдаляемыеРеквизиты`, а затем выполнит добавление реквизитов из массива `ДобавляемыеРеквизиты`. После чего произойдет перестроение формы.

При удалении реквизитов происходит потеря данных, которые в них содержатся, однако если добавляемый и удаляемый реквизит имеют совместимые типы или добавляемый реквизит отличается от удаляемого реквизита только свойствами (но не типом), то данные, хранимые реквизитом, будут сохранены.

После добавления реквизита обращение к нему в программном коде возможно только с помощью конструкции `ЭтотОбъект.ИмяРеквизита`. Выражение «`ЭтотОбъект`.» является обязательным для программно добавляемых реквизитов.

Для того чтобы изменить состав или свойства реквизитов, следует вначале получить изменяемые реквизиты. Сделать это можно с помощью метода `ПолучитьРеквизиты()`.

Следует обратить внимание на две особенности полученного списка:

- данный список не является динамическим и не отслеживает изменения реквизитов, которые произошли после вызова метода;
- несмотря на то, что полученный список можно изменять, эти изменения никак не отразятся на реальных свойствах реквизитов формы.

После того как получен интересующий нас список реквизитов формы, можно выполнить с полученными реквизитами какие-либо действия (например, изменить заголовки всех реквизитов) и затем выполнить метод изменения реквизитов. Изменение реквизитов следует делать с предварительным удалением.

Например, если мы хотим изменить свойства реквизита `ПараметрЗаказа`, то это следует делать таким образом:

```
МассивРеквизитов = ПолучитьРеквизиты("ПараметрЗаказа");
```

[Копировать в буфер обмена](#)

```
...
// Выполним изменения реквизита
...
УдаляемыеРеквизиты = Новый Массив;
УдаляемыеРеквизиты.Добавить("ПараметрЗаказа");
ИзменитьРеквизиты(МассивРеквизитов, УдаляемыеРеквизиты);
```

7.9.4.2. Изменение состава команд

Для того чтобы управлять составом команд формы, у объекта `УправляемаяФорма` существует специальная коллекция – `Команды`. С помощью этой коллекции мы можем добавлять, удалять и изменять команды формы.

Так, для добавления команды `КомандаУстановкиСтатуса` с заголовком `Установить статус`, которая будет вызывать обработчик с именем `ОбработчикПрограммныхКоманд`, необходимо выполнить следующий программный код:

```
&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
    Команда = Команды.Добавить("КомандаУстановкиСтатуса");
    Команда.Действие = "ОбработчикПрограммныхКоманд";
    Команда.Заголовок = "Установить статус";
КонечПроцедуры
&НаКлиенте
Процедура ОбработчикПрограммныхКоманд(Команда)
    ...
КонечПроцедуры
```

[Копировать в буфер обмена](#)

Обработчик команды должен существовать в модуле формы и предваряться директивой компиляции `&НаКлиенте`.

ПРИМЕЧАНИЕ. Один обработчик может обслуживать несколько программно добавляемых команд.

7.9.4.3. Работа с элементами формы

После того как созданы все необходимые реквизиты и команды, можно добавлять элементы управления.

Для управления элементами формы у объекта `УправляемаяФорма` существует коллекция `Элементы`, с помощью которой можно добавлять, удалять, изменять свойства элементов формы, а также перемещать элементы формы между родителями.

Коллекция `Элементы` предоставляет доступ к списку элементов формы, который не учитывает возможную иерархию элементов. Для работы с иерархией у объекта коллекции `Элементы` существуют свойства `Родитель` и `ПодчиненныеЭлементы`.

Первое свойство указывает на родительский элемент формы, например, для поля формы, расположенного в группе, свойство `Родитель` будет указывать на элемент формы типа `ГруппаФормы`.

Свойство `ПодчиненныеЭлементы` существует у тех элементов формы, которые могут иметь подчиненные элементы. Например, для элемента формы типа `ГруппаФормы` коллекция `ПодчиненныеЭлементы` будет содержать те элементы, которые расположены в этой группе.

Для перемещения элемента из одной коллекции в другую (например, из одной группы в другую) существует метод `Переместить()`. Параметры данного метода описывают перемещаемый элемент, нового родителя элемента и тот элемент формы, перед которым нужно поместить перемещаемый элемент.

Если последний параметр метода не указан, то перемещаемый элемент будет размещен в конце коллекции элементов нового родителя.

ПРИМЕЧАНИЕ. При программном изменении состава элементов формы возможны вызовы различных обработчиков событий, связанных с активизацией элементов

(например, будет вызвано событие `ПриАктивизацииСтроки` у таблицы).

Рассмотрим подробнее способ добавления элемента управления.

В примере показано, как разместить на форме два элемента:

- поле формы, связанное с реквизитом формы;
- кнопку, связанную с командой формы.

Копировать в буфер обмена

```
&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
    МоиРеквизиты = Новый Массив;
    ТипСтрока = Новый ОписаниеТипов("Строка", , Новый КвалификаторыСтроки());
    МоиРеквизиты.Добавить(Новый РеквизитФормы("ОписаниеОбъекта", ТипСтрока, "", "Описание объекта", Ложь));
    ИзменитьРеквизиты(МоиРеквизиты);
    Команда = Команды.Добавить("ИзменитьСтроку");
    Команда.Действие = "ОбработчикПрограммныхКоманд";
    Команда.Заголовок = "Изменить строку";
    Элемент = Элементы.Добавить("ОписаниеОбъекта", Тип("ПолеФормы"));
    Элемент.Вид = ВидПоляФормы.ПолеВвода;
    Элемент.ПутьКДанным = "ОписаниеОбъекта";
    Элемент = Элементы.Добавить("ИзменитьСтроку", Тип("КнопкаФормы"));
    Элемент.ИмяКоманды = "ИзменитьСтроку";
КонечПроцедуры
...
&НаКлиенте
Процедура ОбработчикПрограммныхКоманд(Команда)
    ЭтотОбъект.ОписаниеОбъекта = "Описание объекта, сформированное командой";
КонечПроцедуры
```

Форма, в которой будет размещен этот код, будет выглядеть как на [рис. 202](#). Форма показана после того, как пользователь нажал кнопку `Изменить строку`.

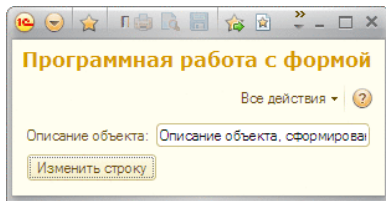


Рис. 202. Результат программной модификации формы

7.9.5. Работа с динамическим списком

В данном разделе приведены примеры выполнения некоторых операций по работе с динамическим списком, размещенным в форме.

ПРИМЕЧАНИЕ. Примеры, приведенные ниже, не являются законченными. Они предназначены для демонстрации различных механизмов работы с динамическим списком.

7.9.5.1. Параметр запроса динамического списка

В данном разделе приведен пример установки параметра запроса динамического списка.

Предположим, что для динамического списка указан следующий запрос к данным:

Копировать в буфер обмена

```
ВЫБРАТЬ
    Товары.Код КАК Код,
    Товары.Наименование КАК Наименование,
    Товары.Артикул КАК Артикул,
    ТоварныеЗапасыОстатки.КоличествоОстаток КАК Количество,
    ЦеныТоваровСрезПоследних.Цена КАК Цена,
    Товары.ЭтоГруппа
ИЗ
    Справочник.Товары КАК Товары
    ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ТоварныеЗапасы.Остатки(, Склад = &Склад) КАК ТоварныеЗапасыОстатки
    ПО (ТоварныеЗапасыОстатки.Товар = Товары.Ссылка)
    ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.ЦеныТоваров.СрезПоследних(, ВидЦен = &ВидЦен) КАК ЦеныТоваровСрезПоследних
    ПО Товары.Ссылка = ЦеныТоваровСрезПоследних.Товар
```

Тогда для установки параметров данного запроса (в обработчике `ПриСозданииНаСервере()`) нужно указать параметры `Склад` и `ВидЦен` следующим образом.

Копировать в буфер обмена

```
// СписокТоваров – реквизит формы типа ДинамическийСписок
// Параметры.Склад и Параметры.ВидЦен – параметры формы соответствующих типов
&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
    СписокТоваров.Параметры.УстановитьЗначениеПараметра("Склад", Параметры.Склад);
    СписокТоваров.Параметры.УстановитьЗначениеПараметра("ВидЦен", Параметры.ВидЦен);
КонечПроцедуры
```

7.9.5.2. Отбор

В данном разделе приведены примеры установки и удаления отбора в динамическом списке.

Установить

Функция добавляет отбор в динамический список и возвращает созданный элемент отбора.

Копировать в буфер обмена

```
Функция ДобавитьОтбор(СписокОтборов, ИмяПоля, Значение, ВидСравнения = Неопределено, Использование = Истина)
    НовыйЭлемент = СписокОтборов.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));
    НовыйЭлемент.ЛевоеЗначение = Новый ПолеКомпоновкиДанных(ИмяПоля);
    НовыйЭлемент.ВидСравнения = ?(ВидСравнения = Неопределено, ВидСравненияКомпоновкиДанных.Равно, ВидСравнения);
    НовыйЭлемент.ПравоеЗначение = Значение;
    НовыйЭлемент.Использование = Использование;
    Возврат НовыйЭлемент;
КонечФункции
```

Пример использования:

```
// ДинамическийСписок – реквизит формы типа "ДинамическийСписок"
// В списке должна быть колонка "Контрагент"
// Выполняется установка отбора по полю "Контрагент", значение
// отбора находится в переменной "КонтрагентСсылка", отбор включен,
// условие отбора – равно.
ДобавитьОтбор(ДинамическийСписок.КомпоновщикНастроек.ФиксированныеНастройки.Отбор, "Контрагент", КонтрагентСсылка);
```

Удалить

Процедура удаляет отбор динамического списка и возвращает результат удаления. Если имя поля с удаляемым отбором не указано, выполняется очистка списка отборов.

[Копировать в буфер обмена](#)

```
Функция УдалитьОтбор(СписокОтборов, ИмяПоля = "")
Если ПустаяСтрока(ИмяПоля) Тогда
    СписокОтборов.Элементы.Очистить();
    Возврат Истина;
КонецЕсли;
Поле = Новый ПолеКомпоновкиДанных(ИмяПоля);
ОтборУдален = Ложь;
Для каждого ЭлементОтбора Из СписокОтборов.Элементы Цикл
    Если ЭлементОтбора.Использование И ЭлементОтбора.ЛевоеЗначение = Поле Тогда
        СписокОтборов.Элементы.Удалить(ЭлементОтбора);
        ОтборУдален = Истина;
    КонецЕсли;
КонецЦикла;
Возврат ОтборУдален;
КонецФункции
```

Пример использования:

[Копировать в буфер обмена](#)

```
// ДинамическийСписок – реквизит формы типа "ДинамическийСписок"
// В списке должна быть колонка "Контрагент"
// Выполняется удаление отбора по полю "Контрагент".
Результат = УдалитьОтбор(ДинамическийСписок.КомпоновщикНастроек.ФиксированныеНастройки.Отбор, "Контрагент");
```

7.9.5.3. Группировка

В данном разделе приведены примеры установки и удаления группировки в динамическом списке.

Установка

Функция добавляет группировку в динамический список и возвращает созданный элемент группировки.

[Копировать в буфер обмена](#)

```
Функция ДобавитьГруппировку(СписокГруппировок, ИмяПоля, Использование = Истина, ТипГруппировки = Неопределено)
Поле = Новый ПолеКомпоновкиДанных(ИмяПоля);
НовыйЭлемент = СписокГруппировок.Элементы.Добавить(Тип("ПолеГруппировкиКомпоновкиДанных"));
НовыйЭлемент.Использование = Использование;
НовыйЭлемент.Поле = Поле;
НовыйЭлемент.ТипГруппировки = ?(ТипГруппировки = Неопределено, ТипГруппировкиКомпоновкиДанных.Элементы, ТипГруппировки);
Возврат НовыйЭлемент;
КонецФункции
```

Пример использования:

[Копировать в буфер обмена](#)

```
// ДинамическийСписок – реквизит формы типа "ДинамическийСписок"
// В списке должна быть колонка "Контрагент"
// Выполняется создание группировки по полю "Контрагент", созданная
// группировка будет использоваться, тип группировки – только элементы
Группировка = Список.КомпоновщикНастроек.Настройки.Структура.Добавить(Тип("ГруппировкаКомпоновкиДанных"));
ДобавитьГруппировку(Группировка.ПоляГруппировки, "Контрагент");
```

Удаление

Процедура удаляет группировку динамического списка и возвращает результат удаления. Если имя поля с удаляемой группировкой не указано, выполняется очистка списка группировок.

[Копировать в буфер обмена](#)

```
Функция УдалитьГруппировку(СписокГруппировок, ИмяПоля = "")
Если ПустаяСтрока(ИмяПоля) Тогда
    СписокГруппировок.Элементы.Очистить();
    Возврат Истина;
КонецЕсли;
Поле = Новый ПолеКомпоновкиДанных(ИмяПоля);
ГруппировкаУдалена = Ложь;
МассивЭлементовДляУдаления = Новый Массив;
Для каждого ЭлементГруппировки Из СписокГруппировок.Элементы Цикл
    Если ЭлементГруппировки.Поле = Поле тогда
        МассивЭлементовДляУдаления.Добавить(ЭлементГруппировки);
    КонецЕсли;
КонецЦикла;
Для каждого УдаляемыйЭлемент Из МассивЭлементовДляУдаления Цикл
    СписокГруппировок.Элементы.Удалить(УдаляемыйЭлемент);
    ГруппировкаУдалена = Истина;
КонецЦикла;
Возврат ГруппировкаУдалена;
КонецФункции
```

Пример использования:

[Копировать в буфер обмена](#)

```
// ДинамическийСписок – реквизит формы типа "ДинамическийСписок"
// Выполняется удаление всех группировок в динамическом списке
Результат = УдалитьГруппировку(ДинамическийСписок.КомпоновщикНастроек.ФиксированныеНастройки.Группировка);
```

7.9.5.4. Порядок (сортировка)

В данном разделе приведены примеры установки и удаления упорядочивания динамического списка.

Установка

Функция добавляет поле сортировки в динамический список и возвращает созданный элемент порядка.

```

Функция ДобавитьПорядок(СписокПорядков, ИмяПоля, Использование = истина, ТипУпорядочивания = Неопределено)
Поле = Новый ПолеКомпоновкиДанных(Поле);
НовыйЭлемент = СписокПорядков.Элементы.Добавить(Тип("ЭлементПорядкаКомпоновкиДанных"));
НовыйЭлемент.Использование = Использование;
НовыйЭлемент.Поле = Поле;
НовыйЭлемент.ТипУпорядочивания = ?(ТипУпорядочивания = Неопределено, НаправлениеСортировкиКомпоновкиДанных.Возр, ТипУпорядочивания);
Возврат НовыйЭлемент;
КонцевФункции

```

Пример использования:

```

// ДинамическийСписок – реквизит формы типа "ДинамическийСписок"
// В списке должна быть колонка "Наименование"
//
// Выполняется установка сортировки по полю "Наименование",
// направление сортировки – по убыванию, сортировка используется
ДобавитьПорядок(ДинамическийСписок.КомпоновщикНастроек.Настройки.Порядок, "Наименование", НаправлениеСортировкиКомпоновкиДанных.Убыв);

```

Копировать в буфер обмена

Удаление

Процедура удаляет сортировку динамического списка и возвращает результат удаления. Если имя поля упорядочивания не указано, выполняется очистка списка упорядочивания.

```

&НаКлиенте
Функция УдалитьПорядок(СписокПорядков, ИмяПоля = "")
Если ПустаяСтрока(ИмяПоля) Тогда
СписокПорядков.Элементы.Очистить();
Возврат Истина;
КонцевЕсли;
Поле = Новый ПолеКомпоновкиДанных(ИмяПоля);
ПорядокУдален = Ложь;
МассивЭлементовДляУдаления = Новый Массив;
Для каждого ЭлементПорядка из СписокПорядков.Элементы Цикл
Если ЭлементПорядка.Поле = Поле тогда
МассивЭлементовДляУдаления.Добавить(ЭлементПорядка);
КонцевЕсли;
КонцевЦикла;
Для каждого УдаляемыйЭлемент из МассивЭлементовДляУдаления Цикл
СписокПорядков.Элементы.Удалить(УдаляемыйЭлемент);
ПорядокУдален = Истина;
КонцевЦикла;
Возврат ПорядокУдален;
КонцевФункции

```

Копировать в буфер обмена

Пример использования:

```

// ДинамическийСписок – реквизит формы типа "ДинамическийСписок"
//
// Выполняется удаление (не отключение) порядка по полю "Наименование"
Результат = УдалитьПорядок(ДинамическийСписок.КомпоновщикНастроек.ФиксированныеНастройки.Порядок, "Наименование");

```

Копировать в буфер обмена

7.9.5.5. Условное оформление

В данном разделе приведены примеры задания условного оформления динамического списка. В условном оформлении используется динамический список `Список`, в котором есть поля `Организация` и `Сумма` (типа `Число`). В примере задаются три элемента условного оформления:

- если значение поля `Сумма` меньше **500**, то текст полей `Организация` и `Сумма` оформляется красным цветом;
- если значение поля `Сумма` больше **500**, но меньше **10 000**, то для всех полей строки текст оформляется синим цветом;
- если значение поля `Сумма` больше **10 000**, то текст полей `Организация` и `Сумма` оформляется зеленым цветом.

Также в примере используется функция установки отбора (см. [здесь](#)).

Копировать в буфер обмена

```

Функция УстановитьУсловноеОформление(СписокОформления, ИменаОформляемыхПолей, СтруктураОформления, Использование = Истина) Экспорт
НовыйЭлемент = СписокОформления.Элементы.Добавить();
НовыйЭлемент.Использование = Использование;
// Зададим оформляемые поля, заданные массивом с именами полей
Для каждого ИмяПоля Из ИменаОформляемыхПолей Цикл
ОформляемоеПоле = НовыйЭлемент.Поля.Элементы.Добавить();
ОформляемоеПоле.Использование = Истина;
ОформляемоеПоле.Поле = Новый ПолеКомпоновкиДанных(ИмяПоля);
КонцевЦикла;
// Зададим оформление, заданное структурой,
// в которой – Ключ: имя параметра оформления,
// а Значение – значение параметра оформления
Для каждого ЭлементОформления Из СтруктураОформления Цикл
НовыйЭлемент.Оформление.УстановитьЗначениеПараметра(ЭлементОформления.Ключ, ЭлементОформления.Значение);
КонцевЦикла;
Возврат НовыйЭлемент;
КонцевФункции

```

Пример задания условного оформления:

```

ЭлементыУсловногоОформления = Список.КомпоновщикНастроек.ФиксированныеНастройки.УсловноеОформление;
ОформляемыеПоля = Новый Массив;
ОформляемыеПоля.Добавить("Сумма");
ОформляемыеПоля.Добавить("Организация");
// ***** Установка красного цвета для суммы менее 500
Оформление = Новый Структура;
Оформление.Вставить("ЦветТекста", Новый Цвет(128,0,0));
НовыйЭлемент = УстановитьУсловноеОформление(ЭлементыУсловногоОформления, ОформляемыеПоля, Оформление);
// Зададим условие оформления: Сумма < 500
ДобавитьОтбор(НовыйЭлемент.Отбор, "Сумма", 500, ВидСравненияКомпоновкиДанных.Меньше);
// ***** Установка зеленого цвета для суммы более 10 000
Оформление = Новый Структура;
Оформление.Вставить("ЦветТекста", Новый Цвет(0,128,0));
НовыйЭлемент = УстановитьУсловноеОформление(ЭлементыУсловногоОформления, ОформляемыеПоля, Оформление);
// Зададим условие оформления: Сумма > 10 000
ДобавитьОтбор(НовыйЭлемент.Отбор, "Сумма", 10000, ВидСравненияКомпоновкиДанных.Больше);
// установка синего цвета, если сумма в интервале от 500 до 10 000
Оформление = Новый Структура;
Оформление.Вставить("ЦветТекста", Новый Цвет(0,0,128));
// Передача пустого массива в качестве списка оформляемых полей

```

Копировать в буфер обмена

1512

```
// означает оформление всех полей строки
НовыйЭлемент = УстановитьУсловноеОформление(ЭлементыУсловногоОформления, НовыйМассив, Оформление);
// Создадим группу условий "И"
ГруппаОтборов = НовыйЭлемент.Отбор.Элементы.Добавить(Тип("ГруппаЭлементовОтбораКомпоновкиДанных"));
ГруппаОтборов.ТипГруппы = ТипГруппыЭлементовОтбораКомпоновкиДанных.ГруппаИ;
// Зададим условие оформления: Сумма > 500 И Сумма < 10 000
ДобавитьОтбор(ГруппаОтборов, "Сумма", 500, ВидСравненияКомпоновкиДанных.Больше);
ДобавитьОтбор(ГруппаОтборов, "Сумма", 10000, ВидСравненияКомпоновкиДанных.Меньше);
```


Глава 8. Работа с запросами

Для формирования и выполнения запросов к таблицам базы данных в системе используется специальный объект **Запрос**. **Запрос** удобно использовать, когда необходимо получить сложную выборку данных, сгруппированную и отсортированную нужным образом. Одним из классических примеров его применения может служить сводка по состоянию регистра учета на определенный момент времени. Кроме того, механизм запросов позволяет легко получать информацию в различных временных разрезах.

8.1. Источники данных (таблицы) запросов

В качестве источников данных языка запросов выступают таблицы базы данных. Таблицы подразделяются на два основных класса: реальные и виртуальные.

Реальные таблицы хранятся в базе данных, то есть интерпретируются из реально существующей таблицы базы данных. В случае использования реальной таблицы могут присутствовать вычисляемые поля, значения которых вычисляются как функция нескольких реальных полей.

Виртуальные таблицы не хранятся в базе данных. При обращении к информации виртуальных таблиц система автоматически собирает информацию реальных таблиц базы данных для выполнения запроса. Виртуальная таблица может быть параметризована, то есть реальное наполнение виртуальной таблицы может определяться значениями параметров, фактически значения которых задаются в тексте запроса. Для каждой виртуальной таблицы определяется имя, которое используется в запросах для идентификации таблицы.

Имя таблицы может быть задано на английском и русском языках. Например, **Справочник.Товары**. Имена таблиц и полей не могут совпадать с ключевыми словами языка запросов.

Отдельный подкласс таблиц образуют так называемые объектные таблицы. В качестве объектной таблицы обязательно выступает реальная таблица базы данных. Смысловое отличие объектных таблиц от прочих следует из названия – объектные таблицы предназначены для хранения состояния объектов системы «1С:Предприятие», таких как справочники, документы и т. п. Каждой объектной таблице соответствует тип объектов системы «1С:Предприятие». Например, объектам типа **Справочник.Товары** соответствует одна таблица, объектам типа **Справочник.Контрагенты** – другая. Каждая отдельная запись объектной таблицы хранит состояние отдельного объекта соответствующего типа. В соответствии с этим у каждой объектной таблицы определено поле типа **Ссылка** на текущую запись. Кроме того, для объектных таблиц определен способ получения пользовательского представления объекта из содержимого полей записи.

Объектные таблицы могут быть также иерархическими. Для иерархических таблиц определяется специально выделенное поле **Родитель**, содержащее ссылку на запись, которой в соответствии с иерархией подчиняется текущая запись.

В качестве поля таблицы может фигурировать:

- поле виртуальной или реальной таблицы,
- вложенная таблица.

Основное отличие обычного поля от вложенной таблицы состоит в том, что в рамках одной записи обычному полю соответствует одно-единственное значение, а вложенной таблице соответствует значение типа **РезультатЗапроса** с заранее заданным набором колонок. Примером вложенной таблицы может являться табличная часть документа или справочника.

В качестве типов значения полей таблиц может выступать значение типа **NULL**. Такие значения содержатся в полях записей таблиц, для которых данное поле не определено или не имеет смысла. Например, значения такого типа содержатся в записях, относящихся к группам справочника, по полям, для которых установлено, что они могут иметь значение только у элементов этого справочника.

8.2. Язык запросов

Как было описано выше, для выполнения запроса необходимо составить текст запроса. Текст запроса – это инструкция, в соответствии с которой должен быть выполнен запрос. В тексте запроса описывается, какие таблицы информационной базы используются в качестве источников данных запроса, поля таблиц, которые требуется обрабатывать в запросе, правила группировки, сортировки результатов и т. д.

Инструкция составляется на специальном языке (языке запросов) и состоит из отдельных частей – секций, предложений, слов, функций и комментариев. Далее в этой главе рассматривается назначение и способы использования всех конструкций языка запросов.

8.2.1. Синтаксическая диаграмма конструкций языка запросов

В данной главе синтаксис языка запросов описывается при помощи набора правил. Каждое правило описывает одну конструкцию языка.

Каждая конструкция языка может содержать в себе ключевые слова; разделители (точки, запятые, круглые скобки), в свою очередь, другие конструкции языка.

<Конструкция языка>

```
ЭТО_КЛЮЧЕВОЕ_СЛОВО
<Это_конструкция_языка>, <Это_конструкция_языка>
ЭТО_ФУНКЦИЯ ( <Это_конструкция_языка> )
```

[Копировать в буфер обмена](#)

В правилах, описывающих язык запросов, конструкции языка указываются в угловых скобках. Ключевые слова и названия функций описываются заглавными буквами.

Конструкции языка могут содержать необязательные элементы – ключевые слова и пр. В правилах, описывающих язык запросов, необязательные элементы заключены в квадратные скобки.

```
[ЭТО_НЕОБЯЗАТЕЛЬНОЕ_СЛОВО]
[<Это_необязательная_конструкция>]
```

[Копировать в буфер обмена](#)

В некоторых случаях в конструкции языка может использоваться один из нескольких альтернативных элементов. Такие элементы в правилах перечисляются через вертикальную черту.

```
ЛИБО_ЭТО_СЛОВО | ЛИБО_ЭТО_СЛОВО
<Либо_эта_конструкция> | <Либо_эта_конструкция>
```

[Копировать в буфер обмена](#)

Описания всех конструкций сопровождаются примерами, поясняющими порядок их использования в языке запросов.

8.2.2. Комментарии в языке запросов

Текст запроса может включать комментарии. Комментарием считается часть строки, начинающаяся с последовательности символов // и продолжающаяся до

// Это комментарий

При выполнении запроса комментарии игнорируются.

ПРИМЕЧАНИЕ. Конструктор запросов удаляет комментарии из текста.

8.2.3. Использование предопределенных данных конфигурации

Текст запроса может содержать предопределенные данные конфигурации, такие как:

- значения перечислений,
- предопределенные данные:
 - справочников,
 - планов видов характеристик,
 - планов счетов,
 - планов видов расчетов,
- пустые ссылки,
- значения точек маршрута бизнес-процессов.

Также текст запроса может содержать значения системных перечислений, которые могут быть присвоены полям в таблицах базы данных: ВидДвиженияНакопления, ВидСчета и ВидДвиженияБухгалтерии.

Обращение в запросах к предопределенным данным конфигурации и значениям системных перечислений осуществляется с помощью литерала функционального типа.

[Копировать в буфер обмена](#)

`ЗНАЧЕНИЕ (<ПредставлениеЗначения>)`

Для системных перечислений представление значения имеет вид:

[Копировать в буфер обмена](#)

`<ИмяСистемногоПеречисления>.<Значение>`

Допустимые имена системных перечислений приведены выше, с перечнем допустимых для каждого из них значений можно ознакомиться в его описании.

Для предопределенных данных конфигурации представление значения имеет вид:

[Копировать в буфер обмена](#)

`<ТипПредопределенногоЗначения>.<ИмяОбъектаМетаданных>.<Значение>`

Тип предопределенного значения может быть:

- Справочник (Catalog);
- ПланВидовХарактеристик (ChartOfCharacteristicTypes);
- ПланСчетов (ChartOfAccounts);
- ПланВидовРасчета (ChartOfCalculationTypes);
- Перечисление (Enum).

В качестве имени объекта метаданных указывается имя объекта метаданных, как оно задано в конфигураторе.

Для определенных в конфигурации перечислений значение указывается как имя соответствующего объекта метаданных типа ЗначениеПеречисления. Для всех остальных типов предопределенных значений – как имя предопределенного элемента данных, как оно указано в конфигураторе, или ПустаяСсылка (EmptyRef) для указания пустой ссылки.

Для точек маршрутов бизнес-процессов представление значения имеет вид:

[Копировать в буфер обмена](#)

`БизнесПроцесс.<ИмяОбъектаМетаданных>.ТочкаМаршрута.<ИмяТочкиМаршрута>`

Ниже приведено несколько фрагментов запросов, поясняющих использование предопределенных данных в запросах.

[Копировать в буфер обмена](#)

ГДЕ Город = ЗНАЧЕНИЕ (Справочник.Города.Москва)
ГДЕ Город = ЗНАЧЕНИЕ (Справочник.Города.ПустаяСсылка)
ГДЕ ТипТовара = ЗНАЧЕНИЕ (Перечисление.ВидыТоваров.Услуга)
ГДЕ ВидДвижения = ЗНАЧЕНИЕ (ВидДвиженияНакопления.Приход)
ГДЕ ТочкаМаршрута = ЗНАЧЕНИЕ (БизнесПроцесс.Согласование.ТочкаМаршрута.Согласие)

8.2.4. Двуязычное представление ключевых слов

Одной из существенных особенностей языка запросов системы «1С:Предприятие» является то, что, как и во встроенном языке, все ключевые слова имеют два варианта написания: на русском и английском языках. Далее в этой главе указываются русские варианты написания ключевых слов. Ниже в таблице приведены соответствия русского и английского вариантов написания ключевых слов языка запросов.

Русское написание	Английское написание
АВТОУПОРЯДОЧИВАНИЕ	AUTOORDER
БУЛЕВО	BOOLEAN

В	IN
ВНЕШНЕЕ	OUTER
ВНУТРЕННЕЕ	INNER
ВОЗР	ASC
ВСЕ	ALL
ВЫБОР	CASE
ВЫБРАТЬ	SELECT
ВЫРАЗИТЬ	CAST
ГДЕ	WHERE
ГОД	YEAR
ДАТА	DATE
ДАТАВРЕМЯ	DATETIME
ДЕКАДА	TENDAYS
ДЕНЬ	DAY
ДЕНЬГОДА	DAYOFYEAR
ДЕНЬНЕДЕЛИ	WEEKDAY
ДЛЯ ИЗМЕНЕНИЯ	FOR UPDATE [OF]
ДОБАВИТЬКДАТЕ	DATEADD
ЕСТЬ	IS
ЕСТЬNULL	ISNULL
ЗНАЧЕНИЕ	VALUE
И	AND
ИЕРАРХИИ	HIERARCHY
ИЕРАРХИЯ	HIERARCHY
ИЗ	FROM
ИЛИ	OR
ИМЕЮЩИЕ	HAVING
ИНАЧЕ	ELSE
ИНДЕКСИРОВАТЬ ПО	INDEX BY
Истина	TRUE
ИТОГИ ... ПО	TOTALS ... BY
КАК	AS
КВАРТАЛ	QUARTER
КОГДА	WHEN
КОЛИЧЕСТВО	COUNT
КОНЕЦПЕРИОДА	ENDOFPERIOD
КОНЕЦ	END
ЛЕВОЕ	LEFT
Ложь	FALSE
МАКСИМУМ	MAX
МЕЖДУ	BETWEEN
МЕСЯЦ	MONTH
МИНИМУМ	MIN
МИНУТА	MINUTE
НАЧАЛОПЕРИОДА	BEGINOFPERIOD
НЕ	NOT
НЕДЕЛЯ	WEEK
НЕОПРЕДЕЛЕНО	UNDEFINED
ОБЩИЕ	OVERALL
ОБЪЕДИНИТЬ	UNION
ПЕРВЫЕ	TOP
ПЕРИОДАМИ	PERIODS
ПОДОБНО	LIKE
ПОЛНОЕ	FULL
ПОЛУГОДИЕ	HALFYEAR
ПОМЕСТИТЬ	INTO
ПРАВОЕ	RIGHT
ПРЕДСТАВЛЕНИЕ	PRESENTATION
ПУСТАЯТАВЛИЦА	EMPTYTABLE
РАЗЛИЧНЫЕ	DISTINCT
РАЗРЕШЕННЫЕ	ALLOWED
СВЯЗАННЫЕ	RELATED

	GROUP BY
СЕКUNДА	SECOND
СОЕДИНЕНИЕ ... ПО	JOIN ... ON
СПЕЦИСМВОЛ	ESCAPE
ПОДСТРОКА	SUBSTRING
СЕКUNДА	SECOND
СРЕДНЕЕ	AVG
ССЫЛКА	REFS
СТРОКА	STRING
СУММА	SUM
ТИП	TYPE
ТИПЗНАЧЕНИЯ	VALUETYPE
ТОГДА	THEN
ТОЛЬКО	ONLY
УВЫВ	DESC
УПОРЯДОЧИТЬ ПО	ORDER BY
ЧАС	HOURL
ЧИСЛО	NUMBER
УНИЧТОЖИТЬ	DROP

ПРИМЕЧАНИЕ. Регистр букв (строчные или заглавные) при написании не имеет значения.

8.2.5. Основные секции текста запроса

Текст запроса можно описать следующим правилом:

<Текст Запроса>

[Копировать в буфер обмена](#)

```
<Описание запроса>
[<Объединение запросов>]
[<Упорядочивание результатов>]
[АВТОУПОРЯДОЧИВАНИЕ]
<Описание итогов>
```

Как видно из этого правила, текст запроса состоит из нескольких частей, или секций:

- **<Описание запроса>** – это единственная обязательная секция в тексте запроса, и во многих случаях достаточно указать только ее. В секции определяются источники данных запроса, поля выборки, группировки и т. д. Эта секция, в свою очередь, описывается целым набором правил и подробно рассматривается ниже.
- **<Объединение запросов>** – язык запросов позволяет объединять результаты выполнения нескольких запросов. Описание объединения запросов см. [здесь](#).
- В секции **<Упорядочивание результатов>** можно определить условия упорядочивания строк в результате запроса. Описание упорядочивания результата запроса см. [здесь](#).
- Секция **АВТОУПОРЯДОЧИВАНИЕ** позволяет включить режим автоматического упорядочивания строк в результате запроса. Описание данного режима см. [здесь](#).
- В секции **<Описание итогов>** можно указать, какие итоги необходимо рассчитывать в запросе. Описание данной секции см. [здесь](#).

8.2.6. Описание запроса

Как уже было сказано, в тексте запроса должна обязательно присутствовать секция описания запроса, в которой определяются:

- поля, которые будут содержаться в результате запроса;
- источники данных запроса – исходные таблицы;
- условия, влияющие на выборку данных в запросе;
- порядок группировки результатов запроса.

Секция описания запроса состоит из нескольких взаимосвязанных предложений:

<Описание запроса>

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ [РАЗРЕШЕННЫЕ] [РАЗЛИЧНЫЕ] [ПЕРВЫЕ <Количество>]
<Список полей выборки>
[ИЗ <Список источников>]
[ГДЕ <Условие отбора>]
[СТРУППИРОВАТЬ ПО <Поля группировки>]
[ИМЕЮЩИЕ <Условие отбора>]
[ДЛЯ ИЗМЕНЕНИЯ [<Список таблиц верхнего уровня>]]
```

Описание запроса начинается с обязательного ключевого слова **ВЫБРАТЬ**.

Ключевое слово **РАЗРЕШЕННЫЕ** означает, что запрос выберет только те записи, на которые у текущего пользователя есть права. Если данное слово не указать, то запрос отработает с ошибкой, когда он выберет записи, на которые у пользователя нет прав. Данное ключевое слово может присутствовать только в предложении **ВЫБРАТЬ** верхнего уровня и распространяется на весь запрос, включая вложенные запросы. Следует учитывать, что использование ключевого слова **РАЗРЕШЕННЫЕ** оказывает влияние на работу только в том случае, если на таблицы наложены ограничения доступа к данным (см. [здесь](#)).

Права доступа на саму таблицу не учитываются при использовании ключевого слова **РАЗРЕШЕННЫЕ**. Например, если на таблицу отсутствует право **ЧТЕНИЕ**, то запрос с использованием такой таблицы, будет выполнен с ошибкой вне зависимости от использования в запросе ключевого слова **РАЗРЕШЕННЫЕ**.

С помощью ключевого слова **РАЗЛИЧНЫЕ** можно указать, что в результат не должны попадать повторяющиеся строки.

Конструкция **ПЕРВЫЕ <Количество>** позволяет задать предельное количество строк в результате запроса. Будут отобраны самые первые (в соответствии с правилами упорядочивания результатов запроса) строки. Количество задается целым числом. В языке запросов добавлена возможность исполнения упорядочивания во вложенных запросах в случае, если вложенный запрос содержит конструкцию **ПЕРВЫЕ**.

В секции **<Список полей выборки>** описываются поля, которые должны содержаться в результате запроса (правила описания полей выборки см. [здесь](#)).

В предложении **ИЗ <Список источников>** указываются источники данных – таблицы информационной базы, содержимое которых обрабатывается в запросе. Описание источников может быть опущено только в том случае, если они полностью определены в списке полей выборки (правила описания источников данных запроса см. [здесь](#)).

Предложение **ГДЕ <Условие отбора>** позволяет отфильтровать результат запроса. В результат попадают только те записи, для которых указанное условие оказывается истинным (правила описания условий отбора см. [здесь](#)).

Предложение **ДЛЯ ИЗМЕНЕНИЯ** предназначено для указания необходимости блокировки считываемых в транзакции данных. Считанные данные становятся недоступными для чтения в других сессиях. Для файлового варианта блокируются указанные таблицы, а для клиент-серверного варианта – только выбранные записи. Блокировка снимается после завершения транзакции.

Предложение **СГРУППИРОВАТЬ** позволяет описать порядок группировки результатов запроса (см. [здесь](#)).

Предложение **ИМЕЮЩИЕ** позволяет накладывать условия на результаты группировки (см. [здесь](#)).

Во всех примерах запросов в данной главе приводится текст запроса и результат запроса. Подразумевается, что текст запроса передается в качестве параметра методу **Выполнить()** объекта **Запрос**.

Приведем пример достаточно простого запроса, состоящего из одного оператора **ВЫБРАТЬ** и списка полей выборки.

Пример:

```
// В отчет необходимо вывести список товарных накладных.
Выбрать
Документ.РасходнаяНакладная.Ссылка
```

[Копировать в буфер обмена](#)

Результат:

Ссылка
Расходная накладная 0000001 от 28.06.2006 14:19:00
Расходная накладная 0000002 от 28.06.2006 14:30:32
Расходная накладная 0000003 от 28.06.2006 14:30:49
Расходная накладная 0000004 от 28.06.2006 14:32:06
Расходная накладная 0000005 от 28.06.2006 14:32:32
Расходная накладная 0000006 от 28.06.2006 14:32:47
Расходная накладная 0000007 от 28.06.2006 14:34:04
Расходная накладная 0000008 от 28.06.2006 14:35:37
Расходная накладная 0000009 от 28.06.2006 14:36:05
Расходная накладная 0000010 от 28.06.2006 14:36:36
Расходная накладная 0000011 от 28.06.2006 14:37:04
Расходная накладная 0000012 от 28.06.2006 14:38:18
Расходная накладная 0000013 от 28.06.2006 14:45:29
Расходная накладная 0000014 от 28.06.2006 14:47:20
Расходная накладная 0000015 от 28.06.2006 14:48:09
Расходная накладная 0000016 от 28.06.2006 14:49:47
Расходная накладная 0000017 от 28.06.2006 14:50:23
Расходная накладная 0000018 от 28.06.2006 14:51:36
Расходная накладная 0000019 от 28.06.2006 14:58:16

Рис. 203. Результат запроса

8.2.6.1. Использование слова «РАЗЛИЧНЫЕ»

Во многих ситуациях желательно, чтобы одинаковые строки в отчете не повторялись.

Пример:

```
// Необходимо узнать, каким вообще контрагентам
// отгружался товар за период.
Выбрать
Документ.РасходнаяНакладная.Контрагент
```

[Копировать в буфер обмена](#)

Результат:

Контрагент
Эльбрус
Эльбрус
Алекс-2002
Большаков Андрей
Завод РТИ
Филипенко
Центр детского творчества
Алекс-2002
Магазин на ул. Алексеева
Никитин Юрий
Магазин на ул. Алексеева
Автохозяйство №34
Алекс-2002
Эльбрус
Русская одежда
Алекс-2002
Турмасов Марат Сергеевич
Завод РТИ
Магазин на ул. Алексеева

Рис. 204. Результат запроса

Видно, что в результате запроса много повторяющихся строк, что снижает его наглядность. Чтобы избежать повторов, в описании запроса следует указать ключевое слово **РАЗЛИЧНЫЕ**.

Пример:

[Копировать в буфер обмена](#)

```
Выбрать Различные
Документ .РасходнаяНакладная .Контрагент
```

Результат:

Контрагент
Эльбрус
Алекс-2002
Центр детского творчества
Магазин на ул. Алексеева
Автохозяйство №34
Завод РТИ
Русская одежда
Турмасов Марат Сергеевич
Большаков Андрей
Филипенко
Никитин Юрий

Рис. 205. Результат запроса

8.2.6.2. Использование слова «ПЕРВЫЕ»

В некоторых случаях необходимо вывести в отчет ограниченное количество строк. Для этого в описании запроса следует указать ключевое слово **ПЕРВЫЕ** и после него – требуемое количество строк.

Пример:

[Копировать в буфер обмена](#)

```
// Необходимо отобразить пять самых дорогих товаров.
// Выборка должна осуществляться в порядке убывания цены товара.
```

```
Выбрать Первые 5
Справочник .Номенклатура .Наименование ,
Справочник .Номенклатура .ЗакупочнаяЦена
Упорядочить По
Справочник .Номенклатура .ЗакупочнаяЦена Убыв
```

Результат:

Наименование	ЗакупочнаяЦена
Копировальный аппарат Omega	3 500,00
Телефон Vega 700	2 500,00
Телефон Vega 300	2 200,00
Сист. блок Hewlett-Packard Vectra VL420	1 699,00
Сист. блок Hewlett-Packard Brio BA410	1 633,00

Рис. 206. Результат запроса

8.2.7. Описание полей выборки

После обязательного ключевого слова **ВЫБРАТЬ** (и уточняющих слов **РАЗЛИЧНЫЕ** и **ПЕРВЫЕ**) в тексте запроса задается список полей выборки. Эти поля будут обрабатываться при выборке данных в запросе. Результат запроса также будет иметь набор полей, определенный в данном списке. Поля выборки описываются по следующим правилам:

<Список полей выборки>

[Копировать в буфер обмена](#)

```
<Поле выборки>[ , <Поле выборки>[ , ...] ] | *
```

<Поле выборки>

[Копировать в буфер обмена](#)

<Описание поля> | <Логическое выражение> [[КАК] <Псевдоним поля>]

<Описание поля>

[Копировать в буфер обмена](#)

<Выражение>[.<Группа полей>]|<Описание пустой таблицы>

<Описание пустой таблицы>

[Копировать в буфер обмена](#)

ПУСТАЯТАБЛИЦА. (<Список псевдонимов>)

<Список псевдонимов>

[Копировать в буфер обмена](#)

[<Псевдоним поля>][,<Список псевдонимов>]

Список полей выборки состоит из одного или нескольких элементов, разделенных запятыми. Каждое поле выборки состоит из описания поля выборки и необязательного псевдонима поля.

Вместо перечисления полей в списке выборки можно указать звездочку (*). Это будет означать, что результат запроса должен содержать все поля, которые есть в исходных таблицах – источниках данных запроса, описанных в списке источников.

ПРИМЕЧАНИЕ 1. При указании звездочки (*) в списке полей выборки в результат не включаются виртуальные поля исходных таблиц.

ПРИМЕЧАНИЕ 2. Получение выборок очень большого размера (более **64 МБ**) требует наличия достаточного количества свободного места на диске, используемом для размещения временных файлов сервера и клиента.

Описание поля определяет, каким образом должны формироваться значения поля. В простейшем случае поле выборки является ссылкой на поле исходной таблицы. Ссылка может задаваться с указанием таблицы, содержащей это поле, или без указания самой таблицы. Разыменование полей см. [здесь](#).

В общем случае поле выборки может представлять собой не только ссылку на поле исходной таблицы, а некоторое выражение (подробнее см. [здесь](#)) или логическое выражение (см. [здесь](#)).

Результаты запроса могут быть сгруппированы (см. [здесь](#)) с помощью агрегатных функций (см. [здесь](#)), указанных в качестве выражений в полях выборки.

Каждому полю выборки может быть назначен псевдоним (см. [здесь](#)). В дальнейшем псевдоним поля может использоваться для более удобного обращения к данному полю.

Группа полей может указываться только тогда, когда поле выборки указывает на вложенную таблицу (см. [здесь](#)). В этом случае можно указать, какие поля должны обрабатываться в выборке по вложенной таблице. Если группа полей не указана, в выборке будут обрабатываться все поля вложенной таблицы.

8.2.7.1. Псевдонимы полей в списке выборки

Если полю выборки назначить псевдоним, то в дальнейшем к этому полю можно будет обращаться, используя его псевдоним, в предложениях **УПОРЯДОЧИТЬ ПО** и **ИТОГИ**, а также при работе с результатом запроса. Такое обращение может быть более удобным и наглядным, а в некоторых случаях единственно возможным.

Ключевое слово **КАК** может предшествовать псевдониму поля. Это слово можно не указывать вообще, но если оно указано, повышается наглядность и удобочитаемость текста запроса.

Псевдонимы полей задаются в соответствии с правилами назначения идентификаторов переменных. Псевдонимы в запросе не могут совпадать.

Назначение псевдонимов полям само по себе никак не влияет на выборку данных в запросе.

Пример:

```
// Необходимо выбрать из справочника товаров
// наименования товаров и наименования групп.
Выбрать
Справочник.Номенклатура.Наименование Как Товар,
Справочник.Номенклатура.Родитель.Наименование Как Группа
Из
Справочник.Номенклатура
```

[Копировать в буфер обмена](#)

Результат:

Товар	Группа
... Установка ПО	Услуги
... Консультации по настройке ОС Windows	Услуги
... Консультации по настройке IC	Услуги
... Ноутбуки	
... Телефоны	
... Копировальные аппараты	
... Копировальный аппарат Omega	Копировальные аппараты
... Ноутбук Rover Computers Navigator KT7	Ноутбуки
... Ноутбук Rover Computers Explorer	Ноутбуки
... Телефон Vega 700	Телефоны
... Телефон Vega 300	Телефоны
... Телефон Siemens SL45	Телефоны
... Телефон I G W7200	Телефоны

Рис. 207. Результат запроса (фрагмент)

Следует обратить внимание, что поля в результате запроса называются **Товар** и **Группа**. Если бы псевдонимы полей не были указаны, поля в результате запроса назывались бы **Наименование** и **Наименование1** (названия полей в результате запроса не могут совпадать, поэтому к названию второго поля автоматически добавлено «1»), что менее наглядно.

8.2.7.2. Вложенные таблицы в списке полей выборки

Поле в списке выборки может ссылаться на вложенную таблицу источника данных запроса. В этом случае поле результата запроса будет иметь тип **РезультатЗапроса**, то есть содержать вложенный результат запроса, сформированный на основе вложенной таблицы – источника.

По умолчанию во вложенный результат включаются все поля вложенной таблицы – источника данных. Имеется возможность явно определить группу полей,

которые должны содержаться во вложенном результате запроса. Группа полей вложенного результата описывается по следующему правилу:

<Группа полей>

[Копировать в буфер обмена](#)

(<Список вложенных полей>) | *

<Список вложенных полей>

[Копировать в буфер обмена](#)

<Вложенное поле>[, <Вложенное поле>[, ...]]

<Вложенное поле>

[Копировать в буфер обмена](#)

<Выражение> [[КАК] <Псевдоним поля>]

Список вложенных полей состоит из одного или нескольких элементов, разделенных запятыми. Если список состоит из одного элемента, его не обязательно заключать в скобки.

Вместо перечисления вложенных полей можно указать звездочку (*); это будет означать, что результат запроса должен содержать все поля, которые есть во вложенной таблице.

Вложенное поле может представлять некоторое выражение (см. [здесь](#)). В простейшем случае выражение – это ссылка на поле вложенной таблицы.

Каждому вложенному полю может быть назначен псевдоним. В дальнейшем псевдоним поля может использоваться для более удобного обращения к данному полю, аналогично псевдонимам полей списка выборки – см. [здесь](#).

Псевдонимы вложенным полям могут быть назначены независимо от того, задан ли псевдоним самой вложенной таблице.

Пример:

[Копировать в буфер обмена](#)

```
// В отчет необходимо вывести спецификацию товарных накладных -
// сам документ, номенклатуру и количество.
```

```
Выбрать
Документ.РасходнаяНакладная.Ссылка,
Документ.РасходнаяНакладная.Состав.(Номенклатура Как Товар,
Количество)
```

Результат:

Ссылка	Состав
Расходная накладная 0000001 от 28.06.2006 14:19:00	ТаблицаЗначений
Расходная накладная 0000002 от 28.06.2006 14:30:32	ТаблицаЗначений
Расходная накладная 0000003 от 28.06.2006 14:30:49	ТаблицаЗначений
Расходная накладная 0000013 от 28.06.2006 14:45:29	ТаблицаЗначений
Расходная накладная 0000014 от 28.06.2006 14:47:20	ТаблицаЗначений
Расходная накладная 0000015 от 28.06.2006 14:48:09	ТаблицаЗначений
Расходная накладная 0000016 от 28.06.2006 14:49:47	ТаблицаЗначений

Товар	Количество
1С:Бухгалтерия 8. Базовая версия	2,00
1С:Бухгалтерия 8.	2,00

Рис. 208. Результат запроса (фрагмент)

Следует обратить внимание, что поле **Состав** результата запроса представляет собой вложенную таблицу, имеющую поля **Номенклатура** и **Количество**.

8.2.7.3. Пустые вложенные таблицы в списке выборки

Если в запросе используется объединение и в некоторых частях объединения присутствуют вложенные таблицы, а в некоторых – нет, возникает необходимость дополнения списка выборки полями – пустыми вложенными таблицами. Делается это при помощи ключевого слова **ПУСТАЯТАБЛИЦА**. После него в скобках указываются псевдонимы полей, из которых будет состоять вложенная таблица.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ Ссылка.Номер, ПУСТАЯТАБЛИЦА.(Ном, Тов, Кол) КАК Состав
ИЗ Документ.РасходнаяНакладная
ОБЪЕДИНИТЬ ВСЕ
ВЫБРАТЬ Ссылка.Номер, Состав.(НомерСтроки, Номенклатура, Количество)
ИЗ Документ.РасходнаяНакладная
```

8.2.8. Описание источников запроса

Задача предложения **ИЗ** состоит в том, чтобы обозначить список исходных таблиц – источников данных, используемых в данном операторе **ВЫБРАТЬ**.

Следует отметить, что предложение **ИЗ** в языке запросов является опциональным. Оно может быть опущено в том случае, если источники данных полностью квалифицированы в описании списка полей выборки, содержащегося в предложении **ВЫБРАТЬ**. Следует обратить внимание, что ряд примеров в предыдущих разделах не содержал предложения **ИЗ**.

После ключевого слова **ИЗ** указывается список источников. В общем случае список источников описывается следующим набором правил:

<Список источников>

[Копировать в буфер обмена](#)

<Источник>[, <Источник>[, ...]]

<Источник>

<Описание источника> [<Перечень соединений>]

<Описание источника>

Копировать в буфер обмена

<Таблица> [[КАК] <Псевдоним источника>]

<Таблица>

Копировать в буфер обмена

<Имя таблицы>[(<Параметры>)] | (<Описание запроса>)

Источники данных запроса перечисляются в списке источников через запятую. Каждый источник в списке источников обязательно включает в себя описание источника; кроме того, может быть указан перечень соединений – правила соединений источника с другими источниками. Описание спецификации соединений см. [здесь](#).

Если в качестве источника данных выступает таблица информационной базы, описание источника содержит имя таблицы.

Если исходная таблица виртуальная, могут быть указаны параметры ее формирования. Подробно параметры виртуальных таблиц описаны в разделе [Встроенный язык – Работа с запросами – Таблицы запросов](#) встроенной справки.

В качестве источника данных запроса может выступать также вложенный запрос. В этом случае описание источника содержит описание запроса. Описание использования вложенных запросов см. [здесь](#).

В описании источника данных может быть также назначен его псевдоним (см. [здесь](#)). В дальнейшем псевдоним источника может использоваться для более удобного обращения к данному источнику.

8.2.8.1. Спецификации соединений

При определении нескольких источников в списке для каждой записи из первой таблицы-источника осуществляется выборка из второй таблицы-источника и т. д. Таким образом, в результате запроса формируются все возможные комбинации всех записей из всех указанных источников.

Пример:

Копировать в буфер обмена

```
ВЫБРАТЬ
Контрагенты.Ссылка как Контрагент,
ТипыЦен.Ссылка КАК ТипЦены
ИЗ
Справочник.Контрагенты КАК Контрагенты,
Справочник.ТипыЦен КАК ТипыЦенРезультат
```

Результат:

Контрагент	ТипЦены
Темп плюс	Оптовая
Темп плюс	Закупочная
Темп плюс	Розничная
Темп плюс	Мелкооптовая
Поставщики	Оптовая
Поставщики	Закупочная
Поставщики	Розничная
Поставщики	Мелкооптовая
Максимус	Оптовая
Максимус	Закупочная
Максимус	Розничная
Максимус	Мелкооптовая
Новация	Оптовая
Новация	Закупочная
Новация	Розничная
Новация	Мелкооптовая

Рис. 209. Результат запроса (фрагмент)

Результат запроса содержит комбинации всех контрагентов со всеми типами цен. Как правило, такой результат сам по себе смысла не имеет. Обычно комбинации записей из разных исходных таблиц требуется ограничить какими-либо условиями. В языке запросов имеется возможность описать такое соединение источников, указывая сами источники и определяя условия, в соответствии с которыми комбинации записей из этих источников требуется включить в результат запроса.

Соединения бывают нескольких видов; они описываются следующими правилами:

<Перечень соединений>

Копировать в буфер обмена

<Соединение> [<Перечень соединений>]

<Соединение>

Копировать в буфер обмена

```
[ВНУТРЕННЕЕ] СОЕДИНЕНИЕ <Описание источника> ПО <Условие отбора> |
ЛЕВОЕ [ВНЕШНЕЕ] СОЕДИНЕНИЕ <Описание источника> ПО <Условие отбора> |
ПРАВОЕ [ВНЕШНЕЕ] СОЕДИНЕНИЕ <Описание источника> ПО <Условие отбора> |
ПОЛНОЕ [ВНЕШНЕЕ] СОЕДИНЕНИЕ <Описание источника> ПО <Условие отбора>
```

В общем случае перечень соединений может содержать и описывать не только одно соединение (двух источников), но и несколько соединений нескольких источников сразу.

Описание источника содержит описание исходной таблицы (см. [здесь](#)).

Условие отбора содержит условия, в соответствии с которыми в выборке необходимо соединить данные из исходных таблиц – источников запроса. Правила описания условий в языке запросов см. [здесь](#).

Ключевые слова **ЛЕВОЕ**, **ПРАВОЕ** и **ПОЛНОЕ** уточняют характер соединения. Слова **ВНУТРЕННЕЕ** или **ВНЕШНЕЕ** можно не указывать вообще, они повышают наглядность и удобочитаемость текста запроса.

Соединяемые источники не равнозначны между собой, и в некоторых случаях результат зависит от того, какая таблица указана первой, до ключевого слова **СОЕДИНЕНИЕ** (слева от него), а какая – второй (справа).

Для описания соединений будем использовать две простые таблицы. Одна таблица называется **Компании**, состоит из двух полей: **Наименование** и **Телефон** и содержит следующие данные:

Наименование	Телефон
Конфетпром	
СервисПог	2-12-85-03

Рис. 210. Таблица «Компании»

Другая таблица называется **Контакты**, состоит из трех полей: **Наименование**, **Телефон** и **Компания** (ссылка на элемент таблицы **Компании**). Таблица содержит следующие данные:

Наименование	Телефон	Компания
Абдулов Юрий Владимирович	222-3-22	
Семенов Геннадий Сергеевич		СервисПог
Шилов Сергей Александрович	222-3-22	Конфетпром

Рис. 211. Таблица «Контакты»

ПРИМЕЧАНИЕ. В терминах «1С:Предприятия» обе таблицы являются справочниками.

8.2.8.1.1. Внутреннее соединение

Внутреннее соединение означает, что из обеих исходных таблиц – источников данных в результате запроса необходимо включить только те комбинации записей, которые соответствуют указанному условию. Остальные записи в результат не попадают.

Пример:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    Компании.Наименование КАК Компания,
    Контакты.Наименование КАК Контакт
ИЗ
    Справочник.Контакты КАК Контакты
ВНУТРЕННЕЕ СОЕДИНЕНИЕ Справочник.Компании КАК Компании
ПО Контакты.Компания = Компании.Ссылка
  
```

Результат:

Компания	Контакт
Конфетпром	Шилов Сергей Александрович
СервисПог	Семенов Геннадий Сергеевич

Рис. 212. Результат запроса

8.2.8.1.2. Левое внешнее соединение

Левое внешнее соединение означает, что в результате запроса нужно включить комбинации записей из обеих исходных таблиц, которые соответствуют указанному условию. Но, в отличие от внутреннего соединения, в результате запроса нужно включить также еще и записи из первого (указанного слева от слова **СОЕДИНЕНИЕ**) источника, для которых не найдено соответствующих условию записей из второго источника.

Таким образом, в результате запроса будут включены все записи из первого источника; они будут соединены с записями из второго источника при выполнении указанного условия. Строки результата запроса, для которых не найдено соответствующих условию записей из второго источника, будут содержать **NULL** в полях, формируемых на основании записей из этого источника.

Пример:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    Контакты.Наименование КАК Контакт,
    Компании.Наименование КАК Компания
ИЗ
    Справочник.Контакты КАК Контакты
ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Компании КАК Компании
ПО Контакты.Компания = Компании.Ссылка
  
```

Результат:

Контакт	Компания
Шилов Сергей Александрович	Конфетпром
Абдулов Юрий Владимирович	NULL
Семенов Геннадий Сергеевич	СервисПог

Рис. 213. Результат запроса

8.2.8.1.3. Правое внешнее соединение

Правое внешнее соединение означает, что в результате запроса нужно включить комбинации записей из обеих исходных таблиц, которые соответствуют указанному условию. Кроме того, в результате запроса нужно также включить и записи из второго (указанного справа от слова **СОЕДИНЕНИЕ**) источника, для которых не найдено соответствующих условию записей из первого источника.

Таким образом, в результате запроса будут включены все записи из второго источника; они будут соединены с записями из первого источника при выполнении указанного условия. Строки результата запроса, для которых не найдено соответствующих условию записей из первого источника, будут содержать **NULL** в полях, формируемых на основании записей из этого источника.

ПРИМЕЧАНИЕ. Фактически правое внешнее соединение можно выразить через левое внешнее соединение. Именно так делает конструктор запросов конфигууратора.

Пример:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    Контакты.Наименование КАК Контакт,
    Компании.Наименование КАК Компания
ИЗ
    Справочник.Контакты КАК Контакты
    ПРАВОЕ СОЕДИНЕНИЕ Справочник.Компании КАК Компании
    ПО Контакты.Телефон = Компании.Телефон

```

Результат:

Контакт	Компания
NULL	СервисПог
Семенов Геннадий Сергеевич	Конфетпром

Рис. 214. Результат запроса

8.2.8.1.4. Полное внешнее соединение

Полное внешнее соединение означает, что в результат запроса нужно включить комбинации записей из обеих исходных таблиц, которые соответствуют указанному условию. Кроме того, в результат запроса нужно также включить и те записи из обоих источников, для которых не найдено соответствий.

Таким образом, в результат запроса будут включены все записи из обоих источников; они будут соединены друг с другом при выполнении указанного условия. Строки результата запроса, для которых не найдено соответствующих условию записей из какого-либо источника, будут содержать `NULL` в полях, формируемых на основании записей из этого источника.

Пример:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    Контакты.Наименование КАК Контакт,
    Компании.Наименование КАК Компания
ИЗ
    Справочник.Компании КАК Компании
    ПОЛНОЕ СОЕДИНЕНИЕ Справочник.Контакты КАК Контакты
    ПО (Контакты.Телефон = Компании.Телефон)

```

Результат:

Контакт	Компания
NULL	СервисПог
Семенов Геннадий Сергеевич	Конфетпром
Шилов Сергей Александрович	NULL
Абдулов Юрий Владимирович	NULL

Рис. 215. Результат запроса

8.2.8.2. Псевдонимы источников данных

Если источнику данных назначить псевдоним, то в дальнейшем к этому источнику можно будет обращаться, используя этот псевдоним (и уже нельзя будет обращаться через указание имени таблицы). Такое обращение может быть более удобным и наглядным, а в некоторых случаях единственно возможным.

Псевдоним задается в соответствии с правилами назначения идентификаторов переменных. Псевдонимы в запросе не могут совпадать.

Ключевое слово `КАК` может предшествовать псевдониму источника. Это слово можно не указывать вообще, но если оно указано, повышается наглядность и удобочитаемость текста запроса.

Назначение псевдонимов источникам само по себе никак не влияет на выборку данных в запросе.

Пример:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    Товар.Наименование,
    Товар.Родитель
ИЗ
    Справочник.Номенклатура КАК Товар

```

Данный пример демонстрирует использование в списке полей выборки псевдонима `Товар`, назначенного исходной таблице `Справочник.Номенклатура`.

8.2.8.3. Вложенные таблицы в списке источников

В списке источников могут фигурировать и вложенные таблицы – табличные части справочников и документов.

Пример:

[Копировать в буфер обмена](#)

```

// В отчет необходимо вывести спецификацию товарных накладных -
// показать сам документ, номенклатуру и количество.
// В списке источников указана вложенная таблица "Состав" -
// табличная часть накладной.
// Выборка ограничена восемью записями, чтобы не перегружать пример.
ВЫБРАТЬ ПЕРВЫЕ 8
    РасходнаяНакладнаяСостав.Ссылка,
    РасходнаяНакладнаяСостав.Номенклатура,
    РасходнаяНакладнаяСостав.Количество
ИЗ
    Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав

```

Результат:

Ссылка	Номенклатура	Количество
Расходная накладная 0000001 от 28.06.2006 14:19:00	1С:Бухгалтерия 8. Базовая версия	2,00
Расходная накладная 0000001 от 28.06.2006 14:19:00	1С:Бухгалтерия 8.	2,00
Расходная накладная 0000002 от 28.06.2006 14:30:32	Монитор 15" LG Studioworks 575N	5,00
Расходная накладная 0000002 от 28.06.2006 14:30:32	Мышь 2-кноп А4Tech PS/2	10,00
Расходная накладная 0000002 от 28.06.2006 14:30:32	Мышь LOGITECH M-S48 PS/2	10,00
Расходная накладная 0000002 от 28.06.2006 14:30:32	Мышь OK-720 Mouse А4Tech PS/2	2,00
Расходная накладная 0000003 от 28.06.2006 14:30:49	1С:Зарплата и Управление Персоналом 8	1,00
Расходная накладная 0000004 от 28.06.2006 14:32:06	Windows XP Home Edition Russian CD	1,00

Рис. 216. Результат запроса

Следует обратить внимание, что при указании вложенной таблицы в списке источников допускается обращение как к полям самой вложенной таблицы, так и к полям таблицы верхнего уровня (той, которая содержит вложенную таблицу) через поле *Ссылка*. В данном случае происходит обращение к полю *Ссылка*. *Наименование* Поля самого документа.

8.2.8.4. Вложенные запросы в списке источников

В списке источников запроса в качестве таблицы-источника может использоваться вложенный запрос. В этом случае описание источника содержит описание вложенного запроса. Описание вложенного запроса составляется точно так же, как и обычного (см. [здесь](#)).

Использование вложенного запроса в качестве источника данных ничем не отличается от использования таблицы информационной базы. В качестве полей такого источника доступны все поля, описанные в списке полей выборки вложенного запроса.

Пример:

```

ВЫБРАТЬ
    ВложенныйЗапрос.Ссылка,
    ВложенныйЗапрос.Номенклатура,
    ВложенныйЗапрос.Количество
ИЗ
    (ВЫБРАТЬ ПЕРВЫЕ 8
        РасходнаяНакладнаяСостав.Ссылка КАК Ссылка,
        РасходнаяНакладнаяСостав.Номенклатура КАК Номенклатура,
        РасходнаяНакладнаяСостав.Количество КАК Количество
    ИЗ
        Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав) КАК ВложенныйЗапрос

```

[Копировать в буфер обмена](#)

Результат будет точно таким же, как и в предыдущем примере.

8.2.9. Фильтрация результатов запроса

Предложение *ГДЕ* <Условие отбора> позволяет задать условие отбора данных из исходных таблиц – источников запроса; в запросе будут обрабатываться только те записи, для которых данное условие оказывается истинным.

Пример:

```

// Необходимо выяснить, какие контрагенты являются частными лицами.
ВЫБРАТЬ
    Контрагенты.Наименование
ИЗ
    Справочник.Контрагенты КАК Контрагенты
ГДЕ
    Контрагенты.Вид = ЗНАЧЕНИЕ(Перечисление.ВидыКонтрагентов.ЧастноеЛицо)

```

[Копировать в буфер обмена](#)

Результат:

Наименование
Эльбрус
Турмасов Марат Сергеевич
Большаков Андрей
Филипенко
Пугачев Максим Олегович
Никитин Юрий

Рис. 217. Результат запроса

ПРИМЕЧАНИЕ. Совершенно не обязательно, чтобы поле, фигурирующее в предложении *ГДЕ*, входило в список выборки.

Условие отбора может определяться и как простое логическое выражение, и как более сложное, в котором простые логические выражения соединяются между собой логическими операторами *и*, *или*, *не*. Описание условий в языке запросов см. [здесь](#).

8.2.10. Группировка результатов запроса

Исходные данные в запросе могут быть сгруппированы с помощью агрегатных функций, указанных в качестве полей в списке выборки. Это означает, что строки в результате запроса будут содержать результаты вычисления указанных агрегатных функций, рассчитанные (сгруппированные) по записям исходных таблиц.

Сами агрегатные функции указываются в списке полей выборки. В предложении *СГРУППИРОВАТЬ ПО* <Поля группировки> необходимо указать список полей, по которым следует произвести группировку. В запросе будут группироваться записи исходных таблиц, содержащие одинаковые значения указанных полей.

Список полей группировки содержит ссылку на поля исходных таблиц – источников запроса, указанные через запятую:

<Поля группировки>

<Разыменование поля> [, <Разыменование поля> [, ...]]

ВНИМАНИЕ! При группировке результатов запроса в списке полей выборки обязательно должны быть указаны агрегатные функции, а помимо них допускается указывать только поля, по которым осуществляется группировка. В запросах, содержащих группировку по какому-либо полю, не следует использовать это поле внутри агрегатных функций. Использование поля одновременно для группировки и внутри агрегатной функции – следует считать ошибкой. Такие запросы могут трактоваться по-разному, в зависимости от того, какая СУБД используется. При нарушении данного правила возможны

ошибки, приводящие к завершению программы.

Исключение составляют ситуации, когда агрегатные функции применены к полям вложенной таблицы. В этом случае в списке полей выборки возможны обращения к полям таблицы верхнего уровня, без группировки результатов по этим полям.

При использовании агрегатных функций предложение **СТРУППИРОВАТЬ ПО** может быть и не указано совсем; при этом все результаты запроса будут сгруппированы в одну-единственную строку.

Пример:

```
// Требуется получить статистику по продаже товаров:
// максимальную, минимальную и среднюю цены в расходных накладных.
Выбрать
    Накладная.Номенклатура ,
    Среднее (Накладная.Цена) Как Среднее ,
    Максимум(Накладная.Цена) Как Максимум ,
    Минимум (Накладная.Цена) Как Минимум
Из
    Документ.РасходнаяНакладная.Состав Как Накладная
Сгруппировать По
    Накладная.Номенклатура
```

[Копировать в буфер обмена](#)

Результат:

Номенклатура	Среднее	Максимум	Минимум
Монитор 15' LG Studioworks 575N	151,666667	155,00	145,00
Монитор 17' Philips 107520	210	210,00	210,00
Монитор 19' Hitachi CM715ET	350	350,00	350,00
Монитор LCD 22' M8537ZM/A	2 000	2 000,00	2 000,00
Сист. блок Hewlett-Packard Brio BA410	2 000	2 000,00	2 000,00
Сист. блок Hewlett-Packard Vectra VL420	2 100	2 100,00	2 100,00
Сист. блок IBM NetVista A22p	1 816,666667	2 500,00	1 350,00
Сист. блок IBM NetVista M41	1 450	1 450,00	1 450,00
Лазерный принтер Canon I RP-R10	300	300,00	300,00

Рис. 218. Результат запроса (фрагмент)

8.2.11. Условия на значения агрегатных функций

Предложение **ИМЕЮЩИЕ <Условие отбора>** позволяет накладывать условия на значения агрегатных функций. В других конструкциях языка запросов, например, в предложении **ГДЕ**, указывать в условиях агрегатные функции нельзя.

Пример:

```
// Необходимо выбрать товары, которых продали более 20 штук.
ВЫБРАТЬ
    Накладная.Номенклатура ,
    СУММА(Накладная.Количество) КАК Количество
ИЗ
    Документ.РасходнаяНакладная.Состав КАК Накладная
СТРУППИРОВАТЬ ПО
    Накладная.Номенклатура
ИМЕЮЩИЕ
    СУММА(Накладная.Количество) > 20
```

[Копировать в буфер обмена](#)

Результат:

Номенклатура	Количество
Лазерный принтер HP LaserJet 2200	26
Мышь 2-кноп А4Tech PS/2	35
Мышь Ice Mouse MUS-2	30
Мышь LOGITECH M-S48 PS/2	41
Мышь GENIUS "EASY" (3 кнопки).	21
Клавиатура Apple Pro Keyboards	23
Клавиатура LK-601 KB-2000 PS/2	23

Рис. 219. Результат запроса

ВНИМАНИЕ! В условии отбора можно использовать только агрегатные функции и поля, по которым осуществляется группировка.

8.2.12. Объединение запросов

В языке запросов имеется возможность объединять несколько запросов. При этом записи, полученные с помощью каждого из объединяемых запросов, будут собраны в один результат запроса.

При объединении каждый запрос собирает данные независимо, а такие операции, как упорядочивание результатов и расчет итогов, выполняются уже над результатом объединения запросов.

Поля результата запроса будут называться так, как описано в списке полей выборки первого из объединяемых запросов. Поля выборки остальных запросов сопоставляются с полями результата в соответствии с порядком их следования в списке полей выборки. Объединяемые запросы должны иметь одинаковое количество полей в списке полей выборки.

Если поля выборки объединяемых запросов имеют разный тип, то поля результата запроса будут иметь составной тип.

Объединение запросов описывается по следующему правилу:

<Объединение запросов>

```
ОБЪЕДИНИТЬ [ВСЕ]
<Описание запроса>
[<Объединение запросов>]
```

[Копировать в буфер обмена](#)

Объединение запросов начинается с обязательного ключевого слова **ОБЪЕДИНИТЬ**, после которого следует описание присоединяемого запроса. Далее может присоединяться еще один запрос и т. д.

По умолчанию при объединении запросов полностью одинаковые строки в результате запроса, сформированные разными запросами, заменяются одной. Если требуется, чтобы были оставлены разные строки, необходимо указать ключевое слово **ВСЕ**.

Пример:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    Приход.Номенклатура КАК Товар,
    СУММА(Приход.Количество) КАК Приход,
    СУММА(0) КАК Расход
ИЗ
    Документ.ПриходнаяНакладная.Состав КАК Приход
СГРУППИРОВАТЬ ПО
    Приход.Номенклатура
ОБЪЕДИНИТЬ
ВЫБРАТЬ
    Расход.Номенклатура,
    СУММА(0),
    СУММА(Расход.Количество)
ИЗ
    Документ.РасходнаяНакладная.Состав КАК Расход
СГРУППИРОВАТЬ ПО
    Расход.Номенклатура
  
```

Результат:

Товар	Приход	Расход
Клавиатура LK-601 KB-2000 PS/2	160	
Windows XP Home Edition Russian CD		3
Windows XP Home Edition Russian CD	25	
Windows XP Home Edition Russian UPG CD		3
Windows XP Home Edition Russian UPG CD	42	
Windows XP Professional Russian CD		6
Windows XP Professional Russian CD	15	
1С.Бухгалтерия 8. Базовая версия		2
1С.Бухгалтерия 8. Базовая версия	16	
1С.Бухгалтерия 8.		2

Рис. 220. Результат запроса (фрагмент)

8.2.13. Упорядочивание результатов запроса

Предложение **УПОРЯДОЧИТЬ ПО** позволяет сортировать строки в результате запроса.

<Упорядочивание результатов>

УПОРЯДОЧИТЬ ПО <Условия упорядочивания>

[Копировать в буфер обмена](#)

<Условия упорядочивания>

<Поле упорядочивания> [<Порядок>]
[, <Поле упорядочивания> [<Порядок>][, ...]]

[Копировать в буфер обмена](#)

<Поле упорядочивания>

<Выражение> <Порядок> ВОЗР | УБЫВ | ИЕРАРХИЯ | ИЕРАРХИЯ УБЫВ

[Копировать в буфер обмена](#)

В предложении **УПОРЯДОЧИТЬ ПО** через запятую перечисляются условия, в соответствии с которыми необходимо упорядочить результат запроса. Выборки упорядочиваются сначала по первому условию, потом по второму и т. д.

Условие упорядочивания в общем случае может представлять собой некоторое выражение (см. [здесь](#)). Строки результата запроса будут упорядочены по значениям этого выражения, рассчитанным для каждой строки.

Упорядочивание может осуществляться в порядке возрастания или убывания значений, а для таблиц, для которых задано свойство иерархичности, – также и по иерархии (см. [здесь](#)). Порядок может задаваться для каждого поля независимо. Описание правил сравнения значений см. [здесь](#).

Поле, фигурирующее в условиях упорядочивания, совсем не обязательно должно попадать в результат запроса.

Пример:

[Копировать в буфер обмена](#)

```

// Требуется отобразить 5 самых дорогих товаров,
// расположив их в порядке убывания цены.
ВЫБРАТЬ ПЕРВЫЕ 5
    Номенклатура.Наименование,
    Номенклатура.ЗакупочнаяЦена
ИЗ
    Справочник.Номенклатура КАК Номенклатура
УПОРЯДОЧИТЬ ПО
    Номенклатура.ЗакупочнаяЦена УБЫВ
  
```

Результат:

Наименование	ЗакупочнаяЦена
Копировальный аппарат Omega	3 500,00
Телефон Vega 700	2 500,00
Телефон Vega 300	2 200,00
Сист. блок Hewlett-Packard Vectra VL420	1 699,00
Сист. блок Hewlett-Packard Bno BA410	1 633,00

Рис. 221. Результат запроса

8.2.13.1. Упорядочивание по иерархии

Для справочников можно назначать упорядочивание по иерархии справочника.

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ
 Справочник.Номенклатура.Наименование,
 Справочник.Номенклатура.ПолноеНаименование
УПОРЯДОЧИТЬ ПО
 Справочник.Номенклатура.Наименование Иерархия

Результат:

Наименование	ПолноеНаименование
<input checked="" type="checkbox"/> Клавиатуры	
<input type="checkbox"/> Копировальные аппараты	
... Копировальный аппарат Omega	Копировальный аппарат Omega
<input type="checkbox"/> Мониторы	
... Монитор 15' LG Studioworks 575N	Монитор 15 LG Studioworks 575N <0.28, 50-160Hz, 128...
... Монитор 17' Philips 107S20	Монитор Philips 107S20 17' (0.27), Flat SM, HC, AGRAS, ...
... Монитор 19' Hitachi CM715ET	Монитор 19' Hitachi CM715ET FST 0.21, 1600x1200@7...
... Монитор LCD 22' M8537ZM/A	Монитор LCD 22' M8537ZM/A Apple Cinema HD Display ...
<input type="checkbox"/> Мышки	

Рис. 222. Результат запроса (фрагмент)

Иерархически сортировать можно только по полю, но не по некоторой операции над ним: поле упорядочивания должно содержать ссылку на поле исходной таблицы – источника данных запроса.

ВНИМАНИЕ! Упорядочивание по иерархии имеет смысл задавать в том случае, если в качестве источника определена именно таблица справочника, а не какая-либо другая таблица, содержащая только ссылку на справочник.

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ
 РасходнаяНакладная.Контрагент.Наименование,
 РасходнаяНакладная.Номер,
 РасходнаяНакладная.Склад,
ИЗ
 Документ.РасходнаяНакладная **КАК** РасходнаяНакладная

В данном примере иерархического упорядочивания не получится, поскольку нет связи со справочником, и группы из этого справочника в результат запроса не попадут.

Для упорядочивания по иерархии необходимо организовать соединение со справочником.

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ
 Контрагенты.Наименование **КАК** Наименование,
 РасходнаяНакладная.Номер,
 РасходнаяНакладная.Склад
ИЗ
 Справочник.Контрагенты **КАК** Контрагенты
ЛЕВОЕ СОЕДИНЕНИЕ Документ.РасходнаяНакладная **КАК** РасходнаяНакладная
ПО Контрагенты.Ссылка = РасходнаяНакладная.Контрагент
УПОРЯДОЧИТЬ ПО
 Наименование **ИЕРАРХИЯ**

Результат:

Наименование	Номер	Склад
<input checked="" type="checkbox"/> Покупатели		
... Автохозяйство №34	0000012	Склад отдела продаж
... Алекс-2002	0000003	Витрина в офисе
... Алекс-2002	0000016	Основной склад
... Алекс-2002	0000008	Основной склад
... Алекс-2002	0000013	Склад отдела продаж
... Завод РТИ	0000005	Витрина в офисе
... Завод РТИ	0000018	Склад отдела продаж
... Магазин на ул. Алексеева	0000019	Склад отдела продаж
... Магазин на ул. Алексеева	0000011	Склад отдела продаж
... Магазин на ул. Алексеева	0000009	Основной склад
... Русская одежда	0000015	Основной склад
... Центр детского творчества	0000007	Основной склад
<input type="checkbox"/> Частные лица		
... Большаков Андрей	0000004	Основной склад
... Никитин Юрий	0000010	Склад отдела продаж

Рис. 223. Результат запроса (фрагмент)

8.2.13.2. Упорядочивание во вложенных таблицах

В предложении **УПОРЯДОЧИТЬ ПО** можно определять также и условия упорядочивания записей из вложенных таблиц; причем их можно комбинировать с условиями упорядочивания по таблице верхнего уровня.

При этом важен порядок указания полей таблицы одного уровня (вложенной или верхнего уровня) относительно друг друга, но не важно, в каком порядке указаны поля таблицы одного уровня относительно полей таблицы другого уровня: упорядочивание выполняется всегда сначала по таблице верхнего уровня, а потом по вложенной таблице.

```
// В отчет необходимо вывести спецификацию товарных накладных -
// показать сам документ, номенклатуру и количество.
// Документы требуется упорядочить по номеру,
// а состав - по наименованию товара.
Выбрать
    Документ.РасходнаяНакладная.Ссылка,
    Документ.РасходнаяНакладная.Состав.(Номенклатура Как Товар, Количество)
Упорядочить По
    Документ.РасходнаяНакладная.Номер,
    Документ.РасходнаяНакладная.Состав.Номенклатура.Наименование
```

8.2.14. Автоупорядочивание результатов

Предложение **АВТОУПОРЯДОЧИВАНИЕ** позволяет включить режим автоматического формирования полей для упорядочивания результата запроса.

Автоупорядочивание работает по следующим принципам:

- Если в запросе было указано предложение **УПОРЯДОЧИТЬ ПО**, то каждая ссылка на таблицу, находящаяся в этом предложении, будет заменена полями, по которым по умолчанию сортируется таблица (для справочников это код или наименование, для документов – дата документа). Если поле для упорядочивания ссылается на иерархический справочник, то будет применена иерархическая сортировка по этому справочнику.
- Если в запросе отсутствует предложение **УПОРЯДОЧИТЬ ПО**, но есть предложение **ИТОГИ**, тогда результат запроса будет упорядочен по полям, присутствующим в предложении **ИТОГИ** после ключевого слова **ПО**, в той же последовательности; и если итоги рассчитывались по полям-ссылкам, то по полям сортировки по умолчанию таблиц, на которые были ссылки.
- Если в запросе отсутствуют предложения **УПОРЯДОЧИТЬ ПО** и **ИТОГИ**, но есть предложение **СГРУППИРОВАТЬ ПО**, тогда результат запроса будет упорядочен по полям, присутствующим в предложении, в той же последовательности; и если группировка велась по полям-ссылкам, то по полям сортировки по умолчанию таблиц, на которые были ссылки.
- Если в запросе отсутствуют предложения **УПОРЯДОЧИТЬ ПО**, **ИТОГИ** и **СГРУППИРОВАТЬ ПО**, результат будет упорядочен по полям сортировки по умолчанию для таблиц, из которых выбираются данные, в порядке их появления в запросе.

8.2.15. Расчет итогов запроса

Предложение **ИТОГИ** позволяет определить, расчет каких итогов необходим в запросе. При расчете итогов вычисляются значения агрегатных функций по выборкам с одинаковыми значениями полей – контрольных точек. Итоги добавляются в результат запроса как итоговые строки.

Порядок расчета итогов запроса описывается в соответствии со следующими правилами:

<Описание итогов>

```
<Итоги> [<Описание итогов>]
```

[Копировать в буфер обмена](#)

<Итоги>

```
ИТОГИ [<Список итоговых полей>] ПО [ОБЩИЕ] [<Список контрольных точек>]
```

[Копировать в буфер обмена](#)

<Список итоговых полей>

```
<Итоговое поле> [, <Список итоговых полей> [, ...]]
```

[Копировать в буфер обмена](#)

<Итоговое поле>

```
<Агрегатная функция> | <Выражение> [[КАК] <Псевдоним поля>]
```

[Копировать в буфер обмена](#)

<Список контрольных точек>

```
<Контрольная точка> [, <Контрольная точка> [, ...]]
```

[Копировать в буфер обмена](#)

<Контрольная точка>

```
<Выражение> [[ТОЛЬКО] ИЕРАРХИЯ] | [ПЕРИОДАМИ(Секунда | Минута | Час | День | Неделя | Месяц | Квартал | Год | Декада | Полугодие)
[, <Литерал типа DATE> | <Идентификатор параметра>]
[, <Литерал типа DATE> | <Идентификатор параметра>]] [[КАК] Псевдоним поля]
```

[Копировать в буфер обмена](#)

Описание итогов начинается с обязательного ключевого слова **ИТОГИ**.

Список агрегатных функций содержит перечень агрегатных функций (см. [здесь](#)), которые необходимо рассчитывать в итогах.

Ключевое слово **ОБЩИЕ** означает, что необходимо сформировать итоговую строку по всему результату запроса. Описание расчета общих итогов см. [здесь](#).

Помимо общих итогов можно задать расчет итогов по контрольным точкам. Для этого после обязательного ключевого слова **ПО** необходимо указать **<Список контрольных точек>**. Каждая контрольная точка содержит выражение, вычисляемое при выполнении запроса. По каждой комбинации значений этих выражений будут рассчитаны и добавлены в результат запроса итоговые строки.

Если контрольная точка является ссылкой на справочник, возможен расчет итогов по иерархии справочника (см. [здесь](#)). Для этого после такой ссылки нужно указать обязательное ключевое слово **ИЕРАРХИЯ**.

8.2.15.1. Расчет итогов во вложенных таблицах

В настоящей версии программы не поддерживается расчет итогов по вложенным таблицам.

8.2.15.2. Итоги по иерархии

Есть возможность рассчитать итоги по иерархии. Для этого после имени поля, для которого вычисляются итоги, необходимо указать ключевое слово **ИЕРАРХИЯ**. В результате будут рассчитаны итоги по контрольным точкам и итоги по иерархии для контрольных точек.

Пример:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    Документ.Номенклатура КАК Номенклатура ,
    Документ.Количество КАК Количество ,
    Документ.Ссылка.Номер ,
    Документ.Ссылка.Контрагент
ИЗ
    Документ.РасходнаяНакладная.Состав КАК Документ
УПОРЯДОЧИТЬ ПО
    Документ.Номенклатура
ИТОГИ
    СУММА(Количество)
ПО
    Номенклатура ИЕРАРХИЯ

```

Результат:

Номенклатура	Количество	Номер	Контрагент
[-] Системные блоки и комплектующие	24		
[-] Сис. блок Hewlett-Packard Brio BA410	9		
[-] Сис. блок Hewlett-Packard Vectra VL420	7		
Сис. блок Hewlett-Packard Vectra VL420	1	0000010	Никитин Юрий
Сис. блок Hewlett-Packard Vectra VL420	3	0000008	Алекс-2002
Сис. блок Hewlett-Packard Vectra VL420	1	0000007	Центр детского творчества
Сис. блок Hewlett-Packard Vectra VL420	1	0000018	Завод РТИ
Сис. блок Hewlett-Packard Vectra VL420	1	0000016	Алекс-2002
[-] Сис. блок IBM NetVista A22p	4		
Сис. блок IBM NetVista A22p	2	0000012	Автохозяйство №34
Сис. блок IBM NetVista A22p	1	0000007	Центр детского творчества
Сис. блок IBM NetVista A22p	1	0000014	Эльбрус
[-] Сис. блок IBM NetVista M41	4		

Рис. 224. Результат запроса (фрагмент)

При необходимости можно рассчитать только итоги значений по иерархии, без расчета итогов в контрольных точках. Для этого перед ключевым словом ИЕРАРХИЯ нужно указать ключевое слово ТОЛЬКО.

Пример:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    Документ.Номенклатура КАК Номенклатура ,
    Документ.Количество КАК Количество ,
    Документ.Ссылка.Номер ,
    Документ.Ссылка.Контрагент
ИЗ
    Документ.РасходнаяНакладная.Состав КАК Документ
УПОРЯДОЧИТЬ ПО
    Документ.Номенклатура
ИТОГИ
    СУММА(Количество)
ПО
    Номенклатура ТОЛЬКО ИЕРАРХИЯ

```

Результат:

Номенклатура	Количе...	Номер	Контрагент
Сис. блок IBM NetVista A22p	2	0000012	Автохозяйство №34
Сис. блок IBM NetVista A22p	1	0000007	Центр детского творчества
Сис. блок IBM NetVista A22p	1	0000014	Эльбрус
Сис. блок IBM NetVista M41	1	0000012	Автохозяйство №34
Сис. блок IBM NetVista M41	1	0000007	Центр детского творчества
Сис. блок IBM NetVista M41	1	0000008	Алекс-2002
Сис. блок IBM NetVista M41	1	0000016	Алекс-2002
[-] Мониторы	38		
Монитор 15" LG Studioworks 575N	1	0000014	Эльбрус
Монитор 15" LG Studioworks 575N	1	0000016	Алекс-2002
Монитор 15" LG Studioworks 575N	1	0000004	Большаков Андрей
Монитор 15" LG Studioworks 575N	3	0000018	Завод РТИ
Монитор 15" LG Studioworks 575N	5	0000011	Магазин на ул. Алексеева

Рис. 225. Результат запроса (фрагмент)

8.2.15.3. Дополнение дат

В случае если поле, по которому рассчитываем итоги, является полем типа Дата, возможно дополнение результатов датами в заданном периоде. Делается это при помощи ключевого слова ПЕРИОДАМИ, после которого в скобках указывается вид периода (Секунда, Минута, Час, День, Неделя, Месяц, Квартал, Год, Декада, Полугодие), начальная и конечная даты интересующего периода. Если начальная и конечная даты не указаны, будут использованы первая и последняя даты, участвующие в результате.

Пример:

[Копировать в буфер обмена](#)

```

// Получить количество покупок по клиентам по часам выбранного дня
ВЫБРАТЬ
    ПриходнаяНакладная.Контрагент ,
    НАЧАЛОПЕРИОДА(ПриходнаяНакладная.Дата, ЧАС) КАК Период ,
    КОЛИЧЕСТВО(ПриходнаяНакладная.Ссылка) КАК КоличествоПокупок
ИЗ
    Документ.ПриходнаяНакладная КАК ПриходнаяНакладная
СГРУППИРОВАТЬ ПО
    ПриходнаяНакладная.Контрагент ,
    НАЧАЛОПЕРИОДА(ПриходнаяНакладная.Дата, ЧАС)
ИТОГИ
    СУММА(КоличествоПокупок)
ПО

```

Результат:

Контрагент	Период	Количество	Покупок
	28.06.2006 12:00:00		11
Темп плюс	28.06.2006 12:00:00		4
Максимус	28.06.2006 12:00:00		3
Новация	28.06.2006 12:00:00		2
Николаев	28.06.2006 12:00:00		2
	28.06.2006 14:00:00		2
Темп плюс	28.06.2006 14:00:00		1
Максимус	28.06.2006 14:00:00		1

Рис. 226. Результат запроса

Такое представление результата получится, только если при обходе результата по группировке **Период** использовать в качестве источника измерения все записи периода.

8.2.15.4. Расчет общих итогов

Для расчета итогов по всей таблице в предложении **ИТОГИ** следует указать слово **ОБЩИЕ**. В этом случае будут вычислены значения агрегатных функций для всех записей таблицы.

Пример:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    Документ.Номенклатура,
    Документ.Количество КАК Количество,
    Документ.Ссылка.Номер,
    Документ.Ссылка.Контрагент
ИЗ
    Документ.РасходнаяНакладная.Состав КАК Документ
ИТОГИ
    СУММА(Количество)
ПО
    ОБЩИЕ

```

Результат:

Номенклатура	Количество	Номер	Контрагент
		332	
1С:Бухгалтерия 8. Базовая версия		2 0000001	Эльбрус
1С:Бухгалтерия 8.		2 0000001	Эльбрус
Монитор 15' LG Studioworks 575N		5 0000002	Эльбрус
Мышь 2-кноп А4Tech PS/2		10 0000002	Эльбрус
Мышь LOGITECH M-S48 PS/2		10 0000002	Эльбрус
Мышь OK-720 Mouse А4Tech PS/2		2 0000002	Эльбрус
1С:Зарплата и Управление Персоналом 8		1 0000003	Алекс-2002
Windows XP Home Edition Russian CD		1 0000004	Большаков Андрей
Клавиатура Apple Pro Keyboards		1 0000004	Большаков Андрей

Рис. 227. Результат запроса (фрагмент)

8.2.15.5. Совместное использование итогов и группировки

Если итоги используются совместно с группировкой и для итогов не указан список агрегатных функций, он будет автоматически формироваться из агрегатных полей списка выборки. Если запрос содержит объединение, агрегатные функции будут браться из первого запроса.

Пример:

[Копировать в буфер обмена](#)

```

ВЫБРАТЬ
    Документ.Номенклатура КАК Номенклатура,
    Документ.Ссылка.Контрагент КАК Контрагент,
    СУММА(Документ.Количество) КАК Количество
ИЗ
    Документ.РасходнаяНакладная.Состав КАК Документ
СГРУППИРОВАТЬ ПО
    Документ.Номенклатура,
    Документ.Ссылка.Контрагент
ИТОГИ ПО
    Номенклатура,
    Контрагент

```

Результат:

Номенклатура	Контрагент	Количество
Монитор 15' LG Studioworks 575N		16
Монитор 15' LG Studioworks 575N	Эльбрус	6
Монитор 15' LG Studioworks 575N	Эльбрус	6
Монитор 15' LG Studioworks 575N	Алекс-2002	1
Монитор 15' LG Studioworks 575N	Алекс-2002	1
Монитор 15' LG Studioworks 575N	Магазин на ул. Алексеева	5
Монитор 15' LG Studioworks 575N	Магазин на ул. Алексеева	5
Монитор 15' LG Studioworks 575N	Завод РТИ	3
Монитор 15' LG Studioworks 575N	Завод РТИ	3
Монитор 15' LG Studioworks 575N	Большаков Андрей	1
Монитор 15' LG Studioworks 575N	Большаков Андрей	1
Монитор 17 Philips 107S20		4
Монитор 17 Philips 107S20	Алекс-2002	2

Рис. 228. Результат запроса (фрагмент)

8.2.15.6. Псевдонимы итогов

Полям итогов – контрольным точкам, для которых считаются итоги, можно назначить псевдоним, для последующего обращения к ним из встроенного языка предприятия. Для этого после выражения – контрольной точки необходимо указать имя псевдонима аналогично тому, как это делается в списке полей выборки.

Пример:

```

ВЫБРАТЬ
    Документ.Номенклатура КАК Номенклатура ,
    Документ.Количество КАК Количество ,
    Документ.Ссылка.Номер ,
    Документ.Ссылка.Контрагент
ИЗ
    Документ.РасходнаяНакладная.Состав КАК Документ
УПОРЯДОЧИТЬ ПО
    Документ.Номенклатура
ИТОГИ
    СУММА(Количество)
ПО
    Номенклатура ТОЛЬКО ИЕРАРХИЯ КАК Товары

```

[Копировать в буфер обмена](#)

В случае, если псевдоним не указан, система сама даст имя итогу так, чтобы оно было уникально. В приведенном выше примере итог будет иметь имя **Товары**.

8.2.16. Выражения в языке запросов

Во многих конструкциях языка запросов могут использоваться выражения. Выражения языка запросов описываются следующим набором правил:

<Выражение>

```

<Разыменование поля> |
<Агрегатная функция> |
<Встроенная функция> |
<Операция выбора> |
<Приведение типа>[.<Разыменование поля>] |
<Значение> |
<Выражение> <Бинарная операция> <Выражение> |
<Унарная операция> <Выражение> |
( <Выражение> )

```

[Копировать в буфер обмена](#)

<Бинарная операция>

```
+ | - | * | /
```

[Копировать в буфер обмена](#)

<Унарная операция>

```
- | +
```

[Копировать в буфер обмена](#)

В простейшем случае выражение является ссылкой на поле исходной таблицы – источника данных запроса. Ссылка может задаваться с указанием таблицы, содержащей это поле, или без указания самой таблицы. Описание разыменования полей см. [здесь](#).

Выражения в списке полей выборки, в предложениях **ИМЕЮЩИЕ**, **ИТОГИ**, **УПОРЯДОЧИТЬ ПО**, могут быть агрегатными функциями (см. [здесь](#)).

Выражение может быть встроенной функцией языка запросов (см. [здесь](#)). Могут использоваться операции выбора (см. [здесь](#)) и операции приведения типа значения (см. [здесь](#)).

В выражениях могут непосредственно указываться значения логических, числовых, строковых и др. констант; также могут использоваться значения параметров запроса (см. [здесь](#)). В выражениях к значениям соответствующих типов могут применяться бинарные и унарные операции. При использовании операции деления рекомендуется делать проверку делителя на отличие от нуля.

8.2.16.1. Разыменование полей

Выражения языка запросов в простейшем случае представляют собой ссылки на поля таблиц информационной базы. В общем виде ссылки описываются следующими правилами:

<Разыменование поля>

```
[<Таблица>.<Имя поля>[.<Имя поля>[...]]]
```

[Копировать в буфер обмена](#)

<Таблица>

```
<Имя таблицы> | <Псевдоним источника>
```

[Копировать в буфер обмена](#)

Разыменование поля начинается с имени таблицы, содержащей это поле. Если имя поля уникально – существует только у одной из таблиц среди указанных в списке источников, таблица может быть опущена.

Если поле имеет ссылочный тип, язык запросов позволяет обращаться к полям таблицы, на которую ссылается поле, и так далее. Имена полей указываются через точку.

Если исходной таблице в списке источников присвоен псевдоним источника, он может использоваться вместо имени таблицы в разыменовании полей этой таблицы. В противном случае указывается имя таблицы (см. [здесь](#)).

8.2.16.2. Агрегатные функции языка запросов

В языке запросов предусмотрены агрегатные функции, которые используются при группировке результатов запроса и при подсчете итогов. Агрегатные функции предназначены для обобщения значений указанного параметра. Определены следующие агрегатные функции:

<Агрегатная функция>

[Копировать в буфер обмена](#)

```
СУММА ( <Выражение> ) |
СРЕДНЕЕ ( <Выражение> ) |
МИНИМУМ ( <Выражение> ) |
МАКСИМУМ ( <Выражение> ) |
КОЛИЧЕСТВО ( [ РАЗЛИЧНЫЕ ] <Выражение> | * )
```

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    Накладная.Номенклатура.Наименование,
    СУММА(Накладная.Сумма) КАК Сумма,
    СРЕДНЕЕ(Накладная.Сумма) КАК Среднее,
    МАКСИМУМ(Накладная.Сумма) КАК Максимум,
    МИНИМУМ(Накладная.Сумма) КАК Минимум,
    КОЛИЧЕСТВО(Накладная.Сумма) КАК Колич
ИЗ
    Документ.РасходнаяНакладная.Состав КАК Накладная
СГРУППИРОВАТЬ ПО
    Накладная.Номенклатура,
    Накладная.Номенклатура.Наименование
ИТОГИ ПО
    ОБЩИЕ
```

Результат:

НоменклатураНаименование	Сумма	Среднее	Максимум	Минимум	Колич
	104 086,7	873,606641178571...	16 600,00	1,10	28
Клавиатура Apple Pro Keyboards	1 690	241,428571	375,00	75,00	7
Монитор 17 Philips 107S20	840	210	210,00	210,00	4
1С:Бухгалтерия 8.	280	280	280,00	280,00	1
Сист. блок IBM NetVista M41	5 800	1 450	1 450,00	1 450,00	4
Windows XP Home Edition Russian UPG CD	255	127,5	170,00	85,00	2
Монитор 15 LG Studioworks 575N	2 400	400	750,00	155,00	6
Лазерный принтер Canon LBP-810	2 100	350	600,00	300,00	6
Сист. блок Hewlett-Packard Bro BA410	18 000	3 000	6 000,00	2 000,00	6
Мышь Ice Mouse MUS-2	39,8	5,685714	13,00	1,40	7

Рис. 229. Результат запроса (фрагмент)

Агрегатные функции могут использоваться в списке полей выборки, в предложениях **ИМЕЮЩИЕ**, **ИТОГИ**, **УПОРЯДОЧИТЬ ПО**. В списке полей выборки агрегатные функции следует использовать только для получения результата. Не рекомендуется использовать агрегатные функции в списке полей выборки для сравнения.

Корректный запрос:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    КОЛИЧЕСТВО(*),
    Поставщик
ИЗ
    Документ.РасходнаяНакладная КАК Накладная
СГРУППИРОВАТЬ ПО
    Поставщик
ИМЕЮЩИЕ
    КОЛИЧЕСТВО(*) > 1
```

Некорректный запрос:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    КОГДА КОЛИЧЕСТВО(*) > 1 ТОГДА Истина ИНАЧЕ Ложь КОНЕЦ,
    Поставщик
ИЗ
    Документ.РасходнаяНакладная КАК Накладная
СГРУППИРОВАТЬ ПО
    Поставщик
```

Несмотря на то, что некорректный запрос может работать успешно, такие конструкции (участие агрегатной функции не для получения результата, а для проверки условия) в разделе **ВЫБРАТЬ** имеют свою специфику на разных СУБД и могут приводить к ошибкам.

СУММА

Описание:

Функция вычисляет арифметическую сумму всех попавших в выборку значений поля.

В качестве параметра функции можно указывать только поля, содержащие числовое значение.

Если поле не может содержать числовых значений, то применение функции **СУММА** к такому полю вызовет ошибку. Данная функция может быть применена к такому полю, только если поле может содержать числовые значения (имеет составной тип данных). Но если среди значений поля в выборке встретится нечисловое значение (помимо значений **NULL**), это вызовет ошибку.

СРЕДНЕЕ

Описание:

Функция вычисляет среднее значение всех попавших в выборку значений поля.

В качестве параметра функции можно указывать только ссылки на поля, содержащие числовое значение.

Если поле не может содержать числовых значений, то применение функции **СРЕДНЕЕ** к такому полю вызовет ошибку. Данная функция может быть применена к такому полю в том случае, если поле может содержать числовые значения (имеет составной тип данных). Но если среди значений поля в выборке встретится нечисловое значение (помимо значений **NULL**), это вызовет ошибку.

МИНИМУМ

Описание:

Функция вычисляет минимальное значение из всех попавших в выборку значений поля.

В качестве параметра функции можно указывать ссылки на поля, содержащие значения любого типа.

При определении минимального значения применяются правила сравнения значений, описанные в разделе «Правила сравнения значений».

МАКСИМУМ

Описание:

Функция вычисляет максимальное значение из всех попавших в выборку значений поля.

В качестве параметра функции можно указывать выражения, содержащие значения любого типа. Не может применяться к выражениям с типом [ХранилищеЗначения](#), [МоментВремени](#) и строкам неограниченной длины.

При определении максимального значения применяются правила сравнения значений, описанные в разделе «Правила сравнения значений».

КОЛИЧЕСТВО

Описание:

Функция подсчитывает количество значений параметра, попавших в выборку. В отличие от других агрегатных функций, функция [КОЛИЧЕСТВО](#) допускает три способа использования:

- Функция позволяет подсчитать количество значений указанного поля, не равных [NULL](#).
- Функция позволяет подсчитать количество различных значений указанного поля, не равных [NULL](#). Для этого перед спецификацией поля нужно указать ключевое слово [РАЗЛИЧНЫЕ](#).
- Функция позволяет подсчитать количество строк в результате запроса. Для этого в качестве параметра функции нужно указать звездочку «*».

В качестве параметра функции можно указывать ссылки на поля, содержащие значения любого типа.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    КОЛИЧЕСТВО(*) КАК Всего,
    КОЛИЧЕСТВО(РАЗЛИЧНЫЕ Накладная.Номенклатура) КАК Разные
ИЗ
    Документ.РасходнаяНакладная.Состав КАК Накладная
```

Результат:

Всего	Разные
120	28

Рис. 230. Результат запроса

8.2.16.3. Встроенные функции языка запросов

В языке запросов определены встроенные функции, которые могут использоваться в выражениях в списке полей выборки (см. [здесь](#)) и в условии отбора в предложении [ГДЕ](#) (см. [здесь](#)).

Определены следующие встроенные функции:

<Встроенная функция>

[Копировать в буфер обмена](#)

```
ПОДСТРОКА ( <Выражение>, <Значение>, <Значение> ) |
ГОД ( <Выражение> ) |
КВАРТАЛ ( <Выражение> ) |
МЕСЯЦ ( <Выражение> ) |
ДЕНЬГОДА ( <Выражение> ) |
ДЕНЬ ( <Выражение> ) |
НЕДЕЛЯ ( <Выражение> ) |
ДЕНЬНЕДЕЛИ ( <Выражение> ) |
ЧАС ( <Выражение> ) |
МИНУТА ( <Выражение> ) |
СЕКУНДА ( <Выражение> ) |
НАЧАЛОПЕРИОДА ( <Выражение>, Минута | Час | День | Неделя | Месяц | Квартал | Год | Декада | Полугодие ) |
КОНЕЦПЕРИОДА ( <Выражение>, Минута | Час | День | Неделя | Месяц | Квартал | Год | Декада | Полугодие ) |
ДОБАВИТЬКДАТЕ ( <Выражение>, Минута | Час | День | Неделя | Месяц | Квартал | Год | Декада | Полугодие, <Выражение> ) |
РАЗНОСТЬДАТ ( <Выражение>, <Выражение>, Секунда | Минута | Час | День | Месяц | Квартал | Год ) |
ТИПЗНАЧЕНИЯ ( <Выражение> ) |
ПРЕДСТАВЛЕНИЕ ( <Выражение> ) |
ЕСТЬNULL ( <Выражение>, <Выражение> )
```

ПОДСТРОКА

Описание:

Данная функция предназначена для выделения подстроки из строки. В функцию передаются три параметра:

- Строка, из которой необходимо выделить подстроку. Выражение, имеющее тип [Строка](#).
- Позиция символа, с которого начинается выделяемая из строки подстрока. Значение типа [Число](#).
- Длина выделяемой подстроки. Значение типа [Число](#).

Если в качестве первого параметра фигурирует строка, то результатом функции будет строка (возможно, нулевой длины). Если в качестве первого параметра будет использовано значение [NULL](#), то результатом функции также будет значение [NULL](#). Другие значения считаются недопустимыми и вызывают

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ ПЕРВЫЕ 8
Контрагенты.Наименование,
ПОДСТРОКА(Контрагенты.Наименование, 3, 5) КАК Подстрока
ИЗ
Справочник.Контрагенты КАК Контрагенты
```

Результат:

Наименование	Подстрока
Тепл плюс	мп пл
Поставщики	ставщ
Максимус	ксиму
Новация	вация
Николаев	колае
Приборпоставка	иборп
Покупатели	купат
Эльбрус	ьбрус

Рис. 231. Результат запроса

ГОД

Описание:

Данная функция предназначена для вычисления номера года из значения типа *Дата*.

Параметр функции – это выражение, имеющее тип *Дата*.

Если в качестве параметра фигурирует значение типа *Дата*, то результатом функции будет значение типа *Число*. Если в качестве параметра будет использовано значение *NULL*, то результатом функции также будет значение *NULL*. Другие значения считаются недопустимыми и вызывают состояние ошибки.

КВАРТАЛ

Описание:

Данная функция предназначена для вычисления номера квартала из значения типа *Дата*. Номер квартала находится в диапазоне 1–4.

Параметр функции – это выражение, имеющее тип *Дата*.

Если в качестве параметра фигурирует значение типа *Дата*, то результатом функции будет значение типа *Число*. Если в качестве параметра будет использовано значение *NULL*, то результатом функции также будет значение *NULL*. Другие значения считаются недопустимыми и вызывают состояние ошибки.

МЕСЯЦ

Описание:

Данная функция предназначена для вычисления номера месяца из значения типа *Дата*. Номер месяца находится в диапазоне 1–12.

Параметр функции – это выражение, имеющее тип *Дата*.

Если в качестве параметра фигурирует значение типа *Дата*, то результатом функции будет значение типа *Число*. Если в качестве параметра будет использовано значение *NULL*, то результатом функции также будет значение *NULL*. Другие значения считаются недопустимыми и вызывают состояние ошибки.

ДЕНЬГОДА

Описание:

Данная функция предназначена для вычисления дня года из значения типа *Дата*. День года находится в диапазоне 1–366.

Параметр функции – это выражение, имеющее тип *Дата*.

Если в качестве параметра фигурирует значение типа *Дата*, то результатом функции будет значение типа *Число*. Если в качестве параметра будет использовано значение *NULL*, то результатом функции также будет значение *NULL*. Другие значения считаются недопустимыми и вызывают состояние ошибки.

ДЕНЬ

Описание:

Данная функция предназначена для вычисления дня месяца из значения типа *Дата*. День месяца находится в диапазоне 1–31.

Параметр функции – это выражение, имеющее тип *Дата*.

Если в качестве параметра фигурирует значение типа *Дата*, то результатом функции будет значение типа *Число*. Если в качестве параметра будет использовано значение *NULL*, то результатом функции также будет значение *NULL*. Другие значения считаются недопустимыми и вызывают состояние ошибки.

НЕДЕЛЯ

Описание:

Данная функция предназначена для вычисления номера недели года из значения типа *Дата*.

Параметр функции – это выражение, имеющее тип *Дата*.

Если в качестве параметра фигурирует значение типа *Дата*, то результатом функции будет значение типа *Число*. Если в качестве параметра будет использовано значение *NULL*, то результатом функции также будет значение *NULL*. Другие значения считаются недопустимыми и вызывают состояние ошибки.

ДЕНЬНЕДЕЛИ

Описание:

Данная функция предназначена для вычисления дня недели из значения типа *Дата*. День недели находится в диапазоне 1 (понедельник) – 7 (воскресенье).

Параметр функции – это выражение, имеющее тип *Дата*.

Если в качестве параметра фигурирует значение типа *Дата*, то результатом функции будет значение типа *Число*. Если в качестве параметра будет использовано значение *NULL*, то результатом функции также будет значение *NULL*. Другие значения считаются недопустимыми и вызывают состояние ошибки.

ЧАС

Описание:

Данная функция предназначена для вычисления часа суток из значения типа *Дата*. Час суток находится в диапазоне 0–23.

Параметр функции – это выражение, имеющее тип *Дата*.

Если в качестве параметра фигурирует значение типа *Дата*, то результатом функции будет значение типа *Число*. Если в качестве параметра будет использовано значение *NULL*, то результатом функции также будет значение *NULL*. Другие значения считаются недопустимыми и вызывают состояние ошибки.

МИНУТА

Описание:

Данная функция предназначена для вычисления минуты часа из значения типа *Дата*. Минута часа находится в диапазоне 0–59.

Параметр функции – это выражение, имеющее тип *Дата*.

Если в качестве параметра фигурирует значение типа *Дата*, то результатом функции будет значение типа *Число*. Если в качестве параметра будет использовано значение *NULL*, то результатом функции также будет значение *NULL*. Другие значения считаются недопустимыми и вызывают состояние ошибки.

СЕКUNДА

Описание:

Данная функция предназначена для вычисления секунды минуты из значения типа *Дата*. Секунда минуты находится в диапазоне 0–59.

Параметр функции – это выражение, имеющее тип *Дата*.

Если в качестве параметра фигурирует значение типа *Дата*, то результатом функции будет значение типа *Число*. Если в качестве параметра будет использовано значение *NULL*, то результатом функции также будет значение *NULL*. Другие значения считаются недопустимыми и вызывают состояние ошибки.

НАЧАЛОПЕРИОДА

Описание:

Функция предназначена для выделения определенной даты из заданной даты.

Параметры функции – это выражение, имеющее тип *Дата* и тип периода – одно из значений: *Минута*, *Час*, *День*, *Неделя*, *Месяц*, *Квартал*, *Год*, *Декада*, *Полугодие*.

Пример:

`ВЫБРАТЬ НАЧАЛОПЕРИОДА(ДАТАВРЕМЯ (2002, 10, 12, 10, 15, 34), МЕСЯЦ)`

[Копировать в буфер обмена](#)

Результат:

Поле1
01.10.2002

Рис. 232. Результат запроса

Пример:

`ВЫБРАТЬ НАЧАЛОПЕРИОДА(ДАТАВРЕМЯ(2002, 10, 12, 10, 15, 34), ДЕНЬ)`

[Копировать в буфер обмена](#)

Результат:

Поле1
12.10.2002

Рис. 233. Результат запроса

КОНЕЦПЕРИОДА

Функция предназначена для выделения определенной даты из заданной даты.

Параметры функции – это выражение, имеющее тип *Дата* и тип периода – одно из значений: *Минута*, *Час*, *День*, *Неделя*, *Месяц*, *Квартал*, *Год*, *Декада*, *Полугодие*.

Пример:

`ВЫБРАТЬ КОНЕЦПЕРИОДА(ДАТАВРЕМЯ(2002, 10, 12, 10, 15, 34), МЕСЯЦ)`

[Копировать в буфер обмена](#)

Результат:

Поле1
31.10.2002 23:59:59

Рис. 234. Результат запроса

Пример:

`ВЫБРАТЬ КОНЕЦПЕРИОДА(ДАТАВРЕМЯ(2002, 10, 12, 10, 15, 34), ГОД)`

[Копировать в буфер обмена](#)

Результат:

Поле1
31.12.2002 23:59:59

Рис. 235. Результат запроса

ДОБАВИТЬКДАТЕ

Описание:

Функция предназначена для прибавления к дате некоторой величины.

Первый параметр – исходная дата, к значению которой требуется добавить заданную величину, определяемую вторым и третьим параметрами; выражение, имеющее тип *Дата*.

Второй параметр – тип увеличения, одно из значений: *Секунда*, *Минута*, *Час*, *День*, *Неделя*, *Месяц*, *Квартал*, *Год*, *Декада*, *Полугодие*.

Третий параметр – величина, на которую требуется увеличить дату, задаваемую первым параметром; тип *Число* (дробная часть игнорируется).

Пример:

`ВЫБРАТЬ ДОБАВИТЬКДАТЕ(ДАТАВРЕМЯ(2002, 10, 12, 10, 15, 34), МЕСЯЦ, 1)`

[Копировать в буфер обмена](#)

Результат:

Поле1
12.11.2002 10:15:34

Рис. 236. Результат запроса

Пример:

`ВЫБРАТЬ ДобавитьКДате(ДатаВремя(2002, 10, 12, 10, 15, 34), День, 5)`

[Копировать в буфер обмена](#)

Результат:

Поле1
17.10.2002 10:15:34

Рис. 237. Результат запроса

РАЗНОСТЬДАТ

Описание:

Функция предназначена для получения разницы между двумя датами.

Первый параметр – выражение типа *Дата*. Второй параметр – выражение типа *Дата*. Третий параметр – тип разности, одно из значений: *Секунда*, *Минута*, *Час*, *День*, *Месяц*, *Квартал*, *Год*.

Не поддерживается использование данной функции при задании значения параметра виртуальной таблицы *Период*.

Пример:

`ВЫБРАТЬ РАЗНОСТЬДАТ(ДАТАВРЕМЯ(2002, 10, 12, 10, 15, 34), ДАТАВРЕМЯ(2002, 10, 14, 9, 18, 06), ДЕНЬ)`

[Копировать в буфер обмена](#)

Результат:

Поле1
2

Рис. 238. Результат запроса

Пример:

ВЫБРАТЬ РАЗНОСТЬДАТ(ДатаВремя(2002, 10, 12), ДатаВремя(2002, 11, 03), МЕСЯЦ)

[Копировать в буфер обмена](#)

Результат:

Поле1
1

Рис. 239. Результат запроса

ВНИМАНИЕ! Функция рассчитывает календарную разницу между двумя датами, поэтому ее нельзя использовать там, где необходимо рассчитать количество банковских или рабочих дней между двумя датами.

Функция ТИПЗНАЧЕНИЯ

Описание:

Функция определения типа значения в запросе.

Параметры:

Параметры функции – выражение любого типа.

Возвращаемое значение – значение типа **Тип**.

Пример:

ВЫБРАТЬ
ТИПЗНАЧЕНИЯ(УчетНоменклатуры.Регистратор) КАК Документ
ИЗ
РегистрНакопления.УчетНоменклатуры КАК УчетНоменклатуры

[Копировать в буфер обмена](#)**Функция ПРЕДСТАВЛЕНИЕ**

Описание:

Данная функция предназначена для получения строкового представления значения произвольного типа.

Параметр функции – выражение любого типа.

Возвращаемое значение – представление значения, тип **Строка**.Результат работы функции не может быть использован внутри других функций, за исключением функции **ПРЕДСТАВЛЕНИЕ**.

Пример:

ВЫБРАТЬ
ПРЕДСТАВЛЕНИЕ(Документ.Контрагент) КАК Получатель,
ПРЕДСТАВЛЕНИЕ(Документ.Дата) КАК Дата
ИЗ
Документ.РасходнаяНакладная КАК Документ

[Копировать в буфер обмена](#)

Результат:

Получатель	Дата
Эльбрус	28.06.2006 14:19:00
Эльбрус	28.06.2006 14:30:32
Алекс-2002	28.06.2006 14:30:49
Большаков Андрей	28.06.2006 14:32:06
Завод РТИ	28.06.2006 14:32:32
Филлипенко	28.06.2006 14:32:47
Центр детского творчества	28.06.2006 14:34:04
Алекс-2002	28.06.2006 14:35:37
Магазин на ул. Алексеева	28.06.2006 14:36:05
Никитин Юбий	28.06.2006 14:36:36

Рис. 240. Результат запроса

Функция ЕСТЬNULL (ISNULL)

Описание:

Функция предназначена для замены значения **NULL** на другое значение.

Параметры функции:

- первый параметр – выражение любого типа;
- второй параметр – выражение любого типа.

Возвращаемое значение: значение первого параметра, если первый параметр не содержит значение **NULL**; значение второго параметра в противном случае.

Второй параметр будет преобразован к типу первого в том случае, если тип первого параметра является строкой или числом.

Пример:

[Копировать в буфер обмена](#)

1538

```
// Получить сумму по полю Количество. В случае, если нет
// записей, получить 0
ВЫБРАТЬ
    ЕСТЬNULL(СУММА(РасходнаяНакладнаяСостав.Количество), 0)
ИЗ
    Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав
```

8.2.16.4. Операции выбора в языке запросов

В выражениях языка запросов могут применяться операции выбора, которые позволяют получить одно из возможных значений в соответствии с указанными условиями.

Операция выбора описывается следующим набором правил:

<Операция выбора>

[Копировать в буфер обмена](#)

```
ВЫБОР
<Альтернативы выбора>
[ИНАЧЕ <Выражение>]
КОНЕЦ
```

<Альтернативы выбора>

[Копировать в буфер обмена](#)

```
<Одиночный выбор>
[<Альтернативы выбора>]
```

<Одиночный выбор>

[Копировать в буфер обмена](#)

```
КОГДА <Логическое выражение>
ТОГДА <Выражение>
```

В операции выбора может указываться неограниченное количество альтернативных одиночных выборов **КОГДА ... ТОГДА**. Они обрабатываются в запросе последовательно. Если логическое выражение имеет значение *Истина*, обработка операции выбора завершается. Результатом операции является значение выражения, указанного после слова **ТОГДА**. Описание логических выражений см. [здесь](#).

Значение выражения, указанного после слова **ИНАЧЕ**, используется в качестве результата операции выбора в том случае, если во всех ранее указанных альтернативных одиночных выборах предикат имел значение *Ложь*.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    Номенклатура.Наименование,
ВЫБОР
    КОГДА Номенклатура.ЭтоГруппа = ИСТИНА
        ТОГДА "Это группа"
    КОГДА Номенклатура.ЗакупочнаяЦена > 1000
        ТОГДА "1000 -"
    КОГДА Номенклатура.ЗакупочнаяЦена > 100
        ТОГДА "100 - 1000"
    КОГДА Номенклатура.ЗакупочнаяЦена > 10
        ТОГДА "10 - 100"
    КОГДА Номенклатура.ЗакупочнаяЦена > 0
        ТОГДА "0 - 10"
    ИНАЧЕ "Не задана"
КОНЕЦ КАК Цена
ИЗ
    Справочник.Номенклатура КАК Номенклатура
```

Результат:

Наименование	Цена
Мониторы	Это группа
Принтеры	Это группа
Мыши	Это группа
Клавиатуры	Это группа
Программное обеспечение	Это группа
Услуги	Это группа
Монитор 15' LG Studioworks 575N	100 - 1000
Монитор 17' Philips 107520	100 - 1000
Монитор 19' Hitachi CM715ET	100 - 1000
Монитор LCD 22' M8537ZM/A	1000 -
Сист. блок Hewlett-Packard Brio BA410	1000 -
Сист. блок Hewlett-Packard Verba VI 420	1000 -

Рис. 241. Результат запроса (фрагмент)

8.2.16.5. Приведение типа в языке запросов

Поля исходных таблиц могут иметь составной тип. Для таких полей возникает необходимость привести значения поля к какому-либо определенному типу. В языке запросов предусмотрена возможность приведения типа. Ею можно пользоваться в списке полей выборки и в условии отбора в предложении **ГДЕ**.

<Приведение типа>

[Копировать в буфер обмена](#)

```
ВЫРАЗИТЬ ( <Выражение> КАК <Тип значения> )
```

<Тип значения>

[Копировать в буфер обмена](#)

```
Булево |
Число [(Длина[, Точность])]|
Строка [(Длина)]|
Дата |
<Имя таблицы>
```

<Длина>

[Копировать в буфер обмена](#)

Число

<Точность>

[Копировать в буфер обмена](#)

Число

Выражение приводится к одному из примитивных типов или к ссылочному типу данных. В последнем случае имя таблицы указывает на соответствующую таблицу информационной базы.

Если выражение содержит в составном типе требуемый тип значения, то приведение типа считается осуществимым, и для каждого значения указанного типа результатом будет это самое значение. Для значений других типов результатом приведения типа будет значение `NULL`.

Если выражение не содержит в составном типе требуемый тип значения, то выполнение данного запроса завершится ошибкой из-за принципиальной невозможности совершить приведение типов.

8.2.16.6. Константы и параметры в языке запросов

В выражениях языка запросов могут напрямую указываться значения типа `Булево`, `Число`, `Строка` или `Дата`. Также могут использоваться значения параметров запроса.

<Значение>

[Копировать в буфер обмена](#)

```
ИСТИНА |
ЛОЖЬ |
<Литерал типа ЧИСЛО> |
<Литерал типа СТРОКА> |
<Литерал типа ДАТА> |
<Литерал типа ТИП> |
<Имя параметра> |
НЕОПРЕДЕЛЕНО |
NULL
```

<Литерал типа Число>

[Копировать в буфер обмена](#)

<Целое число>[.<Целое число>]

<Литерал типа Строка>

[Копировать в буфер обмена](#)

" <Последовательность символов> "

<Литерал типа Дата>

[Копировать в буфер обмена](#)

```
ДАТАВРЕМЯ ( <Целое число>, <Целое число>, <Целое число>[ ,
<Целое число>, <Целое число>, <Целое число> ] )
```

Значения типа `Булево`, `Число`, `Строка` в языке запросов задаются так же, как и во встроенном языке.

Значения типа `Дата` задаются с помощью ключевого слова `ДАТАВРЕМЯ`, после которого в скобках последовательно указываются год, месяц, день, час, минута, секунда. Последние три указывать необязательно.

ВНИМАНИЕ! Максимальная дата, которую возможно задать при помощи литерала `ДАТАВРЕМЯ` – `31.12.3999 23:59:59`.

В запрос могут передаваться параметры (см. описание объекта `Запрос`). Значения параметров могут использоваться в выражениях языка запросов. Для этого необходимо указать символ "&" и после него имя параметра.

<Литерал типа Тип>

[Копировать в буфер обмена](#)

ТИП(<Имя типа>)

<Имя типа> – имя примитивного типа либо имя таблицы, тип ссылки которой нужно получить. Результатом данной конструкции будет значение типа `Тип` для указанного типа.

Пример:

[Копировать в буфер обмена](#)

```
// Получение типа "Строка"
ТИП(Строка)
// Получение типа - ссылка на справочник "Номенклатура"
ТИП(Справочник.Номенклатура)
```

Значения типа `Тип` в языке запросов можно использовать в операциях сравнения, упорядочивания, группировки.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    ТИПЗНАЧЕНИЯ(Остатки.Регистратор)
ИЗ
    РегистрНакопления.УчетНоменклатуры КАК Остатки
ГДЕ
    ТИПЗНАЧЕНИЯ(Остатки.Регистратор) = ТИП(Документ.РасходнаяНакладная)
```

Возможна передача значения типа `Тип` как параметр запроса.

Пример:

[Копировать в буфер обмена](#)

ВЫБРАТЬ
 ТИПЗНАЧЕНИЯ(Остатки.Регистратор)
 ИЗ
 РегистрНакопления.УчетНоменклатуры КАК Остатки
 ГДЕ
 ТИПЗНАЧЕНИЯ(Остатки.Регистратор) = &Тип

При сравнении значения типа **Тип** значения упорядочиваются в следующем порядке (первый тип считается самым малым):

- NULL,
- Неопределено,
- Булево,
- Число,
- Дата,
- Строка,
- ссылка на таблицу,
- другие типы.

8.2.17. Условия в языке запросов

В языке запросов используются условия отбора, в соответствии с которыми осуществляется отбор данных в предложениях **ГДЕ**, **ИМЕЮЩИЕ** и **СОЕДИНЕНИЕ**. Условия описываются по следующим правилам:

<Условие отбора >

<Логическое слагаемое> [ИЛИ <Логическое слагаемое>]

[Копировать в буфер обмена](#)

<Логическое слагаемое >

<Логический сомножитель > [И <Логический сомножитель >]

[Копировать в буфер обмена](#)

<Логический сомножитель >

НЕ <Логический сомножитель > |
 (<Условие отбора >) |
 <Логическое выражение >

[Копировать в буфер обмена](#)

В простейшем случае условие является выражением, результат которого имеет значение логического типа. Логические выражения описаны в следующем разделе.

Условия могут определяться и как более сложные логические выражения, где фигурируют простые логические выражения, соединенные между собой с помощью логических операторов **И**, **ИЛИ**, **НЕ**.

Логические операторы имеют приоритет:

- Самый высокий приоритет имеет логический оператор **НЕ**.
- Следующим по приоритету является оператор **И**.
- Самый низкий приоритет у оператора **ИЛИ**.
- В условиях сначала вычисляются простые логические выражения, затем операции **НЕ**, затем операции **И**, в последнюю очередь – операции **ИЛИ**. Для того чтобы обеспечить другой порядок вычислений, можно использовать круглые скобки **()**.

8.2.17.1. Логические выражения в языке запросов

В языке запросов в операциях выбора и в условиях отборов используются логические выражения.

<Логическое выражение >

<Выражение > |
 (<Выражение > | <Логическое выражение >) <Операция сравнения > (<Выражение > | <Логическое выражение >) |
 <Выражение > [НЕ] В [ИЕРАРХИИ] (<Список значений >) |
 <Выражение > [НЕ] В [ИЕРАРХИИ](<Описание запроса >) |
 <Выражение > [НЕ] МЕЖДУ <Выражение > И <Выражение > |
 <Выражение > ЕСТЬ [НЕ] NULL |
 <Выражение > ССЫЛКА <Имя таблицы > |
 <Выражение > [НЕ] ПОДОБНО <Литерал типа СТРОКА >
 [СПЕЦСИМВОЛ <Литерал типа СТРОКА >]

[Копировать в буфер обмена](#)

<Операция сравнения >

> | < | = | >= | <= | <>

[Копировать в буфер обмена](#)

<Список значений >

<Выражение >[, <Выражение > [, ...]]

[Копировать в буфер обмена](#)

Логическим выражением может быть:

- обычное выражение языка запросов, если его результат имеет логический тип;

- операция сравнения двух выражений языка запросов; выполняется в соответствии с правилами сравнения значений, описанными в разделе «Правила сравнения значений»;
- оператор проверки совпадения/несовпадения значения выражения с одним из перечисленных или со значениями, содержащимися в результате другого запроса;
- оператор проверки вхождения значения выражения в диапазон;
- оператор проверки значения выражения на **NULL**;
- оператор проверки ссылочного значения выражения на ссылку на определенную таблицу;
- оператор проверки строкового значения наподобие шаблону.

При сравнении значений используются правила сравнения значений, описанные ниже.

8.2.17.2. Правила сравнения значений

Поскольку в языке запросов могут сравниваться значения разных типов, определены правила, по которым выполняется сравнение двух значений.

Данные правила используются:

- для сравнения значений в операторах сравнения;
- для определения максимального и минимального значений в агрегатных функциях **МИНИМУМ** и **МАКСИМУМ**;
- для упорядочивания записей результата запроса в соответствии с порядком, заданным в предложении **УПОРЯДОЧИТЬ ПО**.

Если типы значений отличаются друг от друга, то отношения между значениями определяются на основании приоритета типов:

- тип **NULL** (самый низший);
- тип **Булево**;
- тип **Число**;
- тип **Дата**;
- тип **Строка**;
- ссылочные типы.

Отношения между различными ссылочными типами определяются на основе внутренних ссылочных номеров таблиц, соответствующих тому или иному типу.

Если типы данных совпадают, то производится сравнение значений по следующим правилам:

- У типа **Булево** значение **Истина** больше значения **Ложь**.
- У типа **Число** обычные правила сравнения для чисел.
- У типа **Дата** более ранние даты меньше более поздних.
- У типа **Строка** сравнение производится в соответствии с установленными национальными особенностями базы данных. Операция сравнения строк не учитывает концевые пробелы, в отличие от сравнения строк на встроенном языке, где концевые пробелы участвуют в операции сравнения. Например, сравнение строк «**bb**» и «**bb** » на встроенном языке вернет **Ложь** (строки не равны), и **Истина** в языке запросов (строки равны).
- Ссылочные типы сравниваются на основе своих значений (номера записи и т. п.).
- Не поддерживается сравнение полей типа **УникальныйИдентификатор** с полями других типов;
- Не допускается сравнение полей неограниченной длины (строки неограниченной длины, **ХранилищеЗначения**, поле **ТипЗначения** из таблицы планов видов характеристик).

ВНИМАНИЕ! Любая операция сравнения двух значений, в которой участвует хотя бы одно значение **NULL**, дает результат, аналогичный значению **Ложь**.

8.2.17.3. Оператор проверки совпадения значения

Форма оператора «В» для проверки совпадения с одним из перечисленных

Оператор **В** позволяет проверить, совпадает ли значение выражения, указанного справа от него, с одним из значений, описанных слева. Если совпадает хотя бы с одним, результатом оператора будет **Истина**, иначе – **Ложь**. Применение **НЕ** изменяет действие оператора на обратное. Сравнение значений производится по правилам, описанным в разделе «Правила сравнения значений».

Пример:

```

ВЫБРАТЬ
    Номенклатура.Наименование
ИЗ
    Справочник.Номенклатура КАК Номенклатура
ГДЕ
    Номенклатура.Родитель.Наименование В
    ("Бытовая техника", "Оргтехника")
  
```

[Копировать в буфер обмена](#)

Форма оператора «В» для проверки принадлежности по иерархии

Для справочников проверка может осуществляться и на принадлежность по иерархии. Результатом оператора **В ИЕРАРХИИ** будет **Истина**, если значение выражения слева является ссылкой на элемент справочника и входит во множество значений справа или иерархически принадлежит какой-нибудь группе, содержащейся в этом множестве.

Пример:

```
// В качестве параметра Группа в запрос передается ссылка
// на какую-либо группу справочника Номенклатура.
```

```
ВЫБРАТЬ
    Номенклатура.Наименование
ИЗ
    Справочник.Номенклатура КАК Номенклатура
ГДЕ
    Номенклатура.Ссылка В ИЕРАРХИИ(&Группа)
```

В качестве множества значений, на совпадение с которыми выполняется проверка, может фигурировать и результат запроса. В этом случае справа от оператора **В** необходимо указать описание запроса.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    Номенклатура.Наименование
ИЗ
    Справочник.Номенклатура КАК Номенклатура
ГДЕ
    Номенклатура.Ссылка В ИЕРАРХИИ
        (ВЫБРАТЬ
            Номенклатура.Ссылка
        ИЗ
            Справочник.Номенклатура КАК Номенклатура
        ГДЕ
            Номенклатура.Наименование = "Одежда" )
```

Форма оператора «В» для проверки совпадения значения с одним из результатов запроса

Примером применения данного оператора может послужить следующий.

Пример:

[Копировать в буфер обмена](#)

```
// Выбрать названия товаров, которые присутствовали
// в расходных накладных
ВЫБРАТЬ
    Товары.Наименование
ИЗ
    Справочник.Номенклатура КАК Товары
ГДЕ
    Товары.Ссылка В
        (ВЫБРАТЬ
            РасходнаяНакладнаяСостав.Номенклатура
        ИЗ
            Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав)
```

Результат:

Наименование
Монитор 15' LG Studioworks 575N
Монитор 17' Philips 107S20
Монитор 19' Hitachi CM715ET
Монитор LCD 22' M8537ZM/A
Сист. блок Hewlett-Packard Brio BA410
Сист. блок Hewlett-Packard Vectra VL420
Сист. блок IBM NetVista A22p
Сист. блок IBM NetVista M41
Лазерный принтер Canon LBP-810
Лазерный принтер 5250197-203 Minolta-QMS
Лазерный принтер HP LaserJet 2200
Мышь 2-кноп А4Tech PS/2

Рис. 242. Результат запроса (фрагмент)

Для получения противоположного результата, то есть если нужно определить, что значение не совпадает ни с одним из результатов запроса, запрос выглядит следующим образом.

Пример:

[Копировать в буфер обмена](#)

```
// Выбрать названия товаров, которые присутствовали
// в расходных накладных
ВЫБРАТЬ
    Товары.Наименование
ИЗ
    Справочник.Номенклатура КАК Товары
ГДЕ
    (НЕ Товары.Ссылка В
        (ВЫБРАТЬ
            РасходнаяНакладнаяСостав.Номенклатура
        ИЗ
            Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав))
```

Результат:

Наименование
Мониторы
Принтеры
Мыши
Клавиатуры
Программное обеспечение
Услуги
Windows
Доставка
Инсталляция ПО
Консультации по настройке ОС Windows
Консультации по настройке 1С
Налётбли

Рис. 243. Результат запроса (фрагмент)

Заметим, что из запроса операции **В** возможно обращение к полям таблиц, которые встречались во внешнем запросе до появления операции.

Пример:

[Копировать в буфер обмена](#)

```
// Выбрать названия товаров, которые присутствовали
// в расходных накладных
ВЫБРАТЬ
    Товары.Наименование
ИЗ
    Справочник.Номенклатура КАК Товары
ГДЕ
    Товары.Ссылка В
    (ВЫБРАТЬ
        РасходнаяНакладнаяСостав.Номенклатура
ИЗ
        Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав
ГДЕ
        РасходнаяНакладнаяСостав.Номенклатура = Товары.Ссылка)
```

Результат:

Наименование
Монитор 15' LG Studioworks 575N
Монитор 17' Philips 107S20
Монитор 19' Hitachi CM715ET
Монитор LCD 22' M8537ZM/A
Сист. блок Hewlett-Packard Brio BA410
Сист. блок Hewlett-Packard Vectra VL420
Сист. блок IBM NetVista A22p
Сист. блок IBM NetVista M41
Лазерный принтер Canon LBP-810
Лазерный принтер 5250197-203 Minolta-QMS
Лазерный принтер HP LaserJet 2200
Мышь 2-кноп А4Tech PS/2

Рис. 244. Результат запроса (фрагмент)

Использование операции «В» («НЕ В») по нескольким полям

Синтаксис для вложенного запроса:

```
(выражение1, выражение2, ..., выражениеN) В (ВЫБРАТЬ выражение1, выражение2, ..., выражениеN ...)
```

Синтаксис для таблицы значений:

```
(выражение1, выражение2, ..., выражениеN) В (&Параметр)
```

В качестве параметра следует передавать таблицу значений, первые N колонок которой будут использоваться для операции **В**.

8.2.17.4. Оператор проверки вхождения значения в диапазон

Оператор **МЕЖДУ** позволяет проверить, входит ли значение выражения, указанного слева от него, в диапазон, указанный справа. Если входит, результатом оператора будет **Истина**, иначе – **Ложь**. Применение **НЕ** изменяет действие оператора на обратное. Правила сравнения значений см. [здесь](#).

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    Номенклатура.Наименование,
    Номенклатура.ЗакупочнаяЦена
ИЗ
    Справочник.Номенклатура КАК Номенклатура
ГДЕ
    Номенклатура.ЗакупочнаяЦена МЕЖДУ 100 И 1000
```

8.2.17.5. Оператор проверки значения на NULL

Оператор **ЕСТЬNULL** позволяет проверить значение выражения слева от него на **NULL**. Если значение равно **NULL**, результатом оператора будет **Истина**, иначе – **Ложь**. Применение **НЕ** изменяет действие оператора на обратное.

Пример:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    Номенклатура.Наименование,
    Номенклатура.ЗакупочнаяЦена
ИЗ
    Справочник.Номенклатура КАК Номенклатура
ГДЕ
    Номенклатура.ЗакупочнаяЦена ЕСТЬ NULL
```

8.2.17.6. Оператор проверки ссылочного значения

Оператор **ССЫЛКА** позволяет проверить, является ли значение выражения, указанного слева от него, ссылкой на таблицу, указанную справа. Если да, результатом оператора будет *Истина*, иначе – *Ложь*. Описание разыменования таблиц см. [здесь](#).

Пример:

```

ВЫБРАТЬ
    РасходнаяНакладная.Номер,
    РасходнаяНакладная.Дата
ИЗ
    Документ.РасходнаяНакладная КАК РасходнаяНакладная
ГДЕ
    РасходнаяНакладная.Контрагент ССЫЛКА Справочник.Контрагенты

```

[Копировать в буфер обмена](#)

8.2.17.7. Оператор проверки строки на подобие шаблону

Оператор **ПОДОВНО** позволяет сравнить значение выражения, указанного слева от него, со строкой шаблона, указанной справа. Значение выражения должно иметь тип *Строка*. Если значение выражения удовлетворяет шаблону, результатом оператора будет *Истина*, иначе – *Ложь*.

Следующие символы в строке шаблона являются служебными и имеют смысл, отличный от символа строки:

- «%» (процент): последовательность, содержащая любое количество произвольных символов.
- «_» (подчеркивание): один произвольный символ.
- «[...]» (в квадратных скобках один или несколько символов): любой одиночный символ из перечисленных внутри квадратных скобок. В перечислении могут встречаться диапазоны, например, a-z, означающие произвольный символ, входящий в диапазон, включая концы диапазона.
- «[...]» (в квадратных скобках значок отрицания, за которым следует один или несколько символов): любой одиночный символ, кроме тех, которые перечислены следом за значком отрицания.

Любой другой символ означает сам себя и не несет никакой дополнительной нагрузки.

Если в качестве самого себя необходимо записать один из перечисленных символов, то ему должен предшествовать спецсимвол. Сам спецсимвол (любой подходящий символ) определяется в этом же операторе после ключевого слова **СПЕЦСИМВОЛ**.

Например, шаблон:

```
"%АВВ[0-9]{абвг}\_абв%" СПЕЦСИМВОЛ "\"
```

[Копировать в буфер обмена](#)

означает подстроку, состоящую из последовательности символов:

- буквы А,
- буквы Б,
- буквы В,
- одной цифры,
- одной из букв а, б, в или г,
- символа подчеркивания,
- буквы а,
- буквы б,
- буквы в.

Причем перед этой последовательностью может располагаться произвольный набор символов.

При работе в клиент-серверном варианте, возможна ситуация, когда запрос, содержащий оператор **ПОДОВНО**, не может быть выполнен. Такая ситуация возможна в том случае, когда длина строки с шаблоном превышает некоторое число, зависящее от используемой СУБД.

8.3. Выполнение и работа с запросами во встроенном языке

Для формирования запросов, выборки и обработки результатов запросов в языке предусмотрен специальный набор объектов. С помощью этих объектов выполняется формирование запроса, обход записей запроса и т. д.

8.3.1. Основные приемы работы

Основные приемы работы с запросами во встроенном языке системы «1С:Предприятие» удобнее всего рассматривать на примерах. Приведем типичный пример использования запроса.

```

// Создадим Запрос
Запрос = Новый Запрос( "ВЫБРАТЬ Товар.Наименование Наименование,
    |Товар.Родитель.Наименование НаименованиеРодителя
    |ИЗ Справочник.Товары Товар" );
// Выполним запрос и запишем результат в переменную РезультатЗапроса.
РезультатЗапроса = Запрос.Выполнить();
// Получим выборку из результата запроса.
Выборка = РезультатЗапроса.Выбрать();
// Пока в выборке есть записи ...
Пока Выборка.Следующий() Цикл
    // ... выведем в окно сообщений поля из результата.
    Товар = Выборка.Наименование;
    Родитель = Выборка.НаименованиеРодителя;
    Сообщить( "Товар: " + Товар + " Родитель: " + Родитель );
КонецЦикла;

```

[Копировать в буфер обмена](#)

Как видно из этого примера, работа с запросом ведется при помощи трех основных объектов:

- **Запрос** – объект, выполняющий сам запрос. Представлен в примере переменной с именем `Запрос`.
- **РезультатЗапроса** – объект, содержащий полученные при выполнении запроса данные. Представлен в примере переменной с именем `РезультатЗапроса`.
- **ВыборкаИзРезультатаЗапроса** – объект, позволяющий обходить (т. е. перебирать) записи из результата. Представлен в примере переменной с именем `Выборка`.

Рассмотрим подробнее объект `ВыборкаИзРезультатаЗапроса`.

Для этого нам понадобится следующий запрос:

```

ВЫБРАТЬ
    РасходнаяНакладнаяСостав.Номенклатура КАК Номенклатура,
    РасходнаяНакладнаяСостав.Количество КАК Количество
ИЗ
    Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав
УПОРЯДОЧИТЬ ПО
    РасходнаяНакладнаяСостав.Номенклатура
ИТОГИ
    СУММА(Количество)
ПО
    Номенклатура ИЕРАРХИЯ

```

Результат:

№	Номенклатура	Количество
1	Сантехника	104
2	Кран	84
3	Кран	10
4	Кран	8
5	Кран	44
6	Кран	22
7	Смеситель	20
8	Смеситель	5
9	Смеситель	1
10	Смеситель	14
11	Мебель	134
12	Стол	26
13	Стол	1
14	Стол	15
15	Стол	10
16	Стул	108
17	Стул	55
18	Стул	5
19	Стул	32
20	Стул	16

Рис. 245. Рабочая выборка

В этой таблице мы добавили первый столбец, которого нет в тексте запроса, но он будет использоваться нами в дальнейшем для идентификации записи в результате. Итоговые записи в таблице выделены курсивом, итоговые записи для иерархических уровней справочника выделены жирным шрифтом.

8.3.1.1. Способы обхода результата запроса

8.3.1.1.1. Линейный обход результата

Первый и самый простой способ обхода – линейный. При линейном обходе выборка будет выдавать записи в той последовательности, в которой они располагаются в результате запроса. В нашем примере это будут записи с номерами 1, 2, 3, 4, 5 и так далее до записи с номером 20.

Для получения линейной выборки из результата необходимо вызвать метод `Выбрать()` объекта `РезультатЗапроса` без параметров либо с параметром `ОбходРезультатаЗапроса.Прямой`.

Пример:

```

СпособВыборки = ОбходРезультатаЗапроса.Прямой;
Выборка1 = РезультатЗапроса.Выбрать(СпособВыборки);
// что равнозначно записи
Выборка1 = РезультатЗапроса.Выбрать();

```

8.3.1.1.2. Иерархический обход результата

Следующий способ обхода результата – иерархический. При данном способе обходятся только записи, находящиеся на одном уровне. Для получения иерархической выборки из результата необходимо вызвать метод `Выбрать()` объекта `РезультатЗапроса` с параметром `ОбходРезультатаЗапроса.ПоГруппировкамСИерархией`.

Пример:

```

СпособВыборки = ОбходРезультатаЗапроса.ПоГруппировкамСИерархией;
Выборка2 = РезультатЗапроса.Выбрать(СпособВыборки);

```

Выборка из результата с иерархическим обходом в нашем примере обойдет только записи с номерами 1 и 11, так как только эти две записи находятся на самом верхнем уровне. Проиллюстрируем это, представив наш результат в виде дерева, где узлами будут итоговые записи, а листьями дерева – детальные записи. Вот что у нас получится:

Копировать в буфер обмена

Копировать в буфер обмена

Копировать в буфер обмена

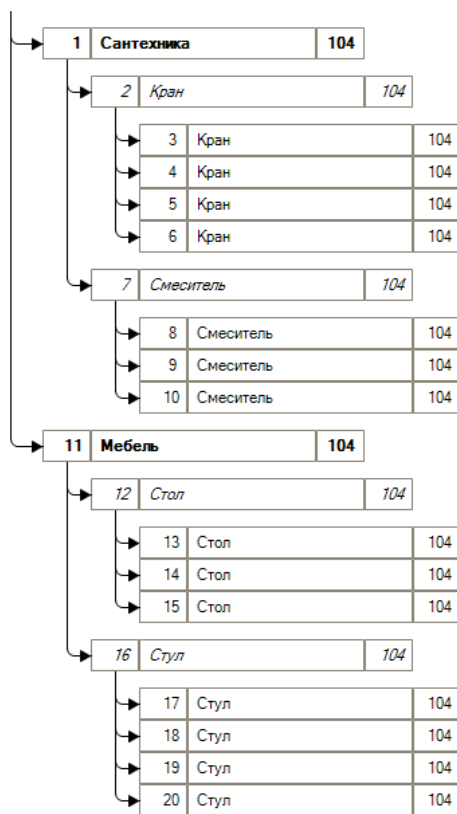


Рис. 246. Иерархический обход результата запроса

Из этого рисунка видно, что именно записи с номерами 1 и 11 и только они находятся на первом уровне дерева, в результате чего только они и попадают в первый проход иерархической выборки.

Возникает вопрос: как получать остальные записи результата запроса? Для этого у объекта `ВыборкаИзРезультатаЗапроса` можно получить еще одну выборку, которая будет обходить подчиненные записи текущей записи выборки. В нашем примере в момент, когда объект `Выборка2` будет позиционирован на записи с номером 1, мы запросим у него иерархическую выборку. Таким образом мы получим выборку, которая нам вернет записи с номерами 2, 7. А когда `Выборка2` будет позиционирована на записи с номером 11, полученная у нее иерархическая выборка вернет записи с номерами 12, 16. Так реализуется иерархический обход результатов запроса.

Заметим, что у выборки можно получать вложенные выборки любого типа. Так, если бы мы запросили у объекта `Выборка2`, позиционированной на записи 1, линейную выборку, то с ее помощью мы бы получили записи с номерами со 2-го по 10-й. Проиллюстрируем описанную методику на примере.

Пример:

[Копировать в буфер обмена](#)

```

Процедура ВыполнитьЗапрос()
// Создадим запрос.
Запрос = Новый Запрос;
// Установим текст запроса
Запрос.Текст = "ВЫБРАТЬ
|РасходнаяНакладнаяСостав.Номенклатура КАК Номенклатура,
|РасходнаяНакладнаяСостав.Количество КАК Количество
|ИЗ
|Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав
|УПОРЯДОЧИТЬ ПО
|РасходнаяНакладнаяСостав.Номенклатура
|ИТОГИ
|СУММА(Количество)
|ПО
|Номенклатура ИЕРАРХИЯ";
// Выполним запрос и запишем результат в переменную
// РезультатЗапроса.
РезультатЗапроса = Запрос.Выполнить();
// Получим выборку из результата запроса.
СпособВыборки = ОбходРезультатаЗапроса.ПоГруппировкамСИерархией;
Выборка = РезультатЗапроса.Выбрать(СпособВыборки);
ВыдатьРекурсивно(Выборка);
КонецПроцедуры
Процедура ВыдатьРекурсивно(Выборка)
// Пока в выборке есть записи ...
Пока Выборка.Следующий() Цикл
// ... выведем в окно сообщений поля из результата
Товар = Выборка.Наименование;
Количество = Выборка.Количество;
Сообщить("Товар: " + Товар + " Количество: " + Количество);
// Продолжим выборку подчиненных записей
СпособВыборки = ОбходРезультатаЗапроса.ПоГруппировкамСИерархией;
ВыдатьРекурсивно(Выборка.Выбрать(СпособВыборки, Выборка.Группировка()));
КонецЦикла;
КонецПроцедуры
  
```

8.3.1.1.3. Обход результата по группам

Третий и последний способ обхода результата – по группам. Он сходен с иерархическим обходом, но с одним различием: записи с иерархическими итогами при обходе в нем рассматриваются как детальные, а не как итоговые. Для получения выборки по группам из результата запроса необходимо вызвать метод `Выбрать()` объекта `РезультатЗапроса` с параметром `ОбходРезультатаЗапроса.ПоГруппировкам`.

Пример:

```
СпособВыборки = ОбходРезультатаЗапроса.ПоГруппировкам;
Выборка2 = РезультатЗапроса.Выбрать(СпособВыборки);
// Перебрав в ней все, мы получим записи с номерами:
// 1, 2, 7, 11, 12, 16.
```

Пример:

Копировать в буфер обмена

```
Процедура ВыполнитьЗапрос()
// Создадим запрос.
Запрос = Новый Запрос;
// Установим текст запроса
Запрос.Текст = "ВЫБРАТЬ
|РасходнаяНакладнаяСостав.Номенклатура КАК Номенклатура,
|РасходнаяНакладнаяСостав.Количество КАК Количество
|ИЗ
|Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав
|УПОРЯДОЧИТЬ ПО
|РасходнаяНакладнаяСостав.Номенклатура
|ИТОГИ
|СУММА(Количество)
|ПО
|Номенклатура ИЕРАРХИЯ";
// Выполним запрос и запишем результат в переменную
// РезультатЗапроса.
РезультатЗапроса = Запрос.Выполнить();
// Получим выборку из результата запроса
СпособВыборки = ОбходРезультатаЗапроса.ПоГруппировкам;
Выборка = РезультатЗапроса.Выбрать(СпособВыборки);
// Пока в выборке есть записи ...
Пока Выборка.Следующий() Цикл
// ... выведем в окно сообщений поля из результата
Товар = Выборка.Наименование;
Количество = Выборка.Количество;
Сообщить("Товар: "+Товар+" Итого по товару: "+Количество);
ВыдатьДочерниеЗаписи(Выборка.Выбрать());
КонецЦикла;
КонецПроцедуры
Процедура ВыдатьДочерниеЗаписи(Выборка)
// Пока в выборке есть записи ...
Пока Выборка.Следующий() Цикл
// ... выведем в окно сообщений поля из результата
Товар = Выборка.Наименование;
Количество = Выборка.Количество;
Сообщить("Товар: "+Товар+" Количество: "+Количество);
КонецЦикла;
КонецПроцедуры
```

8.3.1.2. Работа с выборкой

Объект `ВыборкаИзРезультатаЗапроса` предназначен для обхода записей результата запроса. Можно представить себе выборку как некоторый объект, который содержит указатель на текущую запись в результате и предоставляет программе доступ ко всем полям текущей записи. Для навигации по записям запроса используются три метода:

- `Следующий()` – перейти к следующей записи результата в соответствии с порядком обхода выборки. При первом вызове позиционирует выборку на первую запись. Когда будут выбраны все записи, данный метод просигнализирует об этом, вернув значение `Ложь`.
- `СледующийПоЗначениюПоля()` – получить следующую запись со значением в заданном поле, отличающимся от значения в этом поле текущей записи.
- `НайтиСледующий()` – найти запись с заданными значениями некоторых полей.

8.3.1.2.1. Использование метода «СледующийПоЗначениюПоля()»

Метод позволяет сгруппировать записи результата по значениям полей.

Пример:

Копировать в буфер обмена

```
ВЫБРАТЬ
Док.Номенклатура,
Док.Ссылка.Контрагент КАК Контрагент,
Док.Количество
ИЗ
Документ.РасходнаяНакладная.Состав КАК Док
УПОРЯДОЧИТЬ ПО
Док.Номенклатура.Наименование,
Контрагент
```

Результат:

№	Номенклатура	Контрагент	Количество
1	Кран	Магазин "Гигант"	10,00
2	Кран	Магазин "Хозяин"	8,00
3	Кран	Мосгорторг	44,00
4	Кран	Магазин "Мебель"	22,00
5	Смеситель	Магазин "Гигант"	5,00
6	Смеситель	Магазин "Хозяин"	1,00
7	Смеситель	Мосгорторг	14,00
8	Смеситель	Мосгорторг	13,00
9	Стол	Магазин "Гигант"	1,00
10	Стол	Магазин "Хозяин"	15,00
11	Стол	Мосгорторг	10,00
12	Стул	Магазин "Гигант"	55,00
13	Стул	Магазин "Хозяин"	5,00
14	Стул	Мосгорторг	32,00
15	Стул	Магазин "Мебель"	16,00

Рис. 247. Результат запроса

Получим линейную выборку из результата запроса и обойдем выборку при помощи метода `СледующийПоЗначениюПоля()`.

Пример:

[Копировать в буфер обмена](#)

```
Выборка = РезультатЗапроса.Выбрать();
Пока Выборка.СледующийПоЗначениюПоля("Товар") Цикл
    // здесь мы получим записи с номерами 1, 5, 9, 12
    Пока Выборка.СледующийПоЗначениюПоля("Контрагент") Цикл
        // здесь мы сначала получим записи с номерами 1, 2, 3, 4
        // затем 5, 6, 7
        // затем 9, 10, 11
        // затем 12, 13, 14, 15
    КонечЦикла;
КонечЦикла;
```

Следует обратить внимание на то, что во внутреннем цикле не была выбрана запись с номером 8, т. к. в ней такое же значение поля `Получатель`, как и в предыдущей записи.

Заметим, что если в цикле получения по значению поля получать записи при помощи метода `Следующий()`, то будут выбраны все записи с равным значением поля, заданного в предыдущем вызове метода `СледующийПоЗначениюПоля()`.

Пример:

[Копировать в буфер обмена](#)

```
Выборка = РезультатЗапроса.Выбрать();
Пока Выборка.СледующийПоЗначениюПоля("Товар") Цикл
    // здесь мы получим записи с номерами 1, 5, 9, 12
    Пока Выборка.Следующий() Цикл
        // здесь мы сначала получим записи с номерами 1, 2, 3, 4
        // затем 5, 6, 7, 8
        // затем 9, 10, 11
        // затем 12, 13, 14, 15
    КонечЦикла;
КонечЦикла;
```

8.3.1.2.2. Методы определения типа текущей записи

В тот момент, когда выборка позиционирована на записи, мы можем у выборки узнать характеристики этой записи. Получение характеристик записи осуществляется следующими методами:

- `Уровень()` – определяет уровень записи в результате запроса.
- `ТипЗаписи()` – определяет принадлежность записи к одному из следующих типов:
 - групповой итог,
 - итога по иерархии,
 - детальная запись,
 - общий итог.
- `Группировка()` – определяет имя поля, по которому были рассчитаны итоги.

Для иллюстрации работы этих методов посмотрим, что они будут возвращать в виде записей для запроса, рассматриваемого в начале главы.

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ
    РасходнаяНакладнаяСостав.Номенклатура КАК Номенклатура,
    РасходнаяНакладнаяСостав.Количество КАК Количество
ИЗ
    Документ.РасходнаяНакладная.Состав КАК РасходнаяНакладнаяСостав
УПОРЯДОЧИТЬ ПО
    РасходнаяНакладнаяСостав.Номенклатура
ИТОГИ
    СУММА(Количество)
ПО
    Номенклатура ИЕРАРХИЯ
```

Результат:

№	Номенклатура	Количество	Уровень	ТипЗаписи	Группировка
1	Сантехника	117	0	Итог по иерархии	Номенклатура
2	Кран	84	1	Итог по группировке	Номенклатура
3	Кран	10	2	Детальная запись	
4	Кран	8	2	Детальная запись	
5	Кран	44	2	Детальная запись	
6	Кран	22	2	Детальная запись	
7	Смеситель	33	1	Итог по группировке	Номенклатура
8	Смеситель	5	2	Детальная запись	
9	Смеситель	1	2	Детальная запись	
0	Смеситель	14	2	Детальная запись	
10	Смеситель	13	2	Детальная запись	
11	Мебель	134	0	Итог по иерархии	Номенклатура
12	Стол	26	1	Итог по группировке	Номенклатура
13	Стол	1	2	Детальная запись	
14	Стол	15	2	Детальная запись	
15	Стол	10	2	Детальная запись	
16	Стул	108	1	Итог по группировке	Номенклатура
17	Стул	55	2	Детальная запись	
18	Стул	5	2	Детальная запись	
19	Стул	32	2	Детальная запись	
20	Стул	16	2	Детальная запись	

Рис. 248. Иерархия записей

8.3.2. Работа с временными таблицами

Язык запросов системы «1С:Предприятие» позволяет использовать временные таблицы в запросах. Использование временных таблиц помогает повысить производительность запросов и сделать текст сложных запросов более легким для восприятия.

Работа с временными таблицами обеспечивается двумя составляющими:

- объектом встроенного языка `МенеджерВременныхТаблиц`, который хранит в себе данные временных таблиц;
- синтаксисом языка запросов, позволяющим создавать новые временные таблицы и использовать существующие временные таблицы.

8.3.2.1. Менеджер временных таблиц

Менеджер временных таблиц предназначен для управления временем существования временных таблиц, создаваемых в процессе работы прикладного решения.

В одном прикладном решении может быть создано произвольное количество экземпляров менеджера временных таблиц, каждый из которых хранит свой набор временных таблиц. Каждая временная таблица однозначно идентифицируется своим именем, и в пределах одного менеджера временных таблиц все временные таблицы должны иметь уникальные имена.

ПРИМЕЧАНИЕ. Имена временных таблиц должны соответствовать требованиям, предъявляемым к именам переменных встроенного языка (см. [здесь](#)).

Экземпляр менеджера временных таблиц может быть создан с помощью конструктора `Новый`.

Пример:

```
МенеджерВременныхТаблиц = Новый МенеджерВременныхТаблиц;
```

[Копировать в буфер обмена](#)

Все временные таблицы, созданные в данном экземпляре менеджера, существуют до тех пор, пока существует сам экземпляр менеджера временных таблиц. При уничтожении экземпляра менеджера все временные таблицы, содержащиеся в нем, также удаляются.

Менеджер временных таблиц можно закрыть принудительно при помощи метода `Закрыть()`. При этом будут удалены все созданные в нем таблицы. Дальнейшая работа с данным экземпляром менеджера будет невозможна.

8.3.2.2. Создание временных таблиц

Создание временных таблиц осуществляется с помощью объекта `Запрос` встроенного языка системы «1С:Предприятие».

Связь запроса с менеджером временных таблиц осуществляется с помощью свойства `МенеджерВременныхТаблиц` запроса, в котором указывается тот экземпляр менеджера, в котором должны создаваться временные таблицы.

Пример:

```
МенеджерВременныхТаблиц = Новый МенеджерВременныхТаблиц;
Запрос = Новый Запрос;
Запрос.МенеджерВременныхТаблиц = МенеджерВременныхТаблиц;
```

[Копировать в буфер обмена](#)

Временная таблица может быть создана на основе данных базы данных или на основе внешнего источника данных (например, таблицы значений).

Для того чтобы создать временную таблицу на основе данных базы данных, следует установить объекту `Запрос` менеджер временных таблиц, а затем выполнить запрос к базе данных, используя ключевое слово `ПОМЕСТИТЬ`, после которого указать имя создаваемой временной таблицы. Ключевое слово `ПОМЕСТИТЬ` располагается после списка выборки запроса.

Пример:

```
ВЫБРАТЬ
    Номенклатура.Код,
    Номенклатура.Наименование
ПОМЕСТИТЬ ВременнаяТаблица
ИЗ
    Справочник.Номенклатура КАК Номенклатура
```

[Копировать в буфер обмена](#)

Результат исполнения такого запроса будет содержать одну строку с одной колонкой `Количество`, в которой будут находиться записи, помещенные в созданную таблицу.

Если менеджер временных таблиц не установлен или был закрыт, или в установленном менеджере временных таблиц уже существует таблица с указанным именем, будет выдана ошибка.

При создании временных таблиц не на основании внешнего источника можно использовать конструкцию `ДЛЯ ИЗМЕНЕНИЯ`. Это необходимо в тех случаях, когда требуется получить данные во временную таблицу и одновременно заблокировать их от чтения другими транзакциями.

Пример:

```
ВЫБРАТЬ
    РасходнаяНакладная.Ссылка,
    РасходнаяНакладная.Номер,
    РасходнаяНакладная.Дата
ПОМЕСТИТЬ ВременнаяТаблица
ИЗ
    Документ.РасходнаяНакладная КАК РасходнаяНакладная
ГДЕ
    РасходнаяНакладная.Ссылка В(&Документы)
ДЛЯ ИЗМЕНЕНИЯ
```

[Копировать в буфер обмена](#)

При необходимости создания индекса для временной таблицы следует в запросе указать ключевое слово `ИНДЕКСИРОВАТЬ ПО`, после которого перечислить поля, по которым нужно построить индекс.

Пример:

```
ВЫБРАТЬ
```

[Копировать в буфер обмена](#)

1550

Номенклатура.Код КАК Код,
Номенклатура.Наименование
ПОМЕСТИТЬ ВременнаяТаблица
ИЗ

Справочник.Номенклатура КАК Номенклатура
ИНДЕКСИРОВАТЬ ПО
Код

Поля, по которым происходит индексирование, должны находиться в списке выборки.

Если в качестве источника используется таблица значений, то у этой таблицы значений должны быть явно указаны типы значений, содержащихся в колонках.

Для того чтобы создать временную таблицу и заблокировать данные таблиц, на основании которых создается временная, следует использовать конструкцию для ИЗМЕНЕНИЯ.

Пример:

Копировать в буфер обмена

```
ВЫБРАТЬ
    РасходнаяНакладная.Ссылка ,
    РасходнаяНакладная.Номер ,
    РасходнаяНакладная.Дата
ПОМЕСТИТЬ ВременнаяТаблица
ИЗ
    Документ.РасходнаяНакладная КАК РасходнаяНакладная
ГДЕ
    РасходнаяНакладная.Ссылка В(&Документы)
ДЛЯ ИЗМЕНЕНИЯ
```

Для того чтобы создать временную таблицу на основании внешнего источника, следует в тексте запроса в списке источников указать имя параметра, в который будет помещен внешний источник. Остальной синтаксис идентичен обычному созданию временной таблицы.

В качестве внешнего источника могут выступать:

- таблица значений,
- табличная часть,
- результат запроса.

Ниже приведен пример создания временной таблицы на основе внешнего источника:

Копировать в буфер обмена

```
ВЫБРАТЬ
    Источник.Код ,
    Источник.Наименование
ПОМЕСТИТЬ ВременнаяТаблица
ИЗ
    &ВнешнийИсточник КАК Источник
```

В данном примере во временную таблицу ВременнаяТаблица будет помещено содержимое колонок Код и Наименование из внешнего источника, например, таблицы значений, переданной в качестве параметра ВнешнийИсточник.

ВНИМАНИЕ! Создание временной таблицы невозможно, если она создается на основании таблицы значений, которая содержит в колонках значение типа УникальныйИдентификатор.

ВНИМАНИЕ! Если временная таблица создается на основании внешнего источника, в запросе нельзя использовать объединения и соединения, а также поля, являющиеся реквизитами полей таблиц, на основании которых создается временная таблица.

8.3.2.3. Использование временных таблиц

Для использования существующих временных таблиц в запросе следует установить объекту Запрос менеджер временных таблиц, после чего к временным таблицам, содержащимся в данном менеджере временных таблиц, можно обращаться по имени, как к обычным таблицам запроса.

8.3.2.4. Удаление временных таблиц

Для удаления временной таблицы из менеджера временных таблиц следует воспользоваться ключевым словом языка запроса УНИЧТОЖИТЬ, после которого указывается имя уничтожаемой таблицы, например:

УНИЧТОЖИТЬ ВременнаяТаблица

Если уничтожаемой таблицы не существует, будет выдана ошибка.

8.3.3. Работа с пакетными запросами

Платформа системы «1С:Предприятие» позволяет работать с пакетами запросов. В пакетном запросе тексты запросов разделяются символом «;». Запросы исполняются последовательно, при этом временные таблицы, которые были созданы во время исполнения какого-либо запроса, будут существовать до окончания исполнения всего пакета запроса или до исполнения в пакете запроса, уничтожающего данную временную таблицу, например:

Копировать в буфер обмена

```
Запрос = Новый Запрос;
Запрос.Текст=
"ВЫБРАТЬ
|УчетНоменклатурыОстаткиОбороты.Номенклатура ,
|УчетНоменклатурыОстаткиОбороты.КоличествоПриход ,
|УчетНоменклатурыОстаткиОбороты.КоличествоРасход ,
|УчетНоменклатурыОстаткиОбороты.КоличествоКонечныйОстаток
|ПОМЕСТИТЬ УчетНоменклатуры
|ИЗ
|РегистрНакопления.УчетНоменклатуры.ОстаткиОбороты( , Авто, , ) КАК УчетНоменклатурыОстаткиОбороты
|;
|
|ВЫБРАТЬ
|УчетНоменклатуры.Номенклатура ,
|УчетНоменклатуры.КоличествоРасход ,
```

```

|УчетНоменклатуры.КоличествоКонечныйОстаток
|ИЗ
|УчетНоменклатуры КАК УчетНоменклатуры
|;
Результат=Запрос.Выполнить();

```

Первый запрос создает временную таблицу, данные из которой используются во втором запросе.

Если объекту `Запрос`, исполняющему пакетный запрос, установлен менеджер временных таблиц, временные таблицы, которые не были уничтожены в рамках пакетного запроса, сохранятся в установленном менеджере. В тексте пакетного запроса возможно использование и уничтожение временных таблиц, которые существовали в установленном менеджере временных таблиц на момент запуска пакета на исполнение.

Кроме метода `Выполнить()`, последовательно выполняющего все запросы пакета и возвращающего результат последнего запроса в пакете, платформа системы «1С:Предприятие» предоставляет еще один метод – `ВыполнитьПакет()`. Этот метод последовательно выполняет все запросы и возвращает массив результатов для каждого запроса из пакета в последовательности расположения запросов в тексте пакета. Результатом выполнения запроса на уничтожение временной таблицы является значение `Неопределено`, которое также помещается в массив результатов.

8.3.4. Программная работа с запросами

Система «1С:Предприятие» предоставляет возможность программно формировать или модифицировать запрос. Для этого существует специальный объект `СхемаЗапроса`. С помощью свойства `ПакетЗапросов` разработчик получает доступ ко всем запросам, которые в данный момент редактируются с помощью объекта `СхемаЗапроса`. Методы `УстановитьТекстЗапроса()` предназначен для заполнения объекта `СхемаЗапроса` на основании переданного текста запроса, а метод `ПолучитьТекстЗапроса()` служит для обратной операции.

В качестве примера использования объектной модели работы с запросами будет приведен программный код, необходимый для формирования следующего запроса:

```

ВЫБРАТЬ
    ТоварныеЗапасыОстатки.Товар КАК Товар,
    ТоварныеЗапасыОстатки.Склад КАК Склад,
    ТоварныеЗапасыОстатки.КоличествоОстаток КАК КоличествоОстаток,
ВЫБОР
    КОГДА ТоварныеЗапасыОстатки.КоличествоОстаток > 0
        ТОГДА "Достаточно"
    ИНАЧЕ "Недостаточно"
КОНЕЦ КАК Состояние
ИЗ
    РегистрНакопления.ТоварныеЗапасы.Остатки КАК ТоварныеЗапасыОстатки
ГДЕ
    ТоварныеЗапасыОстатки.Склад = &Склад
УПОРЯДОЧИТЬ ПО
    Товар,
    Склад,
    КоличествоОстаток
ИТОГИ
    СУММА(КоличествоОстаток)
ПО
    ОБЩИЕ,
    Товар,
    Склад

```

[Копировать в буфер обмена](#)

Код на встроенном языке:

```

Конструктор = Новый СхемаЗапроса;
Пакет1 = Конструктор.ПакетЗапросов[0];
Оператор1 = Пакет1.Операторы[0];
// добавим источник в запрос
Источник1 = Оператор1.Источники.Добавить(Конструктор.ПакетЗапросов[0].ДоступныеТаблицы.Найти("РегистрНакопления.ТоварныеЗапасы.Остатки"));
Источник1.Источник.Псевдоним = "ТоварныеЗапасыОстатки";
// добавим поля в запрос
Для каждого ДоступноеПоле Из Источник1.Источник.ДоступныеПоля Цикл
    Если ДоступноеПоле.Имя = "Товар" Тогда
        ПолеТовар = Оператор1.ВыбираемыеПоля.Добавить(ДоступноеПоле);
        Прервать;
    КонецЕсли;
КонецЦикла;
ПолеСклад = Оператор1.ВыбираемыеПоля.Добавить("ТоварныеЗапасыОстатки.Склад");
ПолеКоличество = Оператор1.ВыбираемыеПоля.Добавить("ТоварныеЗапасыОстатки.КоличествоОстаток");
ВыражениеВыбора = Новый ВыражениеСхемыЗапроса("ВЫБОР
|
|    КОГДА ТоварныеЗапасыОстатки.КоличествоОстаток > 0
|        ТОГДА "Достаточно"
|        ИНАЧЕ "Недостаточно"
|КОНЕЦ");
ПолеВыбора = Оператор1.ВыбираемыеПоля.Добавить(Строка(ВыражениеВыбора));
// теперь зададим псевдоним для оператора выбора
Для каждого Колонка Из Пакет1.Колонки Цикл
    Если VReg(Колонка.Поля[0]) = VReg(ВыражениеВыбора) Тогда
        Колонка.Псевдоним = "Состояние";
        Прервать;
    КонецЕсли;
КонецЦикла;
// зададим отбор
Оператор1.Отбор.Добавить("Склад = &Склад");
// зададим упорядочивание
Для каждого ДоступноеПоле Из Источник1.Источник.ДоступныеПоля Цикл
    Если ДоступноеПоле.Имя = "Товар" Тогда
        Пакет1.Порядок.Добавить(ДоступноеПоле);
        Прервать;
    КонецЕсли;
КонецЦикла;
Для каждого ДоступноеПоле Из Источник1.Источник.ДоступныеПоля Цикл
    Если ДоступноеПоле.Имя = "Склад" Тогда
        Пакет1.Порядок.Добавить(ДоступноеПоле);
        Прервать;
    КонецЕсли;
КонецЦикла;
Для каждого ДоступноеПоле Из Источник1.Источник.ДоступныеПоля Цикл
    Если ДоступноеПоле.Имя = "КоличествоОстаток" Тогда
        Пакет1.Порядок.Добавить(ДоступноеПоле);
        Прервать;
    КонецЕсли;
КонецЦикла;

```

[Копировать в буфер обмена](#)

1552

```
    КонечЕсли;
КонецЦикла;
// зададим контрольные точки итоги
Пакет1.ОбщиеИтоги = Истина;
Пакет1.КонтрольныеТочкиИтогов.Добавить("Товар");
Пакет1.КонтрольныеТочкиИтогов.Добавить("Склад");
// установим агрегатные функции для итогов
Для каждого КолонкаСхемы Из Пакет1.Колонки Цикл
    Если КолонкаСхемы.Псевдоним = "КоличествоОстаток" Тогда
        Пакет1.ВыраженияИтогов.Добавить(КолонкаСхемы);
        Прервать;
    КонечЕсли;
КонецЦикла;
Для каждого ВыражениеИтога Из Пакет1.ВыраженияИтогов Цикл
    Если ВыражениеИтога.Поле.Псевдоним = "КоличествоОстаток" Тогда
        ВыражениеИтога.Выражение = Новый ВыражениеСхемыЗапроса("Сумма(" + ВыражениеИтога.Поле.Псевдоним + ")");
        Прервать;
    КонечЕсли;
КонецЦикла;
Текст = Конструктор.ПолучитьТекстЗапроса();
```


Глава 9. Работа с данными

9.1. Механизм объектных блокировок

При работе с объектными данными (справочники, документы, счета и пр.) система «1С:Предприятие» обеспечивает два вида объектных блокировок – **пессимистическую** и **оптимистическую**. Они позволяют выполнять целостные изменения объектов при одновременной работе нескольких пользователей.

9.1.1. Пессимистическая блокировка

Механизм пессимистической блокировки объектов базы данных предназначен для того, чтобы запретить изменение данных объекта другими сеансами или данным сеансом до тех пор, пока блокировка не будет снята (автоматически или с помощью методов встроенного языка).

В основном механизм пессимистической блокировки используется системой «1С:Предприятие» для блокировки объектов, редактируемых в форме. В то же время разработчик имеет возможность задействовать этот механизм, используя средства встроенного языка.

Система «1С:Предприятие» использует механизм пессимистической блокировки с помощью расширений форм прикладных объектов. В тот момент, когда пользователь начинает модификацию объекта в форме, расширение формы устанавливает пессимистическую блокировку. Если после этого другой пользователь попытается выполнить редактирование того же объекта, ему будет выдано сообщение о том, что не удалось заблокировать объект. Когда пользователь, редактировавший объект, закроет форму объекта, расширение формы снимет пессимистическую блокировку.

В том случае, если необходимо в нестандартной форме объекта обеспечить такое же поведение, что и в стандартной форме объекта, можно использовать метод формы `ЗаблокироватьДанныеФормыДляРедактирования()` для установки пессимистической блокировки и метод формы `РазблокироватьДанныеФормыДляРедактирования()` для снятия блокировки.

Разработчик, для того чтобы задействовать пессимистическую блокировку, может использовать метод глобального контекста `ЗаблокироватьДанныеДляРедактирования()`. Возможны два варианта установки пессимистической блокировки:

- С указанием идентификатора формы – в этом случае блокировка устанавливается на время жизни форм, в которой установлена блокировка и снимается автоматически при закрытии формы или завершении сеанса. Также блокировка может быть снята с помощью метода глобального контекста `РазблокироватьДанныеДляРедактирования()` с указанием того же идентификатора формы, который указывался для установки блокировки.
- Без указания идентификатора формы – в этом случае устанавливаемая блокировка не привязана к какой-либо форме. Блокировка будет автоматически снята при завершении сеанса, при возврате управления с сервера или при завершении транзакции (если блокировка устанавливалась в транзакции). Также блокировка может быть снята с помощью метода глобального контекста `РазблокироватьДанныеДляРедактирования()` без указания идентификатора формы.

Однако следует учитывать, что сам по себе факт установки блокировки не препятствует изменению или удалению объекта в базе данных. Поэтому для того, чтобы обеспечить невозможность изменения заблокированного объекта, операции изменения объекта в другом сеансе также должна предшествовать попытка блокировки этого объекта. Блокировка заблокированного объекта базы данных вызывает исключение, которое может быть обработано конструкцией `Попытка ... Исключение ... КонецПопытки`.

[Копировать в буфер обмена](#)

&НаСервере

```

Функция ПримерМодификации()
    ТоварСсылка = Справочники.Товары.НайтиПоКоду("00000001");
    Попытка
        ЗаблокироватьДанныеДляРедактирования(ТоварСсылка);
        // Можно выполнять модификацию данных объекта
        // ...
        ТоварОбъект = ТоварСсылка.ПолучитьОбъект();
        ТоварОбъект.Наименование = "Новое наименование";
        ТоварОбъект.Записать();
        Возврат Истина;
    Исключение
        // Нельзя модифицировать данные объекта
        Сообщение = Новый СообщениеПользователю;
        Сообщение.Текст = "Данные объекта уже заблокированы";
        Сообщение.Сообщить();
        Возврат Ложь;
    КонецПопытки;
КонецФункции

```

Следует помнить, что попытки установить блокировку одного и того же объекта с указанием идентификатора формы и без указания идентификатора несовместимы друг с другом.

[Копировать в буфер обмена](#)

```

ТоварСсылка = Справочники.Номенклатура.НайтиПоКоду(1);
ЗаблокироватьДанныеДляРедактирования(ТоварСсылка);
Попытка
    ЗаблокироватьДанныеДляРедактирования(ТоварСсылка, , ИдентФормы);
Исключение
    // исключение из-за несовместимости блокировок
КонецПопытки;

```

Для снятия пессимистической блокировки разработчик может использовать метод глобального контекста `РазблокироватьДанныеДляРедактирования()`.

9.1.2. Пессимистическая блокировка и транзакции

Операции блокировки объектов влияют только на выполнение других операций блокировки объектов и не влияют на операции над данными и на процесс течения транзакций.

Блокировка заблокированного объекта базы данных вызывает исключение, которое может быть обработано и не приводит к обязательному откату транзакции. Если в течение транзакции при выполнении метода `ЗаблокироватьДанныеДляРедактирования()` возникло исключение, то

оно может быть обработано конструкцией `Попытка ... Исключение ... КонецПопытки` и не требует обязательного отката транзакции.

Блокировки объектов, установленные в течение транзакции, снимаются при окончании транзакции, если блокировка устанавливалась без указания идентификатора формы.

9.1.3. Оптимистическая блокировка

Оптимистическая блокировка запрещает запись объекта в базу данных, если после считывания объекта он был изменен в базе данных.

Строго говоря, оптимистическая блокировка представляет собой проверку, которая выполняется перед записью объекта в базу данных.

Когда объект встроенного языка считывает данные из базы данных, в числе прочего считывается и версия объекта, хранящегося в базе данных.

Если до начала редактирования данных пользователем (до установки пессимистической блокировки) данные объекта в базе данных были изменены (например, другим пользователем), то номер версии объекта, хранящийся в базе данных, также изменится. При попытке пользователя записать этот объект будет выполнена проверка соответствия версии объекта, находящегося в памяти, и версии объекта, хранящейся в базе данных. Так как версии отличаются, будет выдано предупреждение о том, что версия объекта изменилась или он был удален, то есть сработает оптимистическая блокировка.

Оптимистическая блокировка гарантирует, что если пользователь изменяет объект, то его изменения не «затрут» изменения, сделанные другими сеансами или другими программными объектами этого же сеанса.

9.2. Механизм транзакций

Независимо от выбранного варианта работы (файловый или клиент-серверный) система «1С:Предприятие» обеспечивает работу с информацией, хранящейся в базе данных, с использованием механизма транзакций.

Транзакция – это неделимая с точки зрения воздействия на базу данных последовательность операций манипулирования данными. Она выполняется по принципу «все или ничего» и переводит базу данных из одного целостного состояния в другое целостное состояние. Если по каким-либо причинам одно из действий транзакции невыполнимо или произошло какое-либо нарушение работы системы, база данных возвращается в то состояние, которое было до начала транзакции (происходит откат транзакции).

Транзакции могут использоваться как самой системой «1С:Предприятие», так и разработчиком при написании модулей.

Система «1С:Предприятие» осуществляет неявный вызов транзакций при выполнении любых действий, связанных с модификацией информации, хранящейся в базе данных. Например, все обработчики событий, расположенные в модулях объектов и наборов записей, связанные с модификацией данных базы данных, вызываются в транзакции. В транзакции выполняется также чтение объектов следующих типов: `ПланОбменаОбъект`, `ДокументОбъект`, `СправочникОбъект`, `ПланВидовХарактеристикОбъект`, `ПланВидовРасчетаОбъект`, `ПланСчетовОбъект`, `БизнесПроцессОбъект`, `ЗадачаОбъект`, `ПоследовательностьНаборЗаписей`, `РегистрСведенийНаборЗаписей`, `РегистрНакопленияНаборЗаписей`, `РегистрБухгалтерииНаборЗаписей`, `РегистрРасчетаНаборЗаписей`, `ПерерасчетНаборЗаписей`. При этом в режиме управляемых блокировок (см. [здесь](#)) выполняется установка разделяемой блокировки по значению регистратора для наборов записей и по значениям отбора для набора записей независимого регистра сведений.

Наряду с этим разработчик может использовать работу с транзакциями в явном виде. Для этого используются процедуры глобального контекста `НачатьТранзакцию()`, `ЗафиксироватьТранзакцию()` и `ОтменитьТранзакцию()`.

9.2.1. Использование явного вызова транзакций

Метод `НачатьТранзакцию()` позволяет открыть транзакцию. После этого все изменения информации базы данных, выполняемые последующими операторами, могут быть либо целиком приняты, либо целиком отвергнуты.

Для принятия всех выполненных изменений используется метод `ЗафиксироватьТранзакцию()`. Для того чтобы отменить все изменения, выполнявшиеся в открытой транзакции, используется метод `ОтменитьТранзакцию()`. Если количество вызовов метода `НачатьТранзакцию()` превышает количество вызовов методов `ЗафиксироватьТранзакцию()` или `ОтменитьТранзакцию()`, то система выполнит неявный вызов метода `ОтменитьТранзакцию()` в следующих случаях:

- при окончании выполнения встроенного языка (обработчик события, внешнее соединение, automation-сервер);
- при передаче управления с сервера на клиента.

Если количество вызовов методов `ЗафиксироватьТранзакцию()` или `ОтменитьТранзакцию()` превышает количество вызовов метода `НачатьТранзакцию()`, то при выполнении лишнего вызова метода `ЗафиксироватьТранзакцию()` или `ОтменитьТранзакцию()` будет порождено исключение.

Таким образом, схема работы с транзакцией в общем виде может выглядеть следующим образом:

[Копировать в буфер обмена](#)

```
Попытка
    НачатьТранзакцию();
    // Последовательность операторов
    ...
    ЗафиксироватьТранзакцию();
Исключение
    ОтменитьТранзакцию();
КонецПопытки;
```

При использовании такой схемы следует помнить о том, что не все ошибки, возникающие при работе с базой данных, обрабатываются системой одинаково.

В общем случае все ошибки базы данных можно разделить на две категории:

- невосстановимые,
- восстановимые.

Невосстановимые ошибки – это ошибки, при возникновении которых нормальное функционирование системы «1С:Предприятие» может быть нарушено, например, могут быть испорчены данные. При возникновении невозстановимой ошибки выполнение системы «1С:Предприятие» прекращается в любом случае.

Если невозстановимая ошибка произошла в процессе выполнения транзакции, то все изменения, сделанные в рамках этой транзакции, отменяются системой.

Восстановимые ошибки – это ошибки, не вызывающие серьезных нарушений в работе системы «1С:Предприятие». В случае возникновения восстановимой ошибки дальнейшая работа системы может быть продолжена. При этом, естественно, сама операция, вызвавшая ошибку, прекращается, и вызывается исключение, которое может быть перехвачено и обработано конструкцией

```
Попытка ... Исключение ... КонецПопытки.
```

Если восстановимая ошибка произошла в процессе выполнения транзакции, то система автоматически не выполняет отмену транзакции, предоставляя разработчику возможность самостоятельно обработать сложившуюся ситуацию.

В зависимости от характера произошедшей ошибки возможны различные сценарии обработки этой ситуации.

Если произошедшая ошибка не связана с базой данных, то возможно продолжение транзакции и дальнейшей работы модуля. Если разработчик считает это необходимым, он может отменить транзакцию или, наоборот, продолжить выполнение транзакции, если произошедшая ошибка не нарушает атомарность транзакции.

Если же исключительная ситуация была вызвана ошибкой базы данных, то система фиксирует факт возникновения ошибки в этой транзакции, и дальнейшее продолжение транзакции или ее фиксации становятся невозможны. Единственная операция с базой данных, которую разработчик может произвести в данной ситуации, – это отмена транзакции. После этого он может осуществить попытку выполнения этой транзакции еще раз.

Например, фрагмент кода, реализующий этот подход при записи некоторых данных в базу данных, может выглядеть следующим образом.

[Копировать в буфер обмена](#)

```
// Признак окончания попыток выполнения записи
Записано = Ложь;
// Попытки записи выполняются в цикле
Пока Не Записано Цикл
    Попытка
        НачатьТранзакцию();
        Данные.Записать();
        ЗафиксироватьТранзакцию();
        // В случае фиксации транзакции прекратить попытки записи
        Записано = Истина;
    Исключение
        // В случае неудачи отменить текущую транзакцию и
        // следующую попытку начать с новой транзакции
        ОтменитьТранзакцию();
    КонецПопытки;
КонецЦикла;
```

9.2.2. Вложенный вызов транзакций

В рамках уже выполняемой транзакции можно обращаться к процедурам `НачатьТранзакцию()`, `ЗафиксироватьТранзакцию()` и `ОтменитьТранзакцию()`. Например, может использоваться следующая схема вызовов:

[Копировать в буфер обмена](#)

```
НачатьТранзакцию();
...
// Вложенный вызов транзакции
НачатьТранзакцию();
...
ЗафиксироватьТранзакцию();
...
// Вложенный вызов транзакции
НачатьТранзакцию();
...
ЗафиксироватьТранзакцию();
...
ЗафиксироватьТранзакцию();
```

Однако подобное обращение не означает начала новой транзакции в рамках уже выполняющейся.

ВНИМАНИЕ! Система «1С:Предприятие» не поддерживает вложенных транзакций.

Это означает, что всегда действует только транзакция самого верхнего уровня. Все транзакции, вызванные внутри уже открытой транзакции, фактически относятся к той же транзакции, а не образуют вложенную транзакцию. Таким образом, отмена изменений, выполняемая во вложенной транзакции, будет приводить в конечном счете не к отмене изменений самой вложенной транзакции, а к отмене всех изменений транзакции верхнего уровня. В то же время фиксация изменений, выполненная во вложенной транзакции, игнорируется.

9.2.3. Влияние транзакций на работу программных объектов

В общем случае программные объекты, используемые системой «1С:Предприятие», абсолютно «прозрачны» для транзакций базы данных. Иначе говоря, транзакции базы данных могут вызываться при выполнении различных методов программных объектов, однако, например, действия, выполняемые базой данных при откате транзакции, в общем случае никак не влияют на соответствующие программные объекты.

Из этого следует, что при отмене транзакций базы данных разработчик (если в этом есть необходимость) должен самостоятельно обеспечивать адекватное изменение данных соответствующих программных объектов. Это можно выполнять путем повторного чтения всех данных объекта или путем изменения некоторых реквизитов программного объекта (если, например, это необходимо для отображения в интерфейсе).

В этом правиле есть исключения. В силу значительной прикладной специфики программных объектов системы «1С:Предприятие» в некоторых случаях откат изменений, выполненных в базе данных, все же может влиять на значения свойств соответствующих программных объектов. Это происходит в следующих случаях:

- при отмене транзакции признак проведения документа восстанавливает значение, которое было до начала транзакции;
- если объект был создан и записан в транзакции, то при откате транзакции очищается значение ссылки;
- если объект создавался вне транзакции и при записи его в транзакции использовался код/номер, сгенерированный автоматически, то при отмене транзакции код/номер очищается.

9.3. Механизм управляемых блокировок

9.3.1. Общие сведения о блокировках

В идеальном случае в любой СУБД транзакции должны обеспечивать изоляцию изменений, выполняемых в базе данных. Иными словами, несколько транзакций, выполняющих изменение данных, не должны мешать друг другу.

Самым простым способом решения этой проблемы является последовательное выполнение транзакций. Следующая транзакция выполняется после того, как закончилась предыдущая.

Однако в реальной ситуации, при многопользовательской работе, такой подход приводит к резкому снижению производительности системы. Поэтому на практике используются механизмы, позволяющие выполнять несколько транзакций одновременно.

Для того чтобы одновременное выполнение транзакций стало возможным, используется несколько уровней изоляции транзакций. На самом низком уровне изоляции транзакции могут сильно влиять друг на друга. На самом высоком уровне они полностью изолированы.

Таким образом, за большую изоляцию транзакций приходится платить большими накладными расходами и замедлением работы системы.

Возможность изоляции одних транзакций от других обычно реализуется благодаря блокировкам, накладываемым на используемые ими данные. В зависимости от уровня изоляции накладываются различные типы блокировок на различные объекты базы данных, на различное время.

С точки зрения системы «1С:Предприятие» работа с данными может выполняться в одном из двух режимов:

- в транзакции,
- вне транзакции.

Режим работы с данными **вне транзакции** допускает только операции **чтения данных**. Этот режим введен для того, чтобы обеспечить максимальную скорость и параллельность чтения данных. Поэтому любая операция чтения данных, выполняемая вне транзакции, считается **безответственной**. Это означает, что такая операция чтения может вернуть устаревшие данные или даже незафиксированные изменения, произведенные другой транзакцией, т. е. чтение выполняется «не глядя» на блокировки данных, расставленные другими транзакциями.

Режим работы с данными в транзакции допускает любые операции чтения и модификации данных.

При этом должны соблюдаться следующие правила:

- Чтение данных должно быть воспроизводимым, т. е. любая последующая операция чтения данных с тем же условием выборки должна возвращать такой же результат.
- Результат чтения должен содержать наиболее актуальные данные. Это означает, что никакая другая транзакция не может изменить данные, считанные в данной транзакции, до тех пор, пока данная транзакция не будет завершена.
- Результат чтения не должен содержать незафиксированные изменения данных базы данных.

9.3.2. Управляемые блокировки

Система «1С:Предприятие» позволяет использовать два режима работы с базой данных в рамках транзакции: режим автоматических блокировок и режим управляемых блокировок.

Принципиальное отличие этих режимов заключается в следующем. Режим автоматических блокировок не требует от разработчика каких-либо действий по управлению блокировками в транзакции для того, чтобы обеспечивались перечисленные выше правила работы с данными в транзакции. Эти правила обеспечиваются платформой системы «1С:Предприятие» за счет использования определенных уровней изоляции транзакций в той или иной СУБД. Такой режим работы является наиболее простым для разработчика, однако в некоторых случаях (например, при интенсивной одновременной работе большого количества пользователей) используемый уровень изоляции транзакций в СУБД не может обеспечить достаточной параллельности работы, что проявляется в виде большого количества конфликтов блокировок при работе пользователей.

При работе в режиме управляемых блокировок система «1С:Предприятие» использует гораздо более низкий уровень изоляции транзакций в СУБД, что позволяет значительно повысить параллельность работы пользователей прикладного решения. Однако, в отличие от режима автоматических блокировок, данный уровень изоляции транзакций уже не может сам по себе обеспечить выполнение всех правил работы с данными в транзакции (в частности, не обеспечивается воспроизводимость чтения данных в транзакции). Поэтому при работе в управляемом режиме от разработчика требуется самостоятельно управлять блокировками, устанавливаемыми в транзакции.

В сводном виде отличия при работе в режиме автоматических блокировок и в режиме управляемых блокировок приведены в следующей таблице:

	Вид блокировки	Уровень изоляции транзакций
Автоматические блокировки		
Файловая БД	Таблиц	Serializable
MS SQL Server	Записей	Repeatable Read или Serializable
IBM DB2	Записей	Repeatable Read или Serializable
PostgreSQL	Таблиц	Serializable
Oracle Database	Таблиц	Serializable

Управляемые блокировки		
Файловая БД	Таблиц	Serializable
MS SQL Server 2000	Записей	Read Committed
MS SQL Server 2005 и выше	Записей	Read Committed Snapshot
IBM DB2	Записей	Read Committed
PostgreSQL	Записей	Read Committed
Oracle Database	Записей	Read Committed

ПРИМЕЧАНИЕ. Рекомендуется разработку прикладных решений вести в режиме управляемых блокировок с целью максимального использования возможностей СУБД. Использование прикладного решения в автоматическом режиме используется для совместимости с предыдущими версиями прикладных решений и не рекомендуется для реальной эксплуатации.

Поведение системы при безответственном чтении (например, при работе динамических списков, выполнении отчетов) также отличается в различных режимах блокировок. В сводном виде отличия безответственного чтения в разных режимах работы блокировок (автоматическом или управляемом) приведены в следующей таблице:

	Чтение вне транзакции
Автоматические блокировки	
Файловая БД	«Грязное» чтение
MS SQL Server	«Грязное» чтение
IBM DB2	«Грязное» чтение
PostgreSQL	Согласованное чтение
Oracle Database	Согласованное чтение
Управляемые блокировки	
Файловая БД	«Грязное» чтение
MS SQL Server 2000	«Грязное» чтение
MS SQL Server 2005 и выше	Согласованное чтение
IBM DB2 до версии 9.7	«Грязное» чтение
IBM DB2 версии 9.7 и выше	Согласованное чтение
PostgreSQL	Согласованное чтение
Oracle Database	Согласованное чтение

9.3.3. Установка режима блокировок в конфигурации

Конфигурация имеет свойство [Режим управления блокировкой данных](#). Каждый прикладной объект конфигурации также имеет свойство [Режим управления блокировкой данных](#).

Режим управления блокировкой данных для всей конфигурации в целом может быть установлен в значения [Автоматический](#), [Управляемый](#) (установлено по умолчанию для новой конфигурации) и [Автоматический и управляемый](#). Значения [Автоматический](#) и [Управляемый](#) означают, что соответствующий режим блокировки будет использоваться для всех объектов конфигурации, независимо от значений, установленных для каждого из объектов. Значение [Автоматический и управляемый](#) означает, что для конкретного объекта конфигурации будет использован тот режим, который указан в его свойстве [Режим управления блокировкой данных](#): [Автоматический](#) или [Управляемый](#).

Следует отметить, что режим управления блокировкой данных, указанный для объекта метаданных, устанавливается для тех транзакций, которые инициируются системой «1С:Предприятие» при работе с данными этого объекта (например, при модификации данных объекта).

Если же, например, операция записи объекта выполняется в транзакции, инициированной разработчиком (метод `НачатьТранзакцию()`), то режим управления блокировкой данных будет определяться значением параметра [Режим блокировок](#) метода `НачатьТранзакцию()`, а не значением свойства объекта метаданных [Режим управления блокировкой данных](#).

По умолчанию параметр [Режим блокировок](#) имеет значение [РежимУправленияБлокировкойДанных.Автоматический](#), поэтому для того, чтобы в явной транзакции использовать режим управляемых блокировок, следует указывать значение этого параметра [РежимУправленияБлокировкойДанных.Управляемый](#).

9.3.4. Работа с управляемыми блокировками средствами встроенного языка

Для управления блокировками в транзакции предназначен объект встроенного языка [БлокировкаДанных](#). Экземпляр этого объекта может быть создан с помощью конструктора и позволяет описать необходимые пространства блокировок и режимы блокировок. Для установки всех созданных блокировок используется метод `Заблокировать()` объекта [БлокировкаДанных](#). Если этот метод выполняется в транзакции (явной или неявной), блокировки устанавливаются и при окончании транзакции будут сняты автоматически. Если метод `Заблокировать()` выполняется вне транзакции или в автоматическом режиме управления блокировками в рамках транзакции, то будет сгенерировано исключение.

Объект [БлокировкаДанных](#) представляет собой коллекцию элементов блокировки данных, каждый из которых описывает блокировки одного пространства блокировок. Пространства блокировок определены в платформе системы «1С:Предприятие» и соответствуют структуре прикладных объектов конфигурации. Для каждого пространства блокировок в платформе определены имена полей, значения которых могут анализироваться при установке тех или иных блокировок.

Допустимы следующие имена пространств блокировок и имена полей пространств блокировок:

Имя пространства блокировки	Имя поля пространства блокировки
Справочник . <имя>	Ссылка; <имя поля>
Документ . <имя>	Ссылка; <имя поля>
ПланОбмена . <имя>	Ссылка; <имя поля>
ПланСчетов . <имя>	Ссылка; <имя поля>
БизнесПроцесс . <имя>	Ссылка; <имя поля>
Задача . <имя>	Ссылка; <имя поля>
ПланВидовРасчета . <имя>	Ссылка; <имя поля>
ПланВидовХарактеристик . <имя>	Ссылка; <имя поля>
РегистрСведений . <имя> . НаборЗаписей – только для регистра сведений, подчиненного регистратору	Регистратор
РегистрСведений . <имя>	Период – если есть; <имя измерения>
РегистрНакопления . <имя> . НаборЗаписей	Регистратор
РегистрНакопления . <имя>	Период; <имя измерения>
РегистрБухгалтерии . <имя> . НаборЗаписей	Регистратор
РегистрБухгалтерии . <имя>	Период; <имя измерения>; <вид движения> – значение системного перечисления ВидДвиженияБухгалтерии; Счет; Субконто<N>; <вид субконто>.
РегистрРасчета . <имя> . НаборЗаписей	Регистратор
РегистрРасчета . <имя>	ПериодРегистрации; ПериодДействия; <имя измерения>
Перерасчет . <имя> . НаборЗаписей	ОбъектПерерасчета
Последовательность . <имя> . НаборЗаписей	Регистратор
Последовательность . <имя>	<имя измерения>
Константа . <имя>	
ВнешнийИсточникДанных . <имя источника> . Таблица . <имя таблицы>	Ссылка; <имя поля>

Где <имя поля> – имя поля, по которому может быть установлена управляемая блокировка. Перечень допустимых полей (по которым может быть установлена управляемая блокировка) задается в свойстве **Поля блокировки данных** для следующих объектов:

- справочники,
- документы,
- планы видов характеристик,
- планы счетов,

- планы видов расчета,
- планы обмена,
- бизнес-процессы,
- задачи;
- таблицы внешних источников данных.

Запрещено устанавливать блокировки по реквизитам (и общим реквизитам) следующих типов: строка неограниченной длины, хранилище значения, тип значения плана видов характеристик, составной тип, включающий один из вышеперечисленных типов. Запрет относится к следующим объектам: справочники, документы, планы видов характеристик, планы видов расчета, планы счетов, бизнес-процессы, задачи, планы обмена. Также запрещено устанавливать блокировки по полям `Предопределенный` и `ИмяПредопределенныхДанных` в справочниках, планах видов характеристик, планах видов расчета, планах счетов.

При установке управляемой транзакционной блокировки на пространство `РегистрБухгалтерии.<Имя>`, если значение субконто задано ссылкой на характеристику, то в плане счетов эта характеристика должна быть одним из субконто этого счета. Если соответствующее субконто у счета не задано, то будет выдана ошибка.

Для каждого пространства блокировки может быть задано произвольное количество условий на поля, по которым будут определяться записи, подлежащие блокировке. Условия задаются на равенство значения поля указанному значению или на вхождение значения поля в указанный диапазон. Условия могут быть заданы двумя способами:

- с помощью явного указания имени поля и значения (метод `УстановитьЗначение()` объекта `ЭлементБлокировкиДанных`);
- с помощью указания источника данных, содержащего необходимые значения (свойство `ИсточникДанных` объекта `ЭлементБлокировкиДанных`).

При явном указании значения поля в параметры метода `УстановитьЗначение()` передается имя поля и значение:

[Копировать в буфер обмена](#)

```
Блокировка = Новый БлокировкаДанных;
ЭлементБлокировки = Блокировка.Добавить("Справочник.Магазин");
ЭлементБлокировки.УстановитьЗначение("Код", 100);
ЭлементБлокировки.Режим = РежимБлокировкиДанных.Исключительный;
Блокировка.Заблокировать();
Блокировка = Новый БлокировкаДанных;
ЭлементБлокировки = Блокировка.Добавить("РегистрНакопления.ТоварыНаСкладах");
ЭлементБлокировки.УстановитьЗначение("Качество", Справочники.Качество.НайтиПоКоду("1"));
```

Следует иметь в виду, что одну и ту же запись регистра можно заблокировать дважды: первый раз блокируя саму запись, а второй раз – блокируя набор, в который эта запись входит.

При удалении или изменении объекта, для которого доступна блокировка по полям, выполняется блокировка по полю `Ссылка` и по всем полям, указанным в свойстве `Поля блокировки данных`. При выполнении изменения объект блокируется как по «старым» значениям полей (которые были до начала записи), так и по «новым» значениям (которые находятся в записываемом объекте).

В качестве значения может быть передан также объект встроенного языка `Диапазон`, который создается с помощью конструктора и позволяет задать верхнюю и нижнюю границы диапазона (границы диапазона включаются в диапазон).

При использовании источника данных устанавливается значение свойства `ИсточникДанных`, а затем с помощью метода `ИспользоватьИсточникДанных()` задается соответствие полей области блокировки полям источника данных:

[Копировать в буфер обмена](#)

```
Блокировка = Новый БлокировкаДанных;
ЭлементБлокировки = Блокировка.Добавить("РегистрНакопления.ТоварыНаСкладах");
ЭлементБлокировки.ИсточникДанных = ДокументОбъект.ВозвратнаяТара;
ЭлементБлокировки.ИспользоватьИсточникДанных("Номенклатура", "Номенклатура");
ЭлементБлокировки.ИспользоватьИсточникДанных("Склад", "Склад");
```

В качестве источника данных может выступать:

- результат запроса,
- табличная часть,
- набор записей,
- таблица значений.

Соответственно, при установке соответствия полей именами полей источника будут являться:

- имена колонок результата запроса,
- имена реквизитов табличной части,
- имена измерений,
- имена колонок таблицы значений.

Объект `Диапазон` также может являться значением поля источника данных.

Для каждого элемента блокировки может быть задан один из двух режимов блокировки:

- разделяемый,

- исключительный.

Режим блокировки задается с помощью свойства `Режим объекта ЭлементБлокировкиДанных`.

Таблица совместимости управляемых блокировок выглядит следующим образом:

	Разделяемая	Исключительная
Разделяемая	+	-
Исключительная	-	-

Разделяемый режим блокировки подразумевает, что заблокированные данные не могут быть изменены другой транзакцией до окончания текущей транзакции.

Исключительный режим блокировки подразумевает, что заблокированные данные не могут быть изменены другой транзакцией до окончания текущей транзакции, а также не могут быть прочитаны другой транзакцией, устанавливающей разделяемую блокировку на эти данные.

Говоря о совместимости действий, выполняемых в транзакции, следует учитывать, что помимо явных управляемых блокировок, устанавливаемых пользователем, система «1С:Предприятие» устанавливает также неявные управляемые исключительные блокировки при записи данных в транзакции. При чтении данных объекта из базы данных (получении объекта и обращении к ссылке) не выполняется транзакционная блокировка объекта. Если блокировка необходима, то ее нужно устанавливать средствами языка до обращения к объекту.

Таким образом, таблица совместимости действий, выполняемых в транзакции при использовании режима управляемых блокировок, выглядит следующим образом.

Блокировка	Режим	ББ		РБ		ИБ	
		R	W	R	W	R	W
ББ	R	+	+	+	+	+	+
	W	+	-	-	-	-	-
РБ	R	+	-	+	-	-	-
	W	+	-	-	-	-	-
ИБ	R	+	-	-	-	-	-
	W	+	-	-	-	-	-

Где:

- R – чтение,
- W – запись,
- ББ – без блокировки,
- РБ – разделяемая блокировка,
- ИБ – исключительная блокировка.

9.3.5. Особенности работы в режиме «Автоматический и управляемый»

Режим управления блокировками `Автоматический и управляемый` предназначен для решения проблем, возникающих при параллельной работе с отдельными объектами конфигурации.

Например, существует большая конфигурация, которую сложно в один момент перевести в режим управляемых блокировок, однако есть один или два документа, являющиеся «камнем преткновения» при одновременной работе с ними большого количества пользователей. В этом случае вся конфигурация может быть переведена в режим `Автоматический и управляемый`, а затем сам документ и затрагиваемые при его записи объекты конфигурации – в режим `Управляемый`. Это позволит настроить работу с данным документом в режиме управляемых блокировок, в то время как основная часть конфигурации будет продолжать функционировать в автоматическом режиме управления блокировками.

При работе в режиме управления блокировками `Автоматический и управляемый` следует учитывать две особенности:

- Независимо от режима, указанного для данной транзакции, система будет устанавливать соответствующие управляемые блокировки.
- Режим управления блокировками определяется транзакцией самого «верхнего» уровня. Другими словами, если к моменту начала транзакции была начата другая транзакция, то начинаемая транзакция может быть выполнена только в том режиме, который установлен для уже выполняющейся транзакции.

Рассмотрим перечисленные особенности более подробно.

Первая особенность заключается в том, что даже если для транзакции используется автоматический режим управления блокировками, система установит дополнительно и соответствующие управляемые блокировки при записи данных в этой транзакции. Из этого следует, что транзакции, исполняющиеся в режиме управляемых блокировок, могут конфликтовать с транзакциями, исполняющимися в режиме автоматического управления блокировками.

Вторая особенность заключается в том, что режим управления блокировками, указываемый для объекта метаданных в конфигурации или указываемый при начале транзакции в явном виде (как параметр метода `НачатьТранзакцию()`), является лишь «желаемым» режимом. Фактический режим управления блокировками, в котором будет исполняться транзакция, зависит от того, является ли данный вызов начала транзакции первым, или к этому моменту уже начата другая транзакция в данной сессии системы «1С:Предприятие».

Например, если требуется управлять блокировками при записи наборов записей регистра, при проведении документа, то управляемый режим блокировок должен быть установлен как для самого регистра, так и для документа, поскольку запись наборов записей регистра будет

выполняться в транзакции, открытой при записи документа.

Возможны четыре различных сочетания режимов управления блокировками в транзакции, которые представлены в следующей таблице:

Режим существующей транзакции	Режим начинаемой транзакции	Результат
Автоматический	Автоматический	Начинаемая транзакция будет выполнена в автоматическом режиме
Автоматический	Управляемый	Начинаемая транзакция будет выполнена в автоматическом режиме
Управляемый	Автоматический	Будет вызвана исключительная ситуация
Управляемый	Управляемый	Начинаемая транзакция будет выполнена в управляемом режиме

9.3.6. Особенности работы с большим количеством управляемых блокировок

Следует помнить, что если на одно пространство накладывается более 100 000 блокировок, то может произойти эскалация блокировки. При эскалации блокировки блокируется все пространство. При использовании независимых разделителей эскалация происходит для одного набора значений разделителей:

- Блокируется все пространство только в рамках значений разделителей;
- На сеансы с другими значениями разделителей эскалация влияния не оказывает.

9.3.7. Модификация конфигураций при переходе к режиму управляемых блокировок

При переходе к частичной или полной работе в режиме управляемых блокировок прикладное решение требует доработки. Доработка заключается в том, что необходимо выявить участки кода, которые требуют наличия управляемых блокировок, и затем в этих участках кода установить требуемый режим управляемой блокировки данных. Это можно осуществить двумя способами:

- с помощью объекта `БлокировкаДанных`;
- с помощью свойства `БлокироватьДляИзменения` у наборов записей регистров накопления и регистров бухгалтерии. Необходимая управляемая блокировка будет автоматически установлена платформой в том случае, если записывается набор записей регистра, у которого данное свойство имеет значение `Истина`.

9.3.7.1. Определение участков кода, требующих доработки

Управляемые блокировки необходимо устанавливать в тех участках кода, которые удовлетворяют одному из следующих условий:

- Считываются данные, которые в дальнейшем должны быть изменены.
- Считывается некоторая согласованная совокупность данных (содержащихся в нескольких объектах), и согласованность считанных данных необходимо поддержать. В этом случае данные из этой совокупности должны быть заблокированы либо одновременно перед считыванием, либо по мере считывания отдельных элементов. Если некоторый элемент данных считывается однократно, не связан логически с другими элементами данных и при этом не будет изменен в той же транзакции, то его можно не блокировать.
- Важно обеспечить неизменность считываемых данных до конца транзакции. Например, некоторые данные, считываемые в транзакции, могут быть прочитаны еще раз, и важно обеспечить их неизменность.

Типичным примером таких операций может служить запрос к остаткам номенклатуры в модуле проведения документа.

9.3.7.2. Выбор режима управляемой блокировки

Для операций, описанных выше, режим блокировки следует устанавливать исходя из следующих соображений:

- **Исключительный** режим блокировки следует устанавливать для тех данных, которые должны быть изменены в рамках этой же транзакции. Это позволит предотвратить конфликты блокировок.
- **Разделяемый** режим блокировки следует устанавливать для тех данных, которые только считываются и изменять которые не предполагается, но заблокировать от изменений все-таки требуется.

9.3.7.3. Примеры доработки конфигурации

Далее приведем пример установки управляемых блокировок в модуле набора записей регистра накопления `ТоварыНаСкладах`.

Прежде всего, следует создать управляемую блокировку:

```
Блокировка = Новый БлокировкаДанных;
```

[Копировать в буфер обмена](#)

Затем следует проанализировать текст запроса, формируемого в модуле, и создать необходимые блокировки.

Исключительная блокировка на регистр «ТоварыНаСкладах»

В ходе выполнения модуля набора записей формируется запрос, содержащий следующий фрагмент:

```
// ...
ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.ТоварыНаСкладах.Остатки(,
    Номенклатура В (ВЫБРАТЬ РАЗЛИЧНЫЕ
        Номенклатура
        ИЗ
        Документ.РеализацияТоваровУслуг.Товары
```

[Копировать в буфер обмена](#)

```

ГДЕ
Ссылка = &ДокументСсылка))
// ...

```

В данном случае нужно следует установить следующие блокировки:

[Копировать в буфер обмена](#)

```

БлокировкаТоварыНаСкладах1 = Блокировка.
    Добавить ("РегистрНакопления.ТоварыНаСкладах");
БлокировкаТоварыНаСкладах1.Режим = РежимБлокировкиДанных.Исключительный;
БлокировкаТоварыНаСкладах1.ИсточникДанных = ДокументОбъект.Товары;
БлокировкаТоварыНаСкладах1.ИспользоватьИзИсточникаДанных ("Номенклатура", "Номенклатура");
БлокировкаТоварыНаСкладах1.ИспользоватьИзИсточникаДанных ("ХарактеристикаНоменклатуры", "ХарактеристикаНоменклатуры");
БлокировкаТоварыНаСкладах1.ИспользоватьИзИсточникаДанных ("Склад", "Склад");

```

Разделяемая блокировка на регистр «ТоварыВРезервеНаСкладах»

В ходе выполнения модуля формируется запрос, содержащий следующий фрагмент:

[Копировать в буфер обмена](#)

```

// ...
ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.ТоварыВРезервеНаСкладах.Остатки(,
    Номенклатура В (ВЫБРАТЬ РАЗЛИЧНЫЕ
        Номенклатура
        ИЗ
        Документ.РеализацияТоваровУслуг.Товары
        ГДЕ
        Ссылка = &ДокументСсылка))
// ...

```

В данном случае нужно установить следующие блокировки:

[Копировать в буфер обмена](#)

```

БлокировкаТоварыВРезервеНаСкладах1 = Блокировка.Добавить ("РегистрНакопления.ТоварыВРезервеНаСкладах");
БлокировкаТоварыВРезервеНаСкладах1.Режим = РежимБлокировкиДанных.Разделяемый;
БлокировкаТоварыВРезервеНаСкладах1.ИсточникДанных = ДокументОбъект.Товары;
БлокировкаТоварыВРезервеНаСкладах1.ИспользоватьИзИсточникаДанных ("Номенклатура", "Номенклатура");
БлокировкаТоварыВРезервеНаСкладах1.ИспользоватьИзИсточникаДанных ("ХарактеристикаНоменклатуры", "ХарактеристикаНоменклатуры");
БлокировкаТоварыВРезервеНаСкладах1.ИспользоватьИзИсточникаДанных ("Склад", "Склад");

```

Разделяемая блокировка на регистр «ТоварыКПередачеСоСкладов»

В ходе выполнения модуля формируется запрос, содержащий следующий фрагмент:

[Копировать в буфер обмена](#)

```

// ...
ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.ТоварыКПередачеСоСкладов.Остатки(,
    Номенклатура В (ВЫБРАТЬ РАЗЛИЧНЫЕ
        Номенклатура
        ИЗ
        Документ.РеализацияТоваровУслуг.Товары
        ГДЕ
        Ссылка = &ДокументСсылка))
// ...

```

В данном случае нужно установить следующие блокировки:

[Копировать в буфер обмена](#)

```

БлокировкаТоварыКПередачеСоСкладов1 = Блокировка.Добавить ("РегистрНакопления.ТоварыКПередачеСоСкладов");
БлокировкаТоварыКПередачеСоСкладов1.Режим = РежимБлокировкиДанных.Разделяемый;
БлокировкаТоварыКПередачеСоСкладов1.ИсточникДанных = ДокументОбъект.Товары;
БлокировкаТоварыКПередачеСоСкладов1.ИспользоватьИзИсточникаДанных ("Номенклатура", "Номенклатура");
БлокировкаТоварыКПередачеСоСкладов1.ИспользоватьИзИсточникаДанных ("ХарактеристикаНоменклатуры", "ХарактеристикаНоменклатуры");
БлокировкаТоварыКПередачеСоСкладов1.ИспользоватьИзИсточникаДанных ("Склад", "Склад");

```

Блокировки по табличной части «ВозвратнаяТара». Исключительная блокировка на регистр «ТоварыНаСкладах»

В ходе выполнения модуля формируется запрос, содержащий следующий фрагмент:

[Копировать в буфер обмена](#)

```

// ...
ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.ТоварыНаСкладах.Остатки(,
    Номенклатура В (ВЫБРАТЬ РАЗЛИЧНЫЕ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара.Номенклатура
        ИЗ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара
        ГДЕ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара.Ссылка =
            &ДокументСсылка)
        И Качество = &Новый)
// ...

```

В данном случае нужно установить следующие блокировки:

[Копировать в буфер обмена](#)

```

БлокировкаТоварыНаСкладах2 = Блокировка.Добавить ("РегистрНакопления.ТоварыНаСкладах");
БлокировкаТоварыНаСкладах2.Режим = РежимБлокировкиДанных.Исключительный;
БлокировкаТоварыНаСкладах2.ИсточникДанных = ДокументОбъект.ВозвратнаяТара;
БлокировкаТоварыНаСкладах2.ИспользоватьИзИсточникаДанных ("Номенклатура", "Номенклатура");

```

```
БлокировкаТоварыНаСкладах2.ИспользоватьИсИсточникаДанных("Склад", "Склад");
БлокировкаТоварыНаСкладах2.УстановитьЗначение("Качество", Справочники.Качество.НайтиПоКоду("1"));
```

Блокировки по табличной части «ВозвратнаяТара». Разделяемая блокировка на регистр «ТоварыВРезервеНаСкладах»

В ходе выполнения модуля формируется запрос, содержащий следующий фрагмент:

```
// ...
ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.ТоварыВРезервеНаСкладах.Остатки(,
    Номенклатура В (ВЫБРАТЬ РАЗЛИЧНЫЕ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара.Номенклатура
        ИЗ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара
        ГДЕ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара.Ссылка =
            &ДокументСсылка) КАК Резервы
// ...
```

[Копировать в буфер обмена](#)

В данном случае нужно установить следующие блокировки:

```
БлокировкаТоварыВРезервеНаСкладах2 = Блокировка.Добавить("РегистрНакопления.ТоварыВРезервеНаСкладах");
БлокировкаТоварыВРезервеНаСкладах2.Режим = РежимБлокировкиДанных.Разделяемый;
БлокировкаТоварыВРезервеНаСкладах2.ИсточникДанных = ДокументОбъект.ВозвратнаяТара;
БлокировкаТоварыВРезервеНаСкладах2.ИспользоватьИсИсточникаДанных("Номенклатура", "Номенклатура");
БлокировкаТоварыВРезервеНаСкладах2.ИспользоватьИсИсточникаДанных("Склад", "Склад");
```

[Копировать в буфер обмена](#)

Блокировки по табличной части «ВозвратнаяТара». Разделяемая блокировка на регистр «ТоварыКПередачеСоСкладов»

В ходе выполнения модуля формируется запрос, содержащий следующий фрагмент:

```
// ...
ЛЕВОЕ СОЕДИНЕНИЕ
РегистрНакопления.ТоварыКПередачеСоСкладов.Остатки(,
    Номенклатура В (ВЫБРАТЬ РАЗЛИЧНЫЕ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара.Номенклатура
        ИЗ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара
        ГДЕ
        Документ.РеализацияТоваровУслуг.ВозвратнаяТара.Ссылка =
            &ДокументСсылка) КАК ТоварыКПередаче
// ...
```

[Копировать в буфер обмена](#)

В данном случае нужно установить следующие блокировки:

```
БлокировкаТоварыКПередачеСоСкладов2 = Блокировка.Добавить("РегистрНакопления.ТоварыКПередачеСоСкладов");
БлокировкаТоварыКПередачеСоСкладов2.Режим = РежимБлокировкиДанных.Разделяемый;
БлокировкаТоварыКПередачеСоСкладов2.ИсточникДанных = ДокументОбъект.ВозвратнаяТара;
БлокировкаТоварыКПередачеСоСкладов2.ИспользоватьИсИсточникаДанных("Номенклатура", "Номенклатура");
БлокировкаТоварыКПередачеСоСкладов2.ИспользоватьИсИсточникаДанных("Склад", "Склад");
```

[Копировать в буфер обмена](#)

После того как все необходимые блокировки созданы, следует заблокировать перечисленные данные:

```
Блокировка.Заблокировать();
// ... далее следует прежний текст модуля
```

[Копировать в буфер обмена](#)

9.4. Монопольный режим

Монопольный режим – специальный режим доступа к базе данных, при использовании которого работать с базой данных можно, используя только один сеанс. Монопольный режим можно использовать в тех случаях, когда требуется выполнить существенные, согласованные, изменения с информационной базой, которые нет необходимости выполнять в рамках транзакции. При этом желательно исключить влияние других сеансов на результаты изменения. Установку монопольного режима требуют некоторые методы встроенного языка (например, метод `УдалитьОбъекты()`), а также работа Конфигуратора.

Для установки или снятия монопольного режима необходимо использовать метод глобального контекста `УстановитьМонопольныйРежим()`. Проверить установку монопольного режима можно с помощью метода `МонопольныйРежим()`. Невозможно изменить состояние монопольного режима в том случае, если активна транзакция (явная или неявная).

Сеанс может установить монопольный режим, если нет других сеансов работы с этой информационной базой. В монопольном режиме запрещено создание новых сеансов с данной информационной базой за исключением запуска одного фонового задания. Если из сеанса, который установил монопольный режим, будет запущено фоновое задание, то запущенное фоновое задание «отнимет» монопольный режим у родительского сеанса. Во время работы фонового задания родительский сеанс не имеет возможности изменять данные в информационной базе. Монопольный режим «вернется» к родительскому сеансу после того, как запущенное фоновое задание будет завершено.

В случае работы в монопольном режиме не устанавливаются управляемые блокировки (попытки их установки игнорируются).

Приложение 1. Конвертация информационных баз системы «1С:Предприятие 7.7»

Необходимо, чтобы на компьютере, помимо «1С:Предприятия», было установлено «1С:Предприятие 7.7» (релиз не ниже 7.70.021).

Для запуска конвертера нужно запустить программу [V7Cnv.exe](#).

Если на компьютере установлено более одного релиза «1С:Предприятия 7.7», то следует быть уверенным, что последним из них запускался релиз версии не ниже 7.70.021; в противном случае при запуске конвертера будет выдано сообщение о несоответствии релиза и работа программы будет завершена.

На экран выводится окно конвертера.

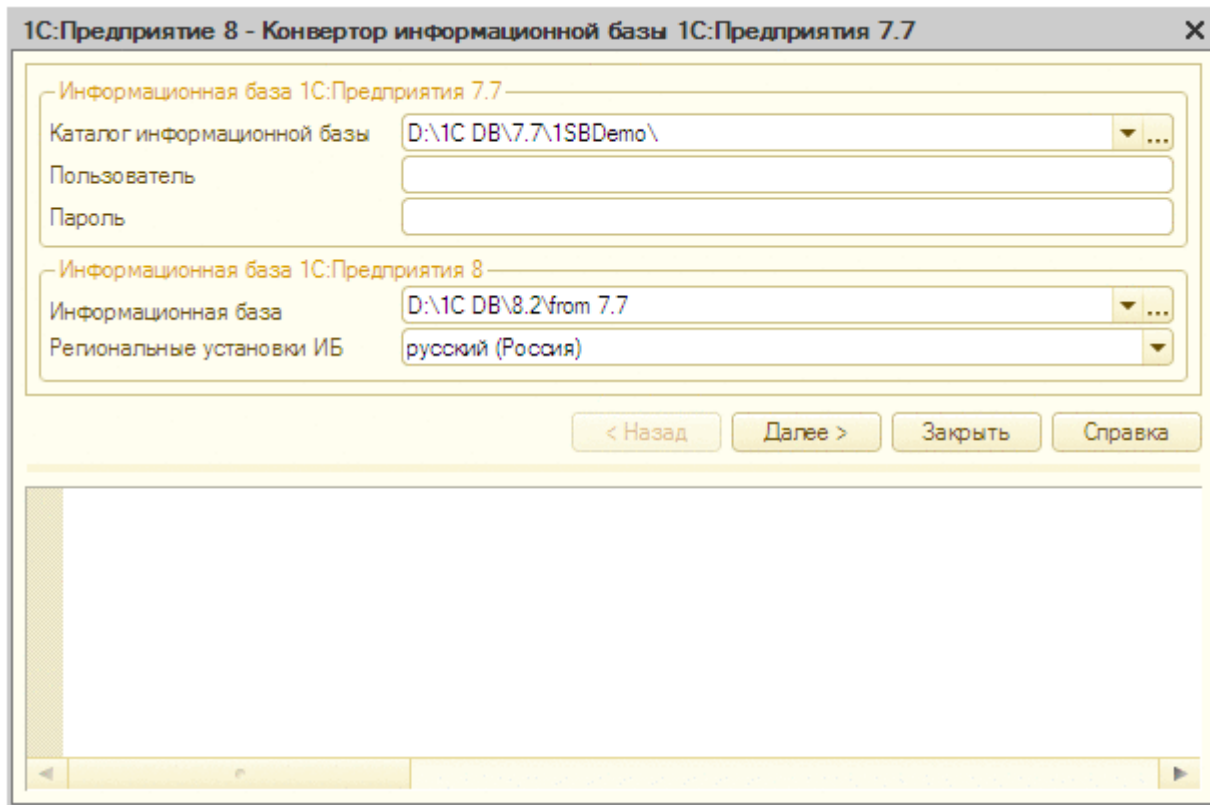


Рис. 573. Окно конвертера

Настройка конвертера производится по шагам. На первом шаге требуется указать параметры информационной базы, которую нужно конвертировать, а также параметры результирующей информационной базы. При этом если необходимо конвертировать информационную базу в базу формата клиент-сервер, в строке [Каталог информационной базы](#) следует указать строку соединения с базой на сервере. Не конвертируются базовые конфигурации и конфигурации, в которых установлен пароль на конфигурацию.

На втором шаге выполняется настройка состава конвертации.

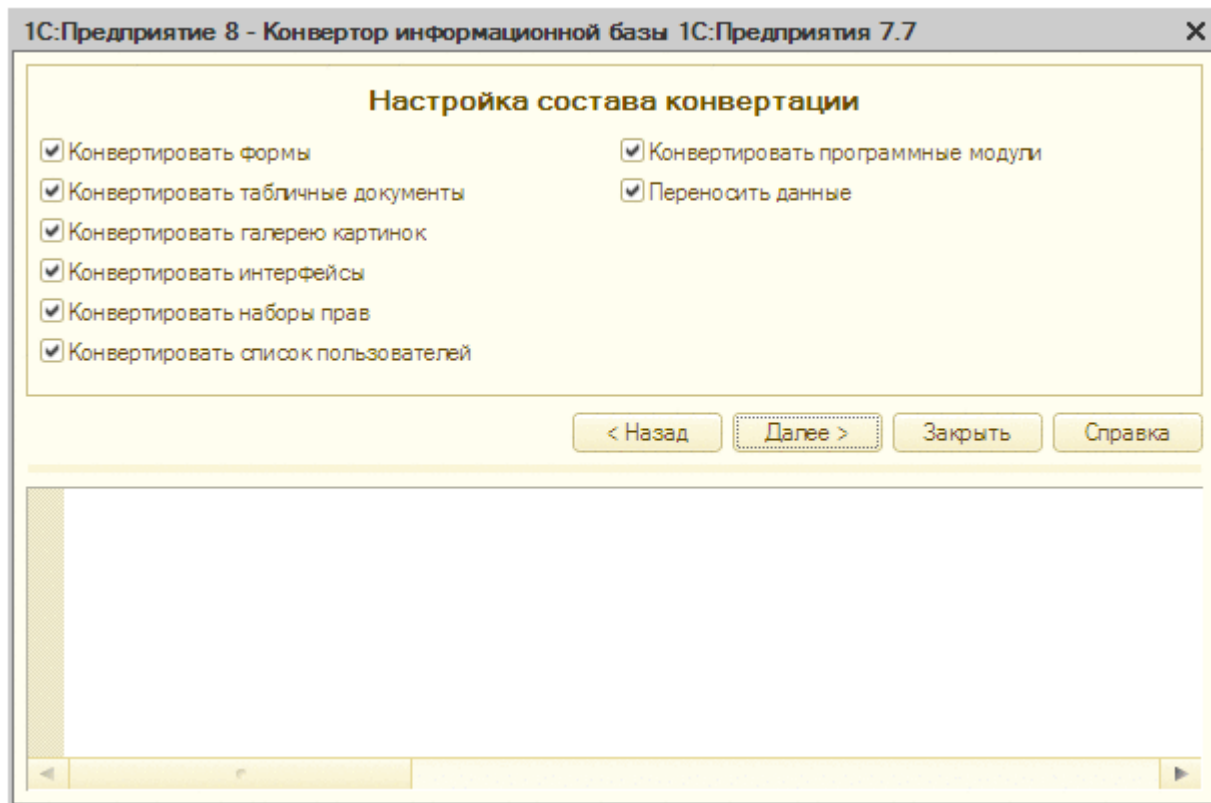


Рис. 574. Настройка состава конвертации

Следует установить флажки для тех действий, которые требуется выполнить.

На третьем шаге для начала выполнения анализа исходной информационной базы нужно нажать кнопку **Выполнить**.

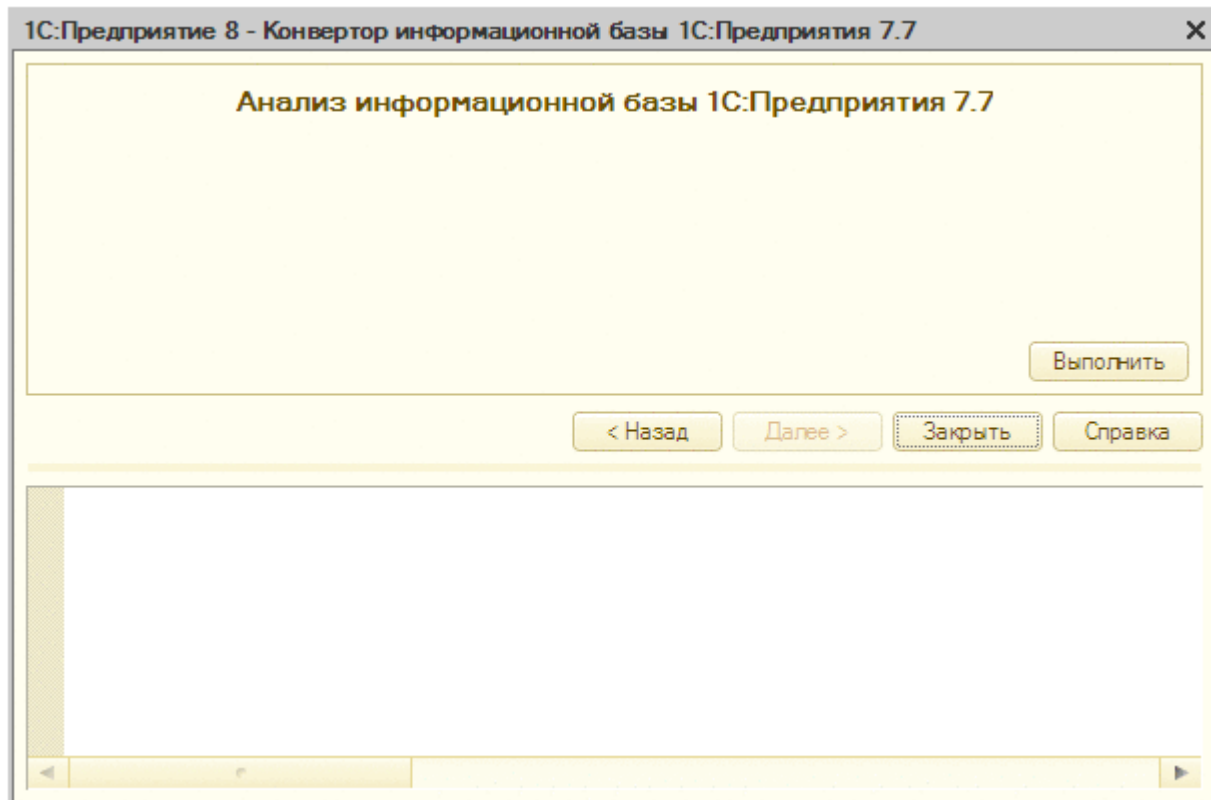


Рис. 575. Перед анализом информационной базы

Открывается исходная информационная база в пакетном режиме, и производится выгрузка данных. После выгрузки база закрывается.

Конвертер выводит информацию о периодических реквизитах объектов конфигурации и константах для

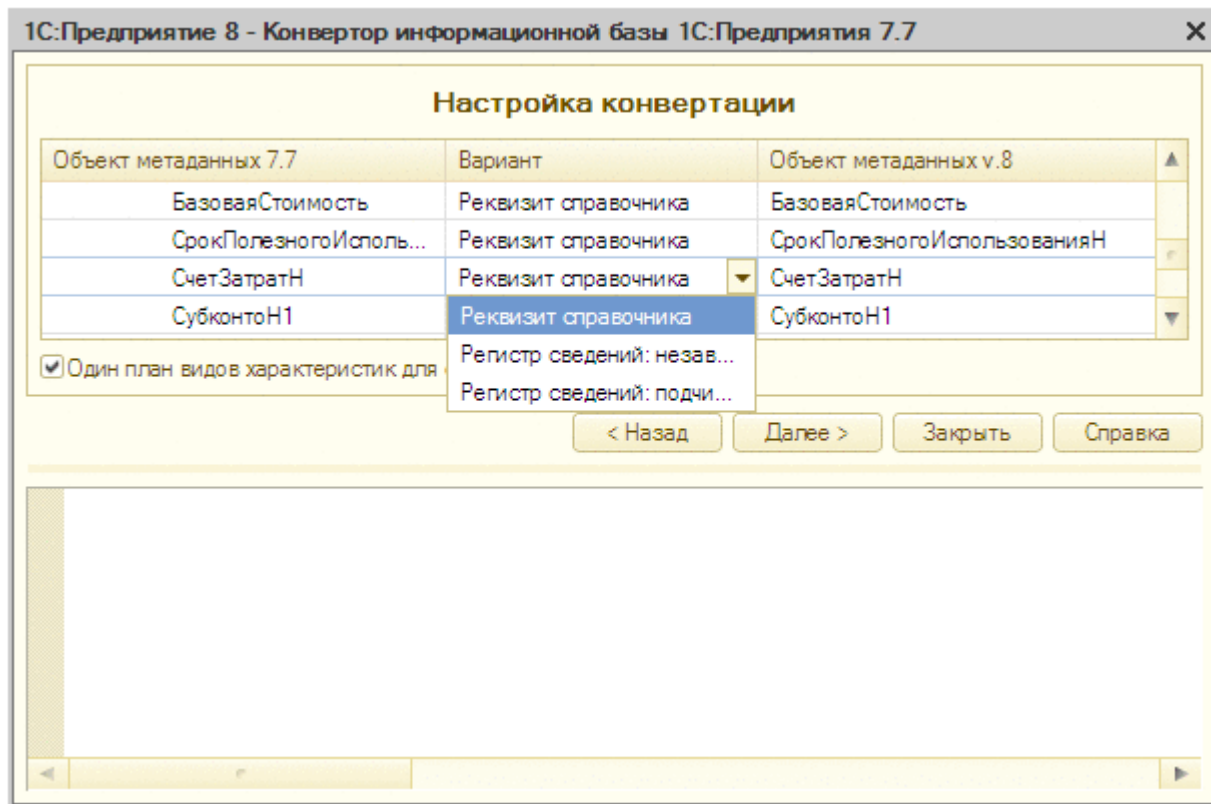


Рис. 576. Настройка конвертации

В колонке **Вариант** пользователь может выбрать, как нужно конвертировать данный реквизит (константу): в реквизит справочника (константу) или регистр сведений.

На следующем шаге для начала конвертации нужно нажать кнопку **Выполнить**.

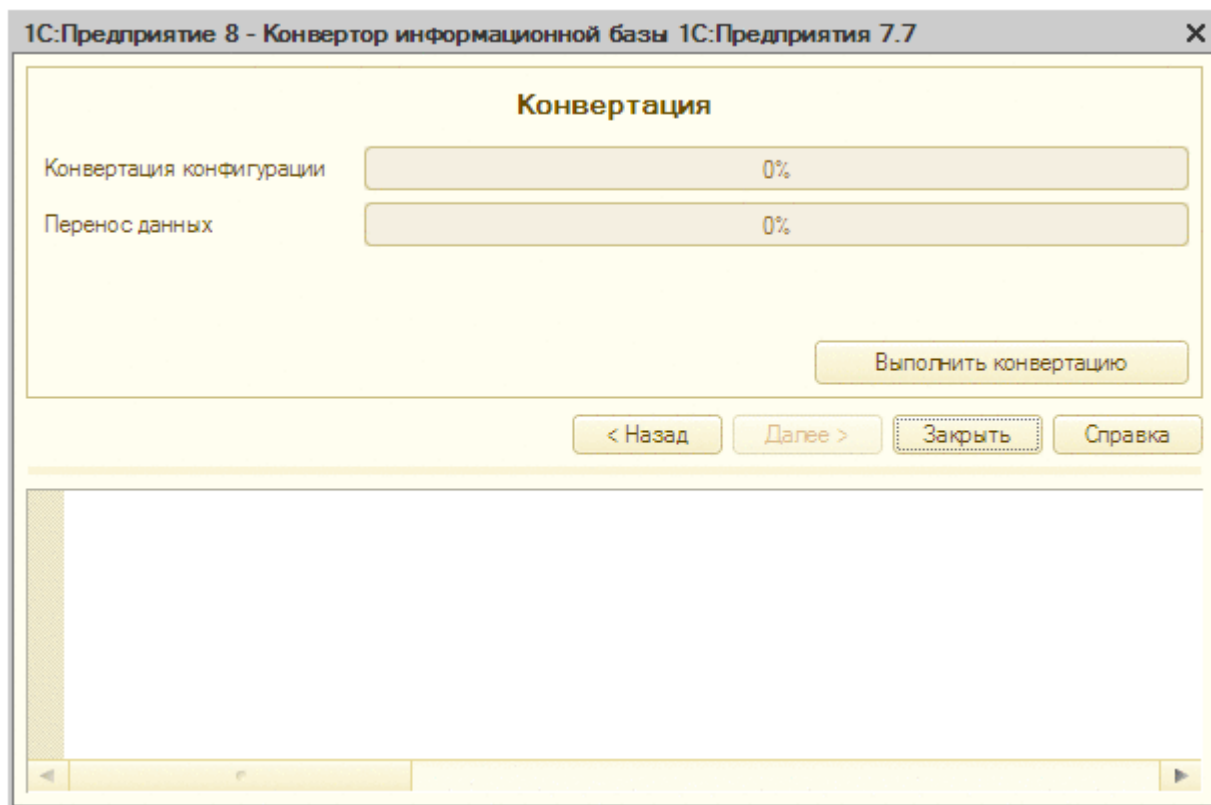


Рис. 577. Перед выполнением конвертации

Конвертер переносит структуру метаданных, формы, табличные документы, интерфейсы, наборы прав, картинки, а также данные базы данных и список пользователей.

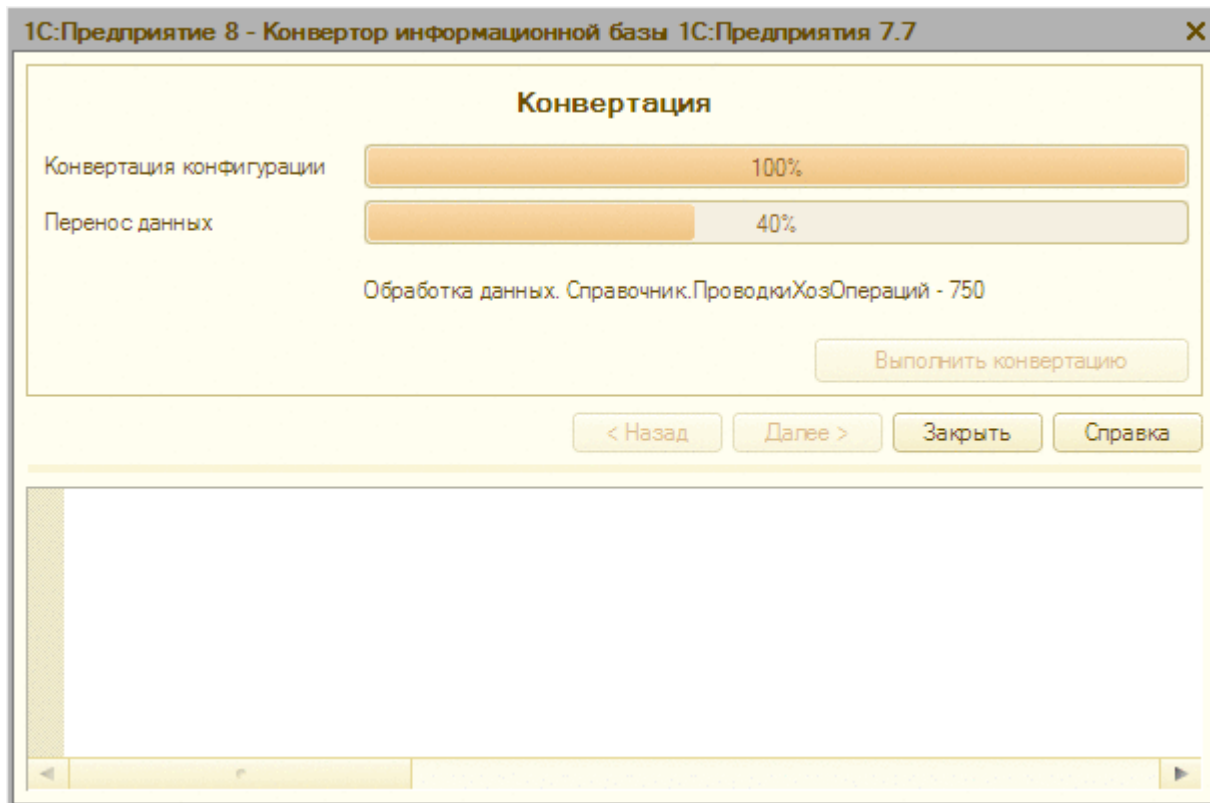


Рис. 578. Процесс конвертации

При создании документов в процессе конвертирования конфигурации принимаются во внимание признаки использования документа в видах учета. Если документ имеет признак использования в оперативном учете, при конвертировании документу добавляются в качестве движений все созданные при конвертировании регистры накопления; при использовании в бухгалтерском учете – все регистры бухгалтерии; при использовании в расчете – все регистры расчета.

Конвертация выполняется со следующими ограничениями:

- Переносятся объявления процедур, функций и переменных. Весь исполняемый код комментируется, за исключением программного кода, созданного самим конвертером.
- Графы журналов документов в процессе конвертации преобразуются в критерии отбора. Однако если, например, реквизит некоторого документа имеет тип *Справочники*, то, хотя он и попадет в состав критерия отбора, имеющего тип *Справочники.Сотрудники*, при следующем сохранении конфигурации будет выдано сообщение о неподходящем типе данного реквизита. Хотя формально тип *Справочники.Сотрудники* и включен в себя типом *Справочники*, критерий отбора не позволяет включать в себя реквизиты такого «составного» типа. Следует либо исправить тип реквизита, указав конкретные справочники, на объекты которых реквизит может хранить ссылки, либо исключить реквизит документа из состава критерия отбора.
- Пароли пользователей не переносятся.
- Конвертер не выполняет заполнение у конвертированных документов свойства *Движения*, определяющего состав регистров, в которых документ сохраняет движения. После выполнения конвертации состав этого свойства для всех документов следует задать вручную.

После завершения конвертации в окно выводится сообщение о завершении.

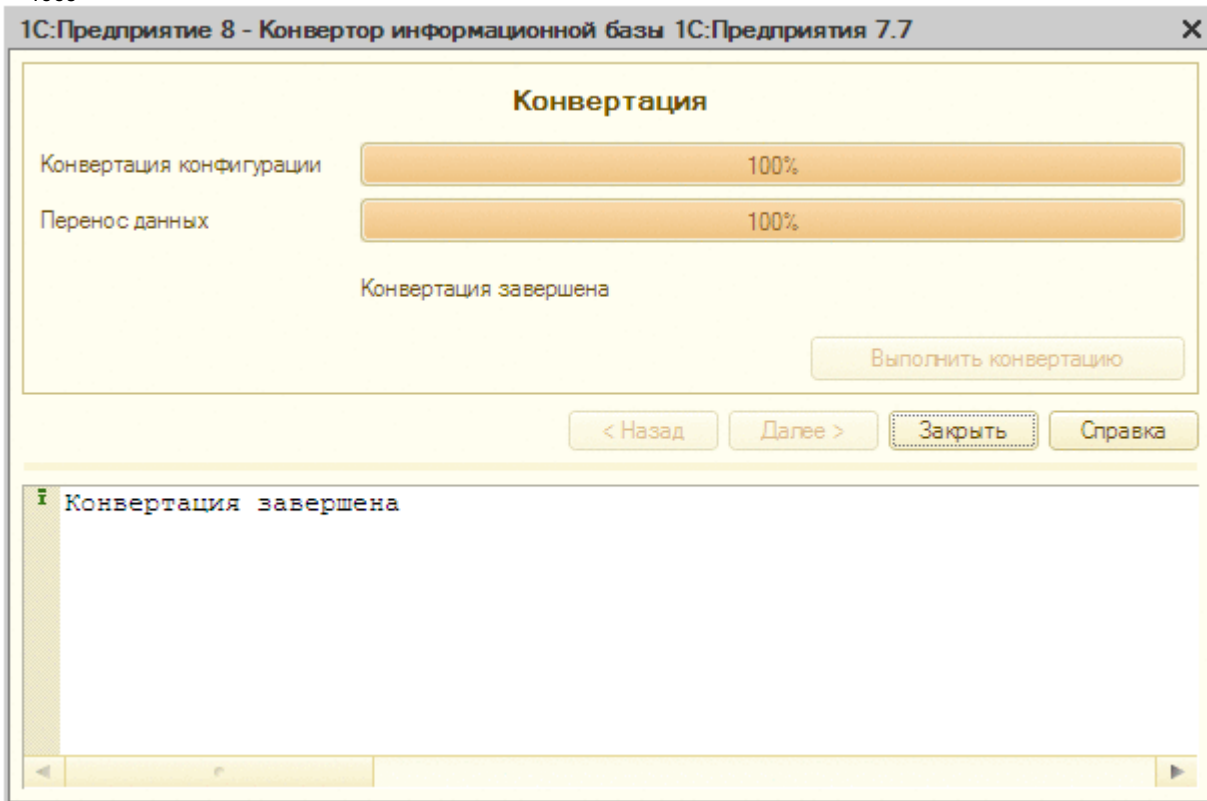


Рис. 579. Конвертация завершена

Приложение 2. Форматы навигационных ссылок

В данном приложении описаны форматы ссылок, которые используются в «1С:Предприятии». Ссылки бывают внешние и внутренние. Внешние ссылки предназначены для применения вне «1С:Предприятия», внутренние ссылки предназначены для внутреннего использования (в том числе для помещения в список избранного и истории работы).

2.1. Общее описание формата ссылки

Ссылка состоит из двух частей:

- Адрес хоста информационной базы – описывает расположение информационной базы относительно клиентского приложения. Для получения этой части ссылки предназначен метод `ПолучитьНавигационнуюСсылкуИнформационнойБазы()`.
- Внутренняя ссылка – описывает расположение запрашиваемой информации внутри информационной базы.

Внутренние ссылки бывают нескольких видов:

- объект информационной базы,
- реквизит объекта информационной базы,
- реквизит табличной части объекта информационной базы,
- реквизит записи регистра информационной базы,
- запись регистра информационной базы,
- отчет,
- обработка,
- раздел,
- команда раздела,
- временное хранилище.

Внешняя ссылка образуется добавлением к адресу хоста внутренней ссылки. Если по ссылке открывается форма или осуществляется навигация в главном окне, то внешняя ссылка образуется по следующему правилу:

```
<адрес хоста ИБ>#<внутренняя ссылка>
```

[Копировать в буфер обмена](#)

Таковыми ссылками являются ссылки на объект информационной базы, отчет, обработку, раздел, команду раздела.

Если ссылка обозначает просто ресурс, то внешняя ссылка формируется следующим образом:

```
<адрес хоста ИБ>/<внутренняя ссылка>
```

[Копировать в буфер обмена](#)

Таковыми ссылками являются ссылки на реквизит объекта информационной базы или записи регистра сведений или временное хранилище.

СОВЕТ. Если с информационной базой работают из разных клиентов, не рекомендуется запоминать внешние ссылки на объекты информационной базы, так как это может привести к неработоспособности ссылки, созданной на одном клиенте, в другом клиенте. Например, внешняя ссылка, созданная в веб-клиенте, может оказаться неработоспособной в тонком клиенте, подключенном «напрямую» к серверу «1С:Предприятия». Рекомендуется запоминать внутренние ссылки.

Если в метод `ПерейтиПоНавигационнойСсылке()` или в диалог перехода по навигационной ссылке передается внешняя ссылка, у которой есть в пути сигнатура `e1cib`, а начало ссылки не совпадает с навигационной ссылкой информационной базы, то вначале осуществляется выделение из переданной ссылки внутренней и выполняется попытка перехода по выделенной части. Если попытка не удастся, то осуществляется переход по полной ссылке.

2.2. Формат адреса хоста

В зависимости от вида информационной базы строка, описывающая адрес хоста, может иметь разный вид.

Файловый вариант информационной базы

Если путь к файлу базы данных задан в формате UNC, то адрес хоста выглядит следующим образом:

```
e1c:/file/<UNC-путь>
```

[Копировать в буфер обмена](#)

Пример:

```
// Путь к файловой информационной базе
\\dbsrvr\bases\mybase
// Адрес хоста
e1c:/file/dbsrvr/bases/mybase
```

[Копировать в буфер обмена](#)

Если путь к файлу указан с указанием буквы диска, то адрес хоста выглядит следующим образом:

```
e1c:/filev/<Буква диска>/<Путь к базе>
```

[Копировать в буфер обмена](#)

Пример:

```
// Путь к файловой информационной базе
s:\bases\mybase
```

[Копировать в буфер обмена](#)

Клиент-серверный вариант информационной базы

Для клиент-серверного варианта информационной базы ссылка выглядит следующим образом:

```
elc://server/<имя сервера>/<имя информационной базы>
```

[Копировать в буфер обмена](#)

Пример:

```
// Строка соединения с информационной базой
srvsr="srv1C";ref="mybase"
// Адрес хоста
elc://server/srv1C/mybase
```

[Копировать в буфер обмена](#)**При работе через веб-сервер (тонкий клиент или веб-клиент)**

Для варианта информационной базы, работа с которой идет через веб-сервер, ссылка выглядит следующим образом:

```
http://<адрес хоста>/<путь к информационной базе>
https://<адрес хоста>/<путь к информационной базе>
```

[Копировать в буфер обмена](#)

Пример:

```
// Строка соединения с информационной базой
http://localhost/mybase
// Адрес хоста
http://localhost/mybase
```

[Копировать в буфер обмена](#)**2.3. Внутренние ссылки**

Общий формат внутренней ссылки выглядит следующим образом:

```
elcib/<тип>/<путь>[?<параметр>=<значение параметра>[&<параметр>=<значение параметра>]]
```

[Копировать в буфер обмена](#)

Более подробное описание форматов ссылок, в зависимости от типа, приведено далее.

2.3.1. Ссылка на объект информационной базы

Формат:

```
elcib/data/<путь к метаданным>?ref=<идентификатор ссылки>
```

[Копировать в буфер обмена](#)

Где:

- **путь к метаданным** – тип объекта, на который указывает ссылка, например `Документ.РасходТовара`;
- **идентификатор ссылки** – уникальный идентификатор объекта в информационной базе.

2.3.2. Ссылка на реквизит объекта информационной базы

Формат:

```
elcib/data/<путь к метаданным>.<имя реквизита>?ref=<идентификатор ссылки>
```

[Копировать в буфер обмена](#)

Где:

- **путь к метаданным** – тип объекта, на который указывает ссылка, например `Документ.РасходТовара`;
- **имя реквизита** – имя реквизита объекта информационной базы;
- **идентификатор ссылки** – уникальный идентификатор объекта в информационной базе.

2.3.3. Ссылка на реквизит табличной части объекта информационной базы

Формат:

```
elcib/data/<путь к метаданным>.<имя табличной части>.<имя реквизита>?ref=<идентификатор ссылки>&index=<индекс строки табличной части>
```

[Копировать в буфер обмена](#)

Где:

- **путь к метаданным** – тип объекта, на который указывает ссылка, например `Документ.РасходТовара`;
- **имя табличной части** – имя табличной части объекта метаданных, например `Товары`;
- **имя реквизита** – имя реквизита табличной части объекта информационной базы;
- **идентификатор ссылки** – уникальный идентификатор объекта в информационной базе;
- **индекс строки табличной части** – индекс строки табличной части.

2.3.4. Ссылка на запись регистра информационной базы

Формат:

`elcib/data/<путь к метаданным>?<имя ключевого поля>=<значение>[&<имя ключевого поля>=<значение>]`

[Копировать в буфер обмена](#)

Где:

- **путь к метаданным** – тип объекта, на который указывает ссылка, например `РегистрСведений.ОстаткиТоваров`;
- **имя ключевого поля** – имя измерения регистра (для периодических регистров добавляется измерение `Период`);
- **значение** – внутреннее представление значения отбора. Значение отбора кодируется в соответствии с разделом [2.2. URL Character Encoding Issues](#) стандарта [RFC 1738](http://tools.ietf.org/html/rfc1738) (<http://tools.ietf.org/html/rfc1738>) – с помощью символа «%» и двух шестнадцатеричных символов.

2.3.5. Ссылка на реквизит записи регистра информационной базы

Формат:

`elcib/data/<путь к метаданным>.<имя реквизита>?<имя ключевого поля>=<значение>[&<имя ключевого поля>=<значение>]`

[Копировать в буфер обмена](#)

Где:

- **путь к метаданным** – тип объекта, на который указывает ссылка, например `РегистрСведений.ОстаткиТоваров`;
- **имя реквизита** – имя реквизита записи;
- **имя ключевого поля** – имя измерения регистра (для периодических регистров добавляется измерение `Период`);
- **значение** – внутреннее представление значения отбора. Значение отбора кодируется в соответствии с разделом [2.2. URL Character Encoding Issues](#) стандарта [RFC 1738](http://tools.ietf.org/html/rfc1738) (<http://tools.ietf.org/html/rfc1738>) – с помощью символа «%» и двух шестнадцатеричных символов.

2.3.6. Ссылка на отчет

Формат:

`elcib/app/<путь к метаданным>`

[Копировать в буфер обмена](#)

Где:

- **путь к метаданным** – тип объекта, на который указывает ссылка, например `Отчет.ОстаткиТоваров`.

2.3.7. Ссылка на обработку

Формат:

`elcib/app/<путь к метаданным>`

[Копировать в буфер обмена](#)

Где:

- **путь к метаданным** – тип объекта, на который указывает ссылка, например `Обработка.ФормированиеЗаказа`.

2.3.8. Ссылка на раздел

Формат:

`elcib/navigationpoint/<имя раздела>`

[Копировать в буфер обмена](#)

Где:

- **имя раздела** – имя подсистемы первого уровня, на который указывает ссылка, например `ПродажиТоваров`.

2.3.9. Ссылка на точку навигации, образованную стандартной командой

Формат:

`elcib/navigationpoint/<имя раздела>/<имя команды>`

[Копировать в буфер обмена](#)

Где:

- **имя раздела** – имя подсистемы первого уровня, на который указывает ссылка, например `ПродажиТоваров`;
- **имя команды** – имя команды, например `Справочник.Товары.ОткрытьСписок`.

2.3.10. Ссылка на временное хранилище

Формат:

`elcib/tempstorage/<идентификатор временного значения>`

[Копировать в буфер обмена](#)

Где:

- **идентификатор временного значения** – идентификатор временного значения.

2.3.11. Ссылка на запись таблицы внешнего источника данных

Формат:

```
elcib/data/<путь к метаданным>?<имя ключевого поля>=<значение>[&<имя ключевого поля>=<значение>]
```

[Копировать в буфер обмена](#)

Где:

- **путь к метаданным** – описывает таблицу внешнего источника данных, например `ВнешнийИсточникДанных.Чеки.Таблица.CheckHeaders`.
- **имя ключевого поля** – имя поля ключа таблицы внешнего источника данных.
- **значение** – внутреннее представление значения отбора. Значение отбора кодируется в соответствии с разделом [2.2. URL Character Encoding Issues](#) стандарта [RFC 1738](http://tools.ietf.org/html/rfc1738) (<http://tools.ietf.org/html/rfc1738>) – с помощью символа «%» и двух шестнадцатеричных символов.

2.3.12. Ссылки на команды глобального командного интерфейса

Формат:

```
elcib/command/<имя команды>
```

[Копировать в буфер обмена](#)

Где:

- **имя команды** – имя команды:
 - стандартная команда – `Справочник.Товары.Создать`;
 - общая команда – `ОбщаяКоманда.ПоискОбъектов`;
 - команда – `Справочник.Товары.Команда.ПечатьШтрихКода`.

2.3.13. Ссылки на списки

Формат:

```
elcib/list/<имя списка>
```

[Копировать в буфер обмена](#)

Где:

- **имя списка** – имя открываемого списка, например `Справочник.Товар.Документы.РасходТоваров`.

2.3.14. Ссылка на начальную страницу

Формат:

```
elcib/navigationpoint/desktop
```

[Копировать в буфер обмена](#)

В режиме совместимости с версией 8.3.2 и младше.

Формат:

```
elcib/navigationpoint/startpage
```

[Копировать в буфер обмена](#)

В режиме совместимости старше, чем версия 8.3.2.

Приложение 3. Правила формирования текстов стандартных команд и автоматических заголовков форм

Для того чтобы иметь возможность переопределять (или уточнять) представление объектов в интерфейсе, в свойствах метаданных реализован специальный набор свойств.

Объект	Расширенное представление	Представление объекта (для регистра – записи)	Расширенное представление объекта (для регистра – записи)	Представление списка	Расширенное представление списка	Пояснение	Картинка
Подсистемы						+	+
Константы	+					+	
Планы обмена		+	+	+	+	+	
Критерии отбора				+	+	+	
Общие формы	+					+	
Константы	+					+	
Справочники		+	+	+	+	+	
Документы		+	+	+	+	+	
Журналы документов				+	+	+	
Перечисления				+	+	+	
Отчеты	+					+	
Обработки	+					+	
Планы видов характеристик		+	+	+	+	+	
Планы счетов		+	+	+	+	+	
Планы видов расчета		+	+	+	+	+	
Регистры сведений (подчиненные)				+	+	+	
Регистры сведений (независимые)		+	+	+	+	+	
Регистры накоплений				+	+	+	
Регистры бухгалтерии				+	+	+	
Регистры расчета				+	+	+	
Бизнес-процессы		+	+	+	+	+	
Задачи		+	+	+	+	+	

При формировании представления стандартных команд (и подсказок к ним) система руководствуется следующими правилами (**Свойство1** " **Свойство2** означает, что если **Свойство1** не задано, то используется **Свойство2**):

- Открытие списка:
 - представление команды: [Представление списка](#);
 - подсказка команды: [Пояснение](#) " [Расширенное представление списка](#) " [Представление списка](#).
- Создание нового объекта (записи):
 - представление команды: [Представление объекта](#): создать или [Представление записи](#): создать;
 - подсказка команды: [Расширенное представление объекта \(Расширенное представление записи\)](#) " [Представление объекта \(Представление записи\)](#).
- Создание новой группы:

- представление команды: [Представление списка](#): создать группу;
- подсказка команды: [Расширенное представление списка](#) " [Представление списка](#).
- Переход к любому подчиненному списку (команда группы [Перейти](#) панели навигации формы):
 - представление команды: [Представление списка](#);
 - подсказка команды: [Пояснение](#) " [Расширенное представление списка](#) " [Представление списка](#).
- Ввод на основании:
 - представление команды: [Представление объекта](#): создать на основании;
 - подсказка команды: [Расширенное представление объекта](#) " [Представление объекта](#).
- Открытие отчета или обработки:
 - представление команды: представление отчета или обработки;
 - подсказка команды: [Пояснение](#) " [Расширенное представление](#).
- Открытие общей формы:
 - представление команды: представление общей формы;
 - подсказка команды: [Пояснение](#) " [Расширенное представление](#).
- Открытие формы редактирования констант:
 - представление команды: представление константы;
 - подсказка команды: [Пояснение](#) " [Расширенное представление](#).
- Переход к подсистеме (в панели разделов):
 - представление команды: представление подсистемы;
 - графическое представление: [Картинка](#) " стандартная картинка раздела панели разделов;
 - подсказка команды: [Пояснение](#).

ПРИМЕЧАНИЕ. Если все используемые (в формировании представления и/или подсказки) свойства не заданы, то используется текстовое представление объекта метаданных, как его возвращает метод [Представление\(\)](#).

При размещении команды в панели действий или командной панели формы действует следующее правило: из текста представления команды удаляется название группы команд (предваряемое символом «:»), если это сочетание является завершением представления команды. Например, стандартная команда создания элемента справочника [Товары](#), выглядящая как [Товары: создать](#), в панели действий будет выглядеть как [Товары](#). Если мы создадим команду, которая будет иметь представление [Наша команда: создать на основании](#), и поместим ее в стандартную группу командной панели формы [Создать на основании](#), то представление команды станет [Наша команда](#). В то же время если поместить эту же команду в группу [Печать](#) (которая создана разработчиком) командной панели формы, то представление нашей команды останется [Наша команда: создать на основании](#).

Свойства, связанные с представлением объектов (см. таблицу выше), также используются при формировании заголовков форм в том случае, если свойство [Автоматический заголовок](#) формы установлено в значение [Истина](#). В этом случае заголовок формы является объединением свойства [Заголовок](#) формы и автоматически формируемого заголовка. Если это свойство имеет значение [Ложь](#), то в качестве заголовка используется свойство формы [Заголовок](#) без каких-либо преобразований.

При автоматическом формировании заголовка формы система руководствуется следующими правилами:

- заголовок формы списка: [Расширенное представление списка](#) " [Представление списка](#);
- заголовок формы справочника, плана счетов, плана видов расчета, плана обмена, плана видов характеристик, задачи:
 - существующий элемент: текстовое представление объекта ([Расширенное представление объекта](#) " [Представление объекта](#)), например:
 - Зиновьев Антон (Сотрудник);
 - 70.1 (Счет бухгалтерского учета);
 - новый элемент: [Расширенное представление объекта](#) (Создание) " [Представление объекта](#) (Создание);
- заголовок формы документа, бизнес-процесса:
 - существующий элемент: текстовое представление объекта, например:
 - приходная накладная № 12001 от 12.05.2008 16:15:32;

- новый элемент: [Расширенное представление объекта](#) (Создание) " [Представление объекта](#) (Создание);
- заголовок формы записи регистра сведений:
 - существующая запись: [Расширенное представление записи](#) " [Представление записи](#);
 - новая запись: [Расширенное представление записи](#) (Создание) " [Представление записи](#) (Создание);
- заголовок формы группы справочника или плана видов характеристик:
 - существующая группа: текстовое представление объекта ([Расширенное представление списка](#) " [Представление списка](#));
 - новая группа: [Расширенное представление списка](#) (Создание группы) " [Представление списка](#) (Создание группы);
- заголовок формы отчета, обработки: [Расширенное представление](#).

ПРИМЕЧАНИЕ. Если все используемые в формировании наименования свойства не заданы, то используется текстовое представление объекта метаданных, как его возвращает метод [Представление \(\)](#).

Приложение 4. Перечень автоматически сохраняемых настроек

Приложение описывает настройки, которые автоматически сохраняются платформой в системном хранилище.

4.1. Хранилище: вариантов отчета

Настройка: варианты отчета

- Ключ объекта – полное имя отчета.

Пример ключа объекта:

Отчет.Продажи

[Копировать в буфер обмена](#)

- Ключ настройки – текстовый идентификатор варианта.

Пример ключа настройки:

ПродажиПоРегионам

[Копировать в буфер обмена](#)

- Тип сохраняемого значения – **НастройкиКомпоновкиДанных**.

4.2. Хранилище: настроек отчета

Настройка: настройки отчета

- Ключ объекта – текстовая строка, составленная из полного имени отчета, символа «/» и ключа варианта.

Пример ключа объекта:

Отчет.Продажи/ПродажиПоРегионам

[Копировать в буфер обмена](#)

- Ключ настройки – текстовый идентификатор настройки.

Пример ключа настройки:

МоиКлиенты

[Копировать в буфер обмена](#)

- Тип сохраняемого значения – **ПользовательскиеНастройкиКомпоновкиДанных**.

4.3. Хранилище: данные форм

Настройка: значения полей формы

- Ключ объекта – полное имя формы.

Пример ключа объекта:

Обработка.ВыгрузкаДанных.Форма.ОсновнаяФорма

[Копировать в буфер обмена](#)

- Ключ настройки – текстовый идентификатор сохраняемых значений.

Пример ключа настройки:

Выгрузка в Сбербанк

- Тип сохраняемого значения – **Соответствие**. Ключом соответствия выступает текстовый путь к сохраняемому реквизиту данных, а значением является значение реквизита.

4.4. Системное хранилище

Настройка: ключ текущего варианта отчета

- Ключ объекта – текстовая строка, составленная из полного имени отчета и строки **/КлючТекущегоВарианта (/CurrentVariantKey)**.

Пример ключа объекта:

Копировать в буфер обмена

Отчет. Продажи/КлючТекущегоВарианта

Если у формы установлено свойство **КлючНазначенияИспользования**, то ключ объекта формируется следующим образом: полное имя отчета, символ "/", значение свойства **КлючНазначенияИспользования** и строка **/КлючТекущегоВарианта**.

Пример ключа объекта:

Копировать в буфер обмена

Отчет. Продажи/ДляФормыНачисленияСкидок/КлючТекущегоВарианта

В данном примере свойство **КлючНазначенияИспользования** имеет значение **ДляФормыНачисленияСкидок**.

- Ключ настройки – пустая строка.
- Тип сохраняемого значения – произвольное значение, содержащее ключ текущего варианта.

Настройка: ключ текущей настройки отчета

- Ключ объекта – является строковым значением, состоящим из полного имени отчета, ключа варианта отчета (в виде строки) и текста **/КлючТекущихПользовательскихНастроек (/CurrentUserSettingsKey)**.

Пример ключа объекта:

Копировать в буфер обмена

Отчет. Продажи/ПродажиПоРегионам/КлючТекущихПользовательскихНастроек

Если у формы установлено свойство **КлючНазначенияИспользования**, то ключ объекта формируется следующим образом: полное имя отчета, символ "/", ключ варианта отчета в виде строки, символ "/", значение свойства **КлючНазначенияИспользования** и строка **/КлючТекущихПользовательскихНастроек**.

Пример ключа объекта:

Копировать в буфер обмена

Отчет. Продажи/ПродажиПоРегионам/ДляФормыНачисленияСкидок/КлючТекущихПользовательскихНастроек

В данном примере ключ варианта отчета имеет значения **ПродажиПоРегионам**, а свойство **КлючНазначенияИспользования** имеет значение **ДляФормыНачисленияСкидок**.

- Ключ настройки – пустая строка.
- Тип сохраняемого значения – произвольное значение, содержащее ключ текущего варианта.

Настройка: настройки варианта отчета при закрытии отчета или смене варианта

- Ключ объекта – является строковым значением, состоящим из полного имени отчета, ключа варианта отчета (в виде строки) и текста **/ТекущиеПользовательскиеНастройки (/CurrentUserSettings)**.

Отчет . Продажи / ПродажиПоРегионам / ТекущиеПользовательскиеНастройки

Если у формы установлено свойство **КлючНазначенияИспользования**, то ключ объекта формируется следующим образом: полное имя отчета, символ "/", ключ варианта отчета в виде строки, символ "/", значение свойства **КлючНазначенияИспользования** и строка **/ТекущиеПользовательскиеНастройки**.

Пример ключа объекта:

Отчет . Продажи / ПродажиПоРегионам / ДляФормыНачисленияСкидок / ТекущиеПользовательскиеНастройки

В данном примере ключ варианта отчета имеет значения **ПродажиПоРегионам**, а свойство **КлючНазначенияИспользования** имеет значение **ДляФормыНачисленияСкидок**.

- Ключ настройки – пустая строка.
- Тип сохраняемого значения – **ПользовательскиеНастройкиКомпоновкиДанных**.

Настройка: ключ текущей настройки полей формы

- Ключ объекта – является строковым значением, состоящим из полного имени формы и текста **/КлючТекущихНастроекДанных (/CurrentDataSettingsKey)**.

Пример ключа объекта:

Обработка . ВыгрузкаДанных . Форма . ОсновнаяФорма / КлючТекущихНастроекДанных

Если у формы установлено свойство **КлючНазначенияИспользования**, то ключ объекта формируется следующим образом: полное имя формы, символ "/", значение свойства **КлючНазначенияИспользования** и строка **/КлючТекущихНастроекДанных**.

Пример ключа объекта:

Обработка . ВыгрузкаДанных . Форма . ОсновнаяФорма / ДляФормыНачисленияСкидок / КлючТекущихНастроекДанных

В данном примере свойство **КлючНазначенияИспользования** имеет значение **ДляФормыНачисленияСкидок**.

- Ключ настройки – пустая строка.
- Тип сохраняемого значения – произвольное значение, содержащее ключ текущей настройки.

Настройка: значения полей формы при закрытии

- Ключ объекта – является строковым значением, состоящим из полного имени формы и текста **/ТекущиеДанные (/CurrentData)**.

Пример ключа объекта:

Обработка . ВыгрузкаДанных . Форма . ОсновнаяФорма / ТекущиеДанные

Если у формы установлено свойство **КлючНазначенияИспользования**, то ключ объекта формируется следующим образом: полное имя формы, символ "/", значение свойства **КлючНазначенияИспользования** и строка **/ТекущиеДанные**.

Пример ключа объекта:

Обработка . ВыгрузкаДанных . Форма . ОсновнаяФорма / ДляФормыНачисленияСкидок / ТекущиеДанные

В данном примере свойство **КлючНазначенияИспользования** имеет значение **ДляФормыНачисленияСкидок**.

- Ключ настройки – пустая строка.
- Тип сохраняемого значения – **Соответствие**. Ключом элемента соответствия выступает текстовый путь к сохраняемому реквизиту данных, а значением является значение реквизита.

Настройка: настройка отображения формы

- Ключ объекта – является строковым значением, состоящим из полного имени формы и текста **/НастройкиФормы (/FormSettings)**.

Пример ключа объекта:

[Копировать в буфер обмена](#)

Обработка.ВыгрузкаДанных.Форма.ОсновнаяФорма/НастройкиФормы

Если у формы установлено свойство **КлючНазначенияИспользования** (значение свойства не пустое), то ключ объекта формируется следующим образом: полное имя формы, символ "/", значение свойства **КлючНазначенияИспользования** и строка **/НастройкиФормы**.

Пример ключа объекта:

[Копировать в буфер обмена](#)

Обработка.ВыгрузкаДанных.Форма.ОсновнаяФорма/ОсновнойВариант/НастройкиФормы

В данном примере свойство **КлючНазначенияИспользования** имеет значение **ОсновнойВариант**.

- Ключ настройки – пустая строка.
- Тип сохраняемого значения – **НастройкиФормы**. Объект без свойств и методов.

Настройка: размеры формы и элементов управления, положение разделителей и ширины колонок таблиц

- Ключ объекта – является строковым значением, состоящим из полного имени формы и текста **/НастройкиОкна (/WindowSettings)**.

Пример ключа объекта:

[Копировать в буфер обмена](#)

Обработка.ВыгрузкаДанных.Форма.ОсновнаяФорма/НастройкиОкна

Если при работе с формой задано свойство формы **КлючСохраненияПоложенияОкна**, то ключ объекта будет включать значение этого ключа **<ЗначениеКлючаСохраненияПоложенияОкна>/НастройкиОкна**.

Например, если значение ключа сохранения окна равно **ОсновнойРежим**, то вышеуказанный пример будет выглядеть следующим образом:

[Копировать в буфер обмена](#)

Обработка.ВыгрузкаДанных.Форма.ОсновнаяФорма/ОсновнойРежим/НастройкиОкна

- Ключ настройки – пустая строка.
- Тип сохраняемого значения – **НастройкиОкна**. Объект без свойств и методов.

Настройка: настройки для главного окна приложения

- Ключ объекта – строковое значение **ОсновноеОкно/НастройкиОкна (MainWindow/WindowSettings)**.
- Ключ настройки – пустая строка.
- Тип сохраняемого значения – **НастройкиОкна**. Объект без свойств и методов.

Настройка: положение разделителей рабочего стола

- Ключ объекта – строковое значение:

- [РабочийСтол/НастройкиОкна \(Desktop/WindowSettings\)](#) в режиме совместимости с версией 8.3.2 и младше.
- [НачальнаяСтраница/НастройкиОкна \(StartPage/WindowSettings\)](#) в режиме совместимости старше, чем версия 8.3.2.
- [НачальнаяСтраница/Такси/НастройкиОкна \(StartPage/Taxi/WindowSettings\)](#) при использовании интерфейса [Такси](#).

- Ключ настройки – пустая строка.

- Тип сохраняемого значения – [НастройкиОкна](#). Объект без свойств и методов.

Настройка: размеры формы и элементов управления, положение разделителей и ширины колонок таблиц в веб-клиенте

- Ключ объекта – является строковым значением, состоящим из полного имени формы и текста [/НастройкиОкнаВебКлиента \(/WebClientWindowSettings\)](#).

Пример ключа объекта:

[Копировать в буфер обмена](#)

Обработка.ВыгрузкаДанных.Форма.ОсновнаяФорма/НастройкиОкнаВебКлиента

- Ключ настройки – пустая строка.
- Тип сохраняемого значения – [НастройкиОкна](#). Объект без свойств и методов.

Настройка: настройки начальной страницы (рабочего стола)

- Ключ объекта – строковое значение:
 - [Общее/НастройкиРабочегоСтола \(Common/DesktopSettings\)](#) в режиме совместимости с версией 8.3.2 и младше.
 - [Общее/НастройкиНачальнойСтраницы \(Common/StartPageSettings\)](#) в режиме совместимости старше, чем версия 8.3.2.
- Ключ настройки – пустая строка.
- Тип сохраняемого значения – [НастройкиНачальнойСтраницы](#). Объект без свойств и методов.

Настройка: избранное

- Ключ объекта – строковое значение [Общее/ИзбранноеРаботыПользователя \(Common/UserWorkFavorites\)](#).
- Ключ настройки – пустая строка.
- Тип сохраняемого значения – [ИзбранноеРаботыПользователя](#).

Настройка: история поиска в таблице, связанной с динамическим списком

- Ключ объекта – [Общее/ИсторияПоискаТаблицы/<Имя формы>.<Имя таблицы> \(Common/TableSearchHistory/<Имя формы>.<Имя таблицы>\)](#).

Пример ключа объекта:

[Копировать в буфер обмена](#)

Общее/ИсторияПоискаТаблицы/Справочник.Товары.Форма.ФормаСписка.Список

- Ключ настройки – пустая строка.

- Тип сохраняемого значения – *ИсторияПоискаТаблицы*. Объект без свойств и методов.

Настройка: настройки глобального командного интерфейса

- Ключ объекта – строковое значение, один из вариантов:
 - [Общее/ПанельРазделов/НастройкиКомандногоИнтерфейса \(Common/SectionsPanel/CommandInterfaceSettings\)](#) – настройки панели разделов;
 - полное имя фрагмента командного интерфейса и [/ПанельДействий/НастройкиКомандногоИнтерфейса \(/ActionsPanel/CommandInterfaceSettings\)](#) – настройки панели действий подсистемы;
 - полное имя фрагмента командного интерфейса и [/ПанельНавигации/НастройкиКомандногоИнтерфейса \(/NavigationPanel/CommandInterfaceSettings\)](#) – настройки панели навигации подсистемы;
 - [РабочийСтол/ПанельДействий/НастройкиКомандногоИнтерфейса \(Desktop/ActionsPanel/CommandInterfaceSettings\)](#) – настройки панели действий рабочего стола (в режиме совместимости с версией 8.3.2 и младше);
 - [ОсновнойРаздел/ПанельДействий/НастройкиКомандногоИнтерфейса \(MainSection/ActionsPanel/CommandInterfaceSettings\)](#) – настройки панели действий рабочего стола (в режиме совместимости старше, чем версия 8.3.2);
 - [РабочийСтол/ПанельНавигации/НастройкиКомандногоИнтерфейса \(Desktop/NavigationPanel/CommandInterfaceSettings\)](#) – настройки панели навигации рабочего стола (в режиме совместимости с версией 8.3.2 и младше);
 - [ОсновнойРаздел/ПанельНавигации/НастройкиКомандногоИнтерфейса \(MainSection/NavigationPanel/CommandInterfaceSettings\)](#) – настройки панели навигации рабочего стола (в режиме совместимости старше, чем версия 8.3.2).

Пример ключа объекта:

[Копировать в буфер обмена](#)

Подсистема .Продажи/ПанельНавигации/НастройкиКомандногоИнтерфейса
РабочийСтол/ПанельДействий/НастройкиКомандногоИнтерфейса

- Ключ настройки – пустая строка.
- Тип сохраняемого значения – *НастройкиКомандногоИнтерфейса*. Объект без свойств и методов.

Настройка: настройки печати табличного документа

- Ключ объекта – строковое значение, состоящее из [Общее/НастройкиПечатиТабличногоДокумента/ \(Common/SpreadsheetDocumentPrintSettings/\)](#) и имени параметров печати.

Пример ключа объекта:

[Копировать в буфер обмена](#)

Общее/НастройкиПечатиТабличногоДокумента/РасходнаяНакладная

- Ключ настройки – пустая строка.
- Тип сохраняемого значения – *НастройкиПечатиТабличногоДокумента*. Объект без свойств и методов.

Настройка: настройки клиентского приложения

- Ключ объекта – строковое значение [Общее/НастройкиКлиентскогоПриложения \(Common/ClientSettings\)](#).

- Ключ настройки – пустая строка.
- Тип сохраняемого значения – *НастройкиКлиентскогоПриложения*. Объект без свойств и методов.

Содержит признак отображения команды [Все функции](#), установки режима отладки при запуске, установки режима низкой скорости соединения (для веб-клиента), признак [Внешняя](#) диалога получения ссылки.

Для интерфейса [Такси](#) в данных настройках дополнительно сохраняются изменения панелей относительно состава и расположения, заданного в конфигураторе.

Настройка: настройки интерфейса клиентского приложения

- Ключ объекта – строковое значение [Общее/НастройкиИнтерфейсаКлиентскогоПриложения \(Common/ClientApplicationInterfaceSettings\)](#).
- Ключ настройки – пустая строка.
- Сохраняемое значение – объект [НастройкиИнтерфейсаКлиентскогоПриложения](#). Содержит настройки расположения панелей приложения.

Настройка: настройки внешних компонент

- Ключ объекта – строковое значение [НастройкиВнешнейКомпоненты/<ИмяНастроекВнешнейКомпоненты> \(AddInSettings/<ИмяНастроекВнешнейКомпоненты>\)](#). Строковое значение [<ИмяНастроекВнешнейКомпоненты>](#) определяется внешней компонентой с помощью метода [RegisterProfileAs\(\)](#).

Пример ключа объекта:

[Копировать в буфер обмена](#)

[НастройкиВнешнейКомпоненты/СканерШтрихКодов](#)

- Ключ настройки – пустая строка.
- Тип сохраняемого значения – [НастройкиВнешнейКомпоненты](#). Объект без свойств и методов.

Настройка: настройки справочной системы

- Ключ объекта – строковое значение [Общее/НастройкиСправки \(Common/HelpSettings\)](#).
- Ключ настройки – пустая строка.
- Тип сохраняемого значения – [НастройкиСправки](#). Объект без свойств и методов.

Настройка: цветовые настройки параметров сравнения файлов

- Ключ объекта – строковое значение [Общее/НастройкиСравнения \(Common/ComparisonSettings\)](#).
- Ключ настройки – пустая строка.
- Тип сохраняемого значения – [НастройкиСравнения](#). Объект без свойств и методов.

Настройка: история поиска

- Ключ объекта – строковое значение [Общее/ПараметрыПоискаТаблиц \(Common/TableSearchParameters\)](#).
- Ключ настройки – пустая строка.
- Тип сохраняемого значения – список значений, содержащий список строк истории. Наиболее поздние элементы находятся в начале списка.

Приложение 5. Поисковые выражения полнотекстового поиска

Поиск может осуществляться по нескольким словам, с использованием поисковых операторов и поиском по точной фразе. По умолчанию поиск с учетом синонимов и нечеткий поиск не производятся. Для выполнения поисков этих видов следует использовать соответствующие операторы.

В строке ввода допускается использовать следующие поисковые операторы (как для поиска в справке и синтакс-помощнике, так и для объекта встроенного языка [ПолнотекстовыйПоиск](#)):

Оператор	Пример выражения	Пояснение
Пробел И AND &	Бизнес Успех Бизнес И Успех Бизнес AND Успех Бизнес & Успех	Должно быть и слово «Бизнес», и слово «Успех»
ИЛИ OR ,	Бизнес ИЛИ Успех Бизнес OR Успех Бизнес Успех Бизнес , Успех	Должно быть хотя бы одно из слов «Бизнес» или «Успех»
НЕ NOT ~	Бизнес НЕ Успех Бизнес NOT Успех Бизнес ~ Успех	Должно быть слово «Бизнес», но не должно быть слова «Успех»
РЯДОМ/[±]n NEAR/[±]n	Пример 1: фен РЯДОМ/3 воздух Пример 2: фен РЯДОМ/+3 воздух Пример 3: фен РЯДОМ/-3 воздух	Поиск данных, содержащих в одном реквизите указанные слова с учетом морфологии на расстоянии n слов между словами. Знак указывает, в каком направлении от первого слова будет искаться второе слово («+» – после первого; «-» – до первого слова). Если знак не указан, то будут найдены данные, содержащие указанные слова на дистанции n слов друг от друга. Порядок слов не имеет значения. В примере 1 будут найдены данные, в которых слово «воздух» находится не более трех слов до или после слова «фен». В примере 2 будут найдены данные, в которых слово «воздух» находится не более трех слов после слова «фен». В примере 3 будут найдены данные, в которых слово «воздух» находится не более трех слов перед словом «фен»
РЯДОМ NEAR	Библиотека РЯДОМ имени РЯДОМ Достоевского	Краткая форма. Запрос найдет данные, в которых слова встречаются в одном реквизите не дальше чем на 8 слов друг от друга в любую сторону
" "	«администратор сети»	Поиск точной фразы (эквивалентно администратор РЯДОМ/+1 сети)

()	(технология изготовление) & (сыра творога)	Группировка слов (сколько угодно уровней вложенности)
*	арх*я арх* & документооб*	Поиск с использованием группового символа – поддерживается «*» в произвольном месте слова и в произвольном количестве. То есть запрос «арх*я» найдет «архивация», «археология»

Написание операторов **И (AND)**, **ИЛИ (OR)**, **НЕ (NOT)**, **РЯДОМ (NEAR)** допускается только в верхнем регистре. Операторы не используются как унарные (в начале строки поиска). Например, нельзя сделать выбор всех глав, в которых отсутствует указанный текст. Все символы в поле поиска, кроме символов поисковых операторов, букв и цифр, игнорируются.

При использовании объекта встроенного языка **ПолнотекстовыйПоиск** необходимо учитывать следующее:

Оператор	Пример выражения	Пояснение
*	арх* арх* & документооб*	Поиск с использованием группового символа – поддерживается только один символ «*», и он должен быть в конце слова. То есть запрос «арх*» найдет «архив», «археология»
#	#Система Система#2	Нечеткий поиск слов с заданным количеством отличий от указанного в строке поиска. Запрос «#Система» (эквивалентно запросу «Система#1») найдет «систама», «сивтема». Запрос «Система#2» найдет «ситтама», «сеттема». Только для полнотекстового поиска в данных!
!	!красный кафель	Поиск с учетом синонимов русского, английского и украинского языков. Оператор «!» ставится перед соответствующим словом. Пример: поиск «!красный кафель», найдет еще и «алый кафель», и «коралловый кафель»

Написание операторов **И (AND)**, **ИЛИ (OR)**, **НЕ (NOT)**, **РЯДОМ (NEAR)** допускается только в верхнем регистре. Операторы не используются как унарные (в начале строки поиска). Например, нельзя сделать выбор всех глав, в которых отсутствует указанный текст.

Для поиска спецсимволов, используемых в тексте, их следует заключать в кавычки. Например: **"РЯДОМ"**.

Приложение 6. Описание прав доступа

Данное приложение содержит описание всех прав доступа, которыми можно управлять при редактировании ролей в «1С:Предприятии». Символом "*" помечены права доступа, которые поддерживаются в мобильной платформе.

- **Automation (Automation)** – использование automation-сервера;
- **Администрирование (Administration)**:
 - Административные действия над всей информационной базой и редактирование списка пользователей (право для конфигурации). Требует наличия права **АдминистрированиеДанных**.
 - Возможность настройки параметров соединения (для внешнего источника данных).
 - Отображение списка лицензий, использовавшихся при работе с конфигурацией и информационной базой, в окне **О программе**.
 - Удаление областей данных (включая удаление всех областей сразу).
 - Возможность выполнения фонового обновления конфигурации базы данных на стороне клиента.
 - Возможность устанавливать имя профиля безопасности в диалогах управления расширениями.
- **АдминистрированиеДанных (DataAdministration)** – данное право регулирует административные действия над данными. К этим действиям относятся:
 - Просмотр записей журнала регистрации и получение значений отбора без ограничений.
 - Установка часового пояса информационной базы (области данных) – вызов метода `УстановитьЧасовойПоясИнформационнойБазы()`.
 - Создание начального образа подчиненного узла распределенной информационной базы (вызов метода `СоздатьНачальныйОбраз()`).
 - Обновление нумерации объектов (вызов метода `ОбновитьНумерациюОбъектов()`).
 - Выполнение методов объекта `СтандартноеХранилищеНастроекМенеджер` в тех случаях, если выполняется работа с настройками пользователя, отличными от текущего.
- **АдминистрированиеРасширенийКонфигурации (ConfigurationExtensionsAdministration)** – управляет доступом к интерфейсу управления расширениями (как интерактивному, так и программному) в режиме «1С:Предприятие». В безопасном режиме администрирование расширений невозможно.
- **АктивныеПользователи (ActiveUsers)** – просмотр списка активных пользователей.
- **ВводПоСтроке*** (`InputByString`) – использование режима ввода по строке.
- **ВебКлиент (WebClient)** – право запуска веб-клиента.
- **ВнешнееСоединение (ExternalConnection)** – внешнее соединение.
- **Вывод*** (`Output`) – вывод на печать, запись и копирование в буфер обмена.
- **Выполнение (Execute)** – выполнение задачи.
- **Добавление (Insert)** – добавление (основное).
- **ЖурналРегистрации (EventLog)** – право просмотра журнала регистрации.

- **Изменение (Update)** – изменение (основное).
- **ИзменениеСтандартнойАутентификации (StandardAuthenticationChange)** – пользователь имеет право изменять свои сохраненные параметры стандартной аутентификации (свойства **Аутентификация стандартная, Имя пользователя, Пароль**) внешнего источника данных.
- **ИзменениеАутентификацииОССеанса (SessionOSAuthenticationChange)** – пользователь имеет право изменять параметры аутентификации ОС внешнего источника данных в своих настройках и в параметрах сеанса.
- **ИзменениеСтандартнойАутентификацииСеанса (SessionStandardAuthenticationChange)** – пользователь имеет право изменять параметры стандартной аутентификации (свойства **Аутентификация стандартная, Имя пользователя, Пароль**) внешнего источника данных для текущего сеанса.
- **ИнтерактивнаяАктивация (InteractiveActivate)** – интерактивная активация.
- **ИнтерактивнаяОтменаПроведения* (InteractiveUndoPosting)** – интерактивная отмена проведения.
- **ИнтерактивнаяПометкаУдаления* (InteractiveSetDeletionMark)** – интерактивная пометка на удаление.
- **ИнтерактивнаяПометкаУдаленияПредопределенныхДанных* (InteractiveSetDeletionMarkPredefinedData)** – разрешает интерактивную пометку удаления предопределенных данных.
- **ИнтерактивноеВыполнение (InteractiveExecute)** – интерактивное выполнение задачи.
- **ИнтерактивноеДобавление* (InteractiveInsert)** – интерактивное добавление.
- **ИнтерактивноеИзменениеПроведенных* (InteractiveChangePosted)** – интерактивное редактирование проведенного документа. Если право не установлено, то пользователь не может удалить проведенный документ, установить пометку удаления, перепровести или сделать непроведенным. Форма такого документа открывается в режиме просмотра.
- **ИнтерактивноеОткрытиеВнешнихОбработок (InteractiveOpenExtDataProcessors)** – интерактивное открытие внешних обработок.
- **ИнтерактивноеОткрытиеВнешнихОтчетов (InteractiveOpenExtDataReports)** – интерактивное открытие внешних отчетов.
- **ИнтерактивноеПроведение* (InteractivePosting)** – интерактивное проведение.
- **ИнтерактивноеПроведениеНеОперативное (InteractivePostingRegular)** – интерактивное проведение (стандартными командами форм) документа в неоперативном режиме.
- **ИнтерактивноеСнятиеПометкиУдаления* (InteractiveClearDeletionMark)** – интерактивное снятие пометки на удаление.
- **ИнтерактивноеСнятиеПометкиУдаленияПредопределенныхДанных* (InteractiveClearDeletionMarkPredefinedData)** – разрешает интерактивно снимать пометку удаления с предопределенных данных.
- **ИнтерактивноеУдаление (InteractiveDelete)** – интерактивное непосредственное удаление.
- **ИнтерактивноеУдалениеПомеченных (InteractiveDeleteMarked)** – интерактивное удаление помеченных объектов.
- **ИнтерактивноеУдалениеПомеченныхПредопределенныхДанных* (InteractiveDeleteMarkedPredefinedData)** – разрешает интерактивное удаление помеченных предопределенных данных.

- **Интерактивное Удаление Предопределенных Данных** (`InteractiveDeletePredefinedData`) – разрешает интерактивное удаление предопределенных данных.
- **Интерактивный Старт** (`InteractiveStart`) – интерактивный старт бизнес-процесса.
- **Использование** (`Use`) – использование того или иного объекта конфигурации.
- **Монопольный Режим** (`ExclusiveMode`) – использование монопольного режима.
- **Обновление Конфигурации Базы Данных** (`UpdateDataBaseConfiguration`) – позволяет выполнять обновление конфигурации базы данных. Для выполнения обновления конфигурации на сервере или фонового обновления конфигурации на сервере достаточно наличия только этого права, право **Администрирование** в этом случае не требуется.
- **Отмена Проведения** (`UndoPosting`) – отмена проведения документов.
- **Получение** (`Get`) – получение значения, не хранящегося в базе данных.
- **Проведение** (`Posting`) – проведение документов.
- **Просмотр*** (`View`) – просмотр. В диалоге **Выбор типа** будут недоступны те типы, для которых у текущего пользователя отсутствует это право.
- **Редактирование*** (`Edit`) – редактирование.
- **Режим Все Функции*** (`AllFunctionsMode`) – управляет доступностью в главном меню команды **Все функции** и соответствующего пункта настроек клиентского приложения.
- **Сохранение Данных Пользователя** (`SaveUserData`) – управляет возможностью сохранения настроек пользователя. Если право выключено, то:
 - Недоступна кнопка **История**, обращение к истории из встроенного языка вызывает исключение, интерактивные операции записи не фиксируются в истории.
 - В панели системных команд недоступна команда вызова списка избранного. В формах **Получение ссылки** и **Переход по ссылке** недоступна кнопка **Добавить в избранное**. При обращении к избранному из встроенного языка вызывается исключение.
 - Недоступна настройка форм (отсутствует команда **Изменить форму**).
 - Недоступны команды настройки панели разделов, панели навигации и начальной страницы.
 - В отчетах недоступны команды сохранения пользовательских настроек и вариантов отчета. Не выдается предложение запомнить изменения, выполненные пользователем в текущем варианте отчета. Изменения не сохраняются. Команда выбора настроек отчета доступна только в том случае, если у отчета или конфигурации установлено хранилище пользовательских настроек отчета. При вызове метода `УстановитьТекущиеПользовательскиеНастройки()` выдается исключение.
 - Недоступны команды сохранения и восстановления данных форм (команды **Сохранить параметры** и **Восстановить параметры**). Не выполняется автоматическое сохранение данных форм.
 - Настройки окон (размер и положение) сохраняются только на время сеанса.
 - Настройки размера панелей главного окна сохраняются только на время сеанса.
 - При программном сохранении настроек выдается исключение.
 - Настройки клиентского приложения сохраняются только на время сеанса. Не отображается флажок **Устанавливать режим разрешения отладки при запуске** (для тонкого и толстого клиента). Не отображается флажок **Устанавливать режим низкой скорости соединения при запуске** (для веб-клиента).

- Не сохраняется настройка периода в списках. В диалоге настройки периода недоступен флажок [Использовать эту настройку периода при открытии](#).
- **Старт (Start)** – старт бизнес-процесса.
- **ТолстыйКлиент (ThickClient)** – право запуска толстого клиента.
- **ТонкийКлиент (ThinClient)** – право запуска тонкого клиента.
- **Удаление (Delete)** – удаление (основное).
- **УправлениеИтогами (TotalsControl)** – управление итогами регистра бухгалтерии и регистра накопления (установка периода, по который рассчитаны итоги, и пересчет итогов).
- **Установка (Set)** – установка значения, не сохраняемого в базе данных.
- **Чтение (Read)** – чтение (основное).

Приложение 7. Особенности поведения системы в различных режимах

7.1. Особенности режима низкой скорости соединения

При функционировании в режиме низкой скорости соединения имеется ряд особенностей поведения системы, которые перечислены ниже:

- Формы начальной страницы автоматически не отображаются. Для отображения рабочего стола следует нажать гиперссылку [Показать рабочий стол](#), расположенную в левом верхнем углу рабочей области рабочего стола. В интерфейсе [Такси](#) начальная страница отображается сразу.
- При создании формы происходит получение представлений ссылочных данных формы (не делается дополнительного вызова сервера для получения представлений после открытия формы).
- При создании формы происходит получение значений данных формы «через точку», отображаемых элементами формы (не делается дополнительного вызова сервера для получения этих значений после открытия формы).
- Оповещение сервера о закрытии формы (а также снятие блокировки с объекта, если он заблокирован) происходит в следующих случаях:
 - получение формы с сервера с помощью методов `ОткрытьФорму()`, `ПолучитьФорму()`;
 - вызов серверных (контекстных и внеконтекстных) методов формы;
 - закрытие 20 форм;
 - система находится в состоянии ожидания более 20 секунд;
 - при выполнении следующих команд динамического списка:
 - удаление элемента;
 - пометка на удаление элемента;
 - проведение или отмена проведения документа;
 - перенос элемента справочника в другую группу.
- Настройки сохраняются только в том случае, если они были изменены, и не сразу при закрытии формы, а отложено – при следующем открытии любой формы, или при накоплении 20 отложено сохраняемых настроек, или по прошествии 20 минут (при бездействии системы), или при закрытии приложения.
- Клиентское приложение не получает и не отображает картинки панели разделов.
- Выполняется кеширование начального состояния формы выбора, которое используется при повторном создании такой формы. Информация о форме удаляется из кеша, если:
 - форма не используется в течение последних 20 минут;
 - в форме выбора вызвана команда [Обновить](#);
 - на стороне клиента отображаются изменения данных объекта соответствующего типа.
- В поле ввода не выполняется формирование списка выбора во время ожидания в процессе ввода по строке.
- Для списков быстрого выбора и списков ввода по строке формируется специальный кеш данных.

Данные кеша сохраняются на время сеанса. Для каждого типа объектов хранится не более 20 последних списков. В кеше может храниться не более 200 списков.

Каждые 20 минут кеш очищается от устаревших данных, относящихся к объектам, по которым отображаются изменения на стороне клиента в этом сеансе.

Повторная операция выбора или ввода по строке с теми же параметрами, без перехода в другое поле, приводит к новому запросу данных с сервера.

- В режиме нормальной скорости соединения, поиск в динамическом списке (см. [здесь](#)) выполняется в процессе набора, со стандартной задержкой. В режиме медленной скорости соединения поиск выполняется после того, как строка поиска теряет фокус ввода.

7.2. Особенности режима совместимости

Ряд механизмов системы имеют различное поведение, которое зависит от состояния свойства конфигурации [Режим совместимости](#). В данном разделе перечислены такие различия в поведении.

- Свойство конфигурации [Основной режим запуска](#) не может быть установлено в значение [Управляемое приложение](#) одновременно с установкой свойства конфигурации [Режим совместимости](#) в значение [Версия 8.1](#). Проверка выполняется при обновлении конфигурации базы данных.
- В режиме совместимости с версией 8.1 для объектов метаданных, которые могли принадлежать подсистемам, свойство [Подсистемы](#) доступно из встроенного языка. Также в режиме совместимости с версией 8.1 доступно свойство конфигурации [Состав](#), которое заполняется ссылками на объекты, принадлежащие корневому элементу дерева подсистем. При выключении режима совместимости с версией 8.1 свойство конфигурации [Состав](#) будет очищено и станет недоступным в панели свойств конфигурации.
- В обычных формах заполнение новых объектов происходит аналогично версии 8.1, т. е. обработчик заполнения вызывается только при вводе на основании и вызове метода [Заполнить\(\)](#).
- В объектах метаданных метод [ПроверитьЗаполнение\(\)](#) становится недоступным, автоматическая проверка заполнения не выполняется.
- Вызов справки из формы выполняется аналогично версии 8.1, т. е. выдается диалог выбора раздела справки, если для данной формы отсутствует справочная информация.
- Таблицы журналов документов не содержат виртуальное поле [Тип](#).
- В конструкторе запроса в списке таблицы информационной базы отображаются таблицы и поля, на которые у текущего пользователя отсутствует право просмотра.
- Система компоновки данных не проверяет интерактивные права на таблицы.
- Картинки, выбранные из библиотеки картинок во внешний отчет или обработку, конвертируются во внешние картинки.
- Доступно свойство [Тип объектов](#) [ДоступноеПолеКомпоновкиДанных](#), [ДоступноеПолеОтбораКомпоновкиДанных](#) и [ДоступныйПараметрКомпоновкиДанных](#).
- Недоступно свойство [ВерсияДанных](#) для справочников, документов, планов видов характеристик, планов счетов, планов видов расчетов, планов обмена, бизнес-процессов и задач. Ограничения доступа к данным для поля [ВерсияДанных](#) проверяются на основе ограничений, установленных на поле [Ссылка](#).
- Установка прав [ИнтерактивноеДобавление](#), [ИнтерактивноеУдаление](#), [ИнтерактивнаяПометкаУдаления](#), [ИнтерактивноеСнятиеПометкиУдаления](#), [ИнтерактивноеУдалениеПомеченных](#), [ИнтерактивноеПроведение](#), [ИнтерактивноеПроведениеНеОперативное](#), [ИнтерактивнаяОтменаПроведения](#), [ИнтерактивноеИзменениеПроведенных](#), [ИнтерактивныйСтарт](#), [ИнтерактивнаяАктивация](#), [ИнтерактивноеВыполнение](#) не связана с установкой права [Редактирование](#).

- При выгрузке файлов конфигурации модуль обычного приложения выгружается в файл [Конфигурация.МодульПриложения.txt](#).
- Форматирование значений типов [Число](#), [Дата](#) и [Булево](#) происходит в соответствии с параметрами, заданными в региональных настройках информационной базы.
- В диалоге выбора цвета веб-цвета упорядочены по английским названиям.
- Свойствам табличного документа [Вывод](#) и [ОриентацияСтраницы](#) можно присваивать любые значения. При этом значения с неподходящим типом игнорируются.
- Для интерактивного выбора доступны картинки [ДокументОбъект](#), [ПланВидовХарактеристикОбъект](#), [РегламентныеЗадания](#), [СправочникОбъект](#), [БизнесПроцессОбъект](#), [ЗадачаОбъект](#), [ПланВидовРасчетаОбъект](#), [ПланОбменаОбъект](#), [ПланСчетовОбъект](#), [РегистрСведенийЗапись](#), [ВыбратьИзСписка](#), [ДобавитьЭлементСписка](#), [НоваяГруппа](#), [ИзменитьЭлементСписка](#), [УстановитьПометкуУдаленияЭлементаСписка](#), [УдалитьЭлементСпискаНепосредственно](#), [УдалитьЭлементСписка](#).
- Вид сравнения [Содержит](#) системы компоновки данных считает символы [_](#), [%](#) и [\[](#) как специальные символы.
- Операция языка запросов [ВЫРАЗИТЬ](#) возвращает строку фиксированной длины (с концевыми пробелами) при приведении к типу [Строка](#).
- Настройки параметров печати у отдельных табличных документов (объектов) не различаются и меняются одновременно (даже если совпадает ключ параметров печати).
- Запрос, в котором указано ключевое слово [РАЗЛИЧНЫЕ](#), а в предложении [УПОРЯДОЧИТЬ ПО](#) указано выражение, отсутствующее в списке выборки, считается корректным, и при исполнении такого запроса не будет выдана ошибка.
- Поля остатка вложенного запроса во внешнем запросе не считаются полями остатка, и итоги по ним рассчитываются как по обычным полям.
- Размер реквизита метаданных типа [Строка фиксированного размера](#) ограничен 1024 символами.
- Для новых регистров накопления и бухгалтерии по умолчанию не устанавливается признак разделения итогов.
- Изменяется представление полных имен модулей в технологической информации (технологический журнал, журнал регистрации и т. д.) – в именах отсутствует конкретизация модуля.
- Проверка значений блокировки на соответствие типу поля блокировки в элементах управляемых блокировок при выполнении метода [Заблокировать\(\)](#) выполняется только для первого значения блокировки для каждого элемента.
- В конструкторе формы и конструкторе макета при изменении синонима изменяется имя, а при изменении имени изменяется синоним, если после открытия конструктора не изменялись значения этих полей, созданные автоматически.
- Редактор картинок не поддерживает работу с альфа-каналом. Не происходит преобразования картинок в формат PNG после завершения редактирования.
- Выделяется заголовок активной колонки таблицы в форме.
- Нажатие клавиш [Tab](#) или [Shift + Tab](#) в таблице формы, не содержащей строк или имеющей режим выделения [Строка](#), приводит к активизации, соответственно, следующей или предыдущей колонки таблицы. Активизация следующего/предыдущего элемента формы выполняется только после достижения крайней колонки таблицы.
- У табличного документа делаются доступными свойства [ЦветФонаГруппировки](#), [ЦветТекстаГруппировки](#), [ЦветФонаЗаголовков](#), [ЦветТекстаЗаголовков](#).

- Если для элемента формы выбрана нестандартная рамка или цвет рамки, а также для элементов [Кнопка](#), [Поле ввода](#) и [Полоса регулирования](#) выбран нестандартный цвет фона кнопки, то элемент формы отображается заданным цветом и рамка не имеет скруглений.
- Не вызывается обработчик формы [ПередЗакрытием](#), если модальная форма закрывается по тайм-ауту.
- При выгрузке журнала регистрации в XML-формате, в поле [UserName](#) выводится полное имя пользователя.
- Пользователь без административных прав может изменять свойство [ЗапрещеноИзменятьПароль](#) объекта [ПользовательИнформационнойБазы](#).
- При получении макета на сервере и во внешнем соединении в качестве языка макета устанавливается язык конфигурации по умолчанию.
- Свойство [Путь к данным](#) пользовательского поля системы компоновки данных формируется на английском языке.
- При открытии обычной формы записи регистра сведений значение свойства [Период](#) устанавливается равным текущей дате, только если это свойство не заполнено и форма записи открывается из формы списка. В остальных случаях свойство не заполняется.
- Изменяется состав таблиц, описывающих регламентные задания:
 - Используется режим совместимости – все регламентные задания хранятся в одной таблице.
 - Не используется режим совместимости – каждое регламентное задание хранится в отдельной таблице.
- Изменяется состав таблиц, хранящих константы:
 - Используется режим совместимости – для хранения констант используется одна таблица. Для регистрации изменений используется также одна таблица.
 - Не используется режим совместимости – каждая константа хранится в собственной таблице. Регистрация изменений каждой константы также ведется в собственной таблице.
- Изменяется состав таблиц, хранящих настройки регистров накопления:
 - Используется режим совместимости – настройки регистров накопления хранятся в одной таблице.
 - Не используется режим совместимости – настройки каждого регистра хранятся в собственной таблице.
- Изменяется состав таблиц, хранящих настройки регистров бухгалтерии:
 - Используется режим совместимости – настройки регистров бухгалтерии хранятся в одной таблице.
 - Не используется режим совместимости – настройки каждого регистра хранятся в собственной таблице.
- Изменилось включение справки по подсистеме в содержание справочной информации:
 - Используется режим совместимости – справка по подсистеме отображалась в содержании справки в том случае, если у какого-либо из объектов, входящих в состав подсистемы, было установлено свойство [Включать в содержание справки](#), а у самой подсистемы это свойство было выключено.
 - Не используется режим совместимости – включение справки по подсистеме в содержание справки полностью зависит от состояния свойства подсистемы [Включать в содержание справки](#) и

не зависит от этого свойства объектов, входящих в состав подсистемы.

- Изменилось количество ресурсов, выводимых в диаграмму системой компоновки данных:
 - Используется режим совместимости – в диаграмму выводится только один ресурс (первый).
 - Не используется режим совместимости – в диаграмму выводится более одного ресурса.
- Изменена возможность группировки по полям-ресурсам системы компоновки данных:
 - Используется режим совместимости – группировка по полям-ресурсам не поддерживается.
 - Не используется режим совместимости – возможна группировка по полям-ресурсам.
- Изменена диагностика ключевых значений, идентифицирующих отображаемую строку, для динамического списка с установленным произвольным запросом:
 - Используется режим совместимости – ошибки, возникающие при получении данных динамического списка, не отображаются.
 - Не используется режим совместимости – если в выборке обнаружены дублирующиеся ключевые значения, то отображается предупреждение и отображение данных становится невозможным.
- Изменена обязательность параметра **Период** метода менеджера периодического регистра сведений **Получить ()**:
 - Используется режим совместимости – параметр не является обязательным, и результат работы метода не определен.
 - Не используется режим совместимости – параметр является обязательным.
- Изменилось поведение системы при автоматическом определении размера элемента формы в том случае, если в свойствах элемента формы задан размер шрифта, отличный от размера шрифта по умолчанию:
 - Используется режим совместимости – размер шрифта элемента не учитывается при расчете размера элемента формы.
 - Не используется режим совместимости – размер шрифта элемента учитывается при определении размеров элемента. Изменение шрифта, выполняемое с помощью условного оформления формы или динамического списка, не оказывает влияния на размер элементов формы.
- Изменено текстовое представление стандартного реквизита **ВерсияДанных**:
 - Используется режим совместимости – представление равно пустой строке.
 - Не используется режим совместимости – представлением является строка из 12 пробелов.
- При передаче значения от реквизита к элементу формы изменено обновление текста в поле ввода:
 - Используется режим совместимости – обновление происходит всегда, кроме случая, когда в обработчике события **АвтоПодбор** происходит установка значения того же типа, что и тип имеющегося в поле ввода значения. Свойство **ОбновлениеТекстаРедактирования** игнорируется.
 - Не используется режим совместимости – управление обновлением возможно с помощью свойства поля ввода **ОбновлениеТекстаРедактирования** и метода **ОбновитьТекстРедактирования ()** расширения формы поля ввода.
- При одновременном использовании методов **УстановитьЗначение ()** и **ИспользоватьИзИсточникаДанных ()** объекта **ЭлементБлокировкиДанных**:
 - Используется режим совместимости – приоритетным является значение, установленное методом

Использовать Из Источника Данных ().

- Не используется режим совместимости – вызывается исключение.
- Изменены ключи объектов, предназначенных для хранения настроек командного интерфейса рабочего стола:
 - Используется режим совместимости:
 - Для настроек панели навигации используется ключ `Подсистема.desktop/ПанельНавигации/НастройкиКомандногоИнтерфейса`.
 - Для настроек панели действий используется ключ `Подсистема.desktop/ПанельДействий/НастройкиКомандногоИнтерфейса`.
 - Не используется режим совместимости:
 - Для настроек панели навигации используется ключ `РабочийСтол/ПанельНавигации/НастройкиКомандногоИнтерфейса`.
 - Для настроек панели действий используется ключ `РабочийСтол/ПанельДействий/НастройкиКомандногоИнтерфейса`.
- Изменен алгоритм автоматического заполнения списка реквизитов формы, заполнение которых необходимо проверить:
 - Используется режим совместимости – реквизиты, отключенные с помощью функциональных опций, попадают в список проверяемых реквизитов.
 - Не используется режим совместимости – реквизиты, отключенные с помощью функциональных опций, не попадают в список проверяемых реквизитов.
- Чтение данных для таблиц управляемой формы, связанных с динамическими списками, выполняемое при открытии формы:
 - Используется режим совместимости – выполняется чтение для всех таблиц, связанных с динамическими списками, в том числе и для тех, которые не видны пользователю.
 - Не используется режим совместимости – чтение выполняется только для таблиц, которые отображаются пользователю при открытии формы.
- Преобразование свойства **Имя в источнике данных** поля таблицы внешнего источника данных при помещении в SQL-запрос:
 - Используется режим совместимости – значение свойства всегда заключается в двойные кавычки, вне зависимости от состава символов в имени.
 - Не используется режим совместимости – значение свойства заключается в двойные кавычки только в том случае, если поле содержит спецсимволы и не заключено в одинарные кавычки.
- Определение типа колонки для вычисляемого поля при выводе результата компоновки данных в коллекцию значений:
 - Используется режим совместимости – тип колонки определяется автоматически, на основании выражения вычисляемого поля.
 - Не используется режим совместимости – в качестве типа значения колонки устанавливается тип значения, указанный для вычисляемого поля. Если тип значения не указан, то тип определяется на основании выражения вычисляемого поля.
- Вывод результатов компоновки данных в коллекцию значений, в случае, когда одно поле компоновки данных попадает в различные группировки:

- Используется режим совместимости – поля различных группировок, ссылающихся на одно поле компоновки данных, попадали в разные колонки коллекции.
- Не используется режим совместимости – поля различных группировок, ссылающихся на одно поле компоновки данных, попадали в разные колонки коллекции.
- Формирование произвольного представления объекта данных:
 - Используется режим совместимости с версией 8.2.16 – не поддерживается формирование произвольных представлений.
 - Не используется режим совместимости – поддерживается формирование произвольных представлений.
- Вывод дополнительных данных для иерархических записей:
 - Используется режим совместимости с версией 8.2.16 – используются данные для иерархической записи.
 - Не используется режим совместимости – используются данные для детальных записей.
- Получение значения функциональной опции, для которой установлено получение в привилегированном режиме:
 - Используется режим совместимости с версией 8.3.2 – в безопасном режиме выдается ошибка в том случае, если отсутствуют права доступа на объект, хранящий значение получаемой функциональной опции.
 - Не используется режим совместимости – проверка прав не выполняется.
- Использование свойства **ОсновныеРоли**:
 - Используется режим совместимости с версией 8.3.3 – допускается включение в свойство только одной роли. При переключении в режим совместимости с версией 8.3.3 проверяется, что в свойстве **ОсновныеРоли** указана только одна роль.
 - Не используется режим совместимости – допускается отмечать несколько ролей.
- Доступные настройки динамического списка, отключенные с помощью функциональных опций:
 - Используется режим совместимости с версией 8.3.3 – не удаляются из списка доступных настроек.
 - Не используется режим совместимости – удаляются из списка доступных настроек. В некоторых случаях, в случае наличия настроек, ссылающихся на поля, отключенные с помощью функциональных опций, может диагностироваться ошибка **Поле не обнаружено**.
- Автоматическое имя кнопки, связанной с командой **Отмена** расширения управляемой формы отчета, управляемой формы варианта отчета и управляемой формы настроек:
 - Используется режим совместимости с версией 8.3.3 – для кнопки назначалось имя **ФормаОтмена (FormCancel)**.
 - Не используется режим совместимости – для кнопки назначается имя **ФормаОтменитьРедактирование (FormCancelEdit)**.
- Вычисление выражения системы компоновки данных:
 - Используется режим совместимости с версией 8.3.4 – вычисление выражения выполняется с правами текущего пользователя.
 - Не используется режим совместимости – при вычислении выражения система переключается в

безопасный режим (см. [здесь](#)).

- Перетаскивание в таблице управляемой формы:
 - Используется режим совместимости с версией 8.3.4 – перетаскивание начинается из любой ячейки таблицы. Если ячейка, откуда начинается перетаскивание, доступная – вначале происходит активизация ячейки и потом – собственно перетаскивание. Если ячейка недоступная – выполняется перетаскивание текущей активной ячейки таблицы управляемой формы.
 - Не используется режим совместимости – перетаскивание можно осуществить только из доступной ячейки таблицы управляемой формы.
- Генерация макета компоновки данных из predetermined макетов:
 - Используется режим совместимости с версией 8.3.4 – при генерации макета компоновки данных из predetermined макетов полностью копируются выражения расшифровки.
 - Не используется режим совместимости – при генерации макета компоновки данных из predetermined макетов удаляются выражения расшифровки, которые содержат поля, недоступные по функциональным опциям.
- Параметр **Текст** условного оформления системы компоновки данных, которое задается для области использования **в заголовке полей**:
 - Используется режим совместимости с версией 8.3.4 – не применяется к заголовкам полей группировки.
 - Не используется режим совместимости – применяется к заголовкам полей группировки.
- Поведение параметра вывода системы компоновки данных **РасположениеГруппировок**:
 - Используется режим совместимости с версией 8.3.4 – общий итог группировки не выводился, если параметр системы компоновки данных **РасположениеГруппировок**, для выбранной группировки, имел значение **Конец** или **Нет**.
 - Не используется режим совместимости – не влияет на вывод общих итогов.
- Поведение генератора макета компоновки данных для вложенных отчетов:
 - Используется режим совместимости с версией 8.3.4 – группировки родительского отчета не учитывались при выводе вложенных отчетов.
 - Не используется режим совместимости – для макетов вложенных отчетов системы компоновки данных всегда генерирует свойство **ВертикальныйУровень**. К значению данного свойства прибавляется уровень родительского отчета, в котором находится вложенный отчет.
- Добавление строк в таблицу формы:
 - Используется режим совместимости с версией 8.3.4 – строка в таблице автоматически удалялась, если эта строка была добавлена при окончании редактирования предыдущей строки или при нажатии клавиши **Стрелка вниз** и в том случае, если после окончания добавления эта строка не изменялась интерактивно.
 - Не используется режим совместимости – строка в таблице автоматически удаляется, если эта строка была добавлена при окончании редактирования предыдущей строки или при нажатии клавиши **Стрелка вниз** и в том случае, если после окончания добавления эта строка не изменялась ни программно, ни интерактивно.
- Ограничение максимальной длины значения свойства управляемой формы **КлючНазначенияИспользования**:
 - Используется режим совместимости с версией 8.3.4 – ограничения длины не установлено.

- Не используется режим совместимости – длина значения для данного реквизита ограничена 128 символами.
- Дополнительные параметры методов `НайтиПомеченныеНаУдаление()`, `НайтиПоСсылке()`, `УдалитьОбъекты()`:
 - Используется режим совместимости с версией 8.3.4 – дополнительные параметры не используются. Их использование приведет к ошибке времени выполнения.
 - Не используется режим совместимости – дополнительные параметры используются.
- Автоматическое формирование ключа настройки отображения формы:
 - Используется режим совместимости с версией 8.3.4 – ключ объекта формируется как `<Полное имя формы>/НастройкиФормы`.
 - Не используется режим совместимости – ключ объекта формируется как `<Полное имя формы>/<Ключ назначения использования>/НастройкиФормы`. Подробнее см. [здесь](#).
- Доступные настройки отчетов и динамических списков:
 - Используется режим совместимости с версией 8.3.4 – удаляются элементы настроек, которые содержат ссылки на недоступные поля, отключенные функциональными опциями, недоступные в соответствии с правами доступа.
 - Не используется режим совместимости – удаляются элементы настроек, которые содержат ссылки на недоступные поля, отключенные функциональными опциями, недоступные в соответствии с правами доступа или связанные с более несуществующими полями схемы компоновки данных.
- Кнопка управляемой формы, связанная с командой:
 - Используется режим совместимости с версией 8.3.5 – всегда отображается доступной, вне зависимости от доступности команды.
 - Не используется режим совместимости – кнопка становится недоступной, если для текущего пользователя команда, связанная с кнопкой, также недоступна.
- Поведение управляемой формы при невозможности создать источник доступных настроек динамического списка:
 - Используется режим совместимости с версией 8.3.5 – при невозможности создать источник доступных настроек компоновки данных для динамического списка, никаких ошибок не генерируется, и работа управляемой формы продолжается так, как будто источник данных динамического списка не содержит ни каких доступных полей. В результате все элементы формы, связанные с данными динамического списка, удаляются с управляемой формы.
 - Не используется режим совместимости – при невозможности создать источник доступных настроек динамического списка генерируется исключение и создание формы не происходит.
- Заполнение значений независимых и совместных разделителей при записи набора записей регистра сведений:
 - Используется режим совместимости с версией 8.3.5 – значения разделителей, в состав которых входит регистр сведений, всегда заменяются на значения разделителей, установленных в текущем сеансе.
 - Не используется режим совместимости – все значения по умолчанию для независимых и совместных разделителей, в состав которых входит регистр сведений, заменяются на значения разделителей, установленных в текущем сеансе.

Если значение разделителя не совпадает со значением по умолчанию для типа разделителя или не равно значению разделителя, установленного в текущем сеансе – генерируется ошибка.

- Поведение системы при работы с объектом `xBase` в случае возникновения ошибок работы с файлом `.dbf`:

- Используется режим совместимости с версией 8.3.5 – в случае возникновения ошибки формировались диалоговые окна.
- Не используется режим совместимости – в случае возникновения ошибки генерируется исключение.

- Работа обработчиков ожидания (подключенных с помощью методов `ПодключитьОбработчикОжидания()`) во время временной блокировки (с помощью команды `Временная блокировка` или с помощью метода `ЗаблокироватьРаботуПользователя()`) интерфейса клиентского приложения:

- Используется режим совместимости с версией 8.3.5 – работа обработчиков не зависит от наличия временной блокировки клиентского приложения.
- Не используется режим совместимости – обработчики ожидания не вызываются.

- Применение значения, указанного у включенного параметра условного оформления `Текст`:

- Используется режим совместимости с версией 8.3.5 – не применялось к полям предопределенных макетов группировок системы компоновки данных.
- Не используется режим совместимости – применяется к полям предопределенных макетов группировок и полей отчетов системы компоновки данных.

- Поведение процессора вывода при выгрузке результата компоновки данных в коллекцию значений при наличии колонок с дублирующимися именами:

- Используется режим совместимости с версией 8.3.5 – уникальность колонок не проверялось, колонки создавались всегда.
- Не используется режим совместимости – проверяется уникальность имени колонки в коллекции. Колонки с повторяющимися именами повторно не создаются.

- Выгрузка результата компоновки данных в коллекцию значений при наличии вложенных отчетов:

- Используется режим совместимости с версией 8.3.5 – выгрузка невозможна. При попытке выполнить выгрузку генерируется исключение.
- Не используется режим совместимости – выгрузка возможна.

- Получение параметров прокси-сервера из переменных окружения в ОС Linux:

- Используется режим совместимости с версией 8.3.5 – переменные среды не использовались при создании объекта `ИнтернетПрокси`. Если при создании объектов `НТТРСоединение` и `ФТПСоединение` не были указаны параметры прокси-сервера, при выполнении методов объектов производилась попытка определения параметров прокси-сервера из переменных окружения.
- Не используется режим совместимости – При создании объекта `ИнтернетПрокси` используются значения переменных окружения ОС `http_proxy`, `https_proxy`, `ftp_proxy`, `ftps_proxy`, `all_proxy` в тех случаях, если параметр конструктора `ПроксиПоУмолчанию` имеет значение `Истина` и система не может обнаружить файл `inetcfg.xml`.

Методы объектов `НТТРСоединение` и `ФТПСоединение` не пытаются определить настройки прокси-сервера, а используются те настройки, которые находятся в свойстве `Прокси` соответствующего объекта.

- Экспорт примечаний табличного документа в файлы формата Microsoft Excel 2007-2010:

- Используется режим совместимости с версией 8.3.5 – примечания выгружались как рисунки

таблицы Excel.

- Не используется режим совместимости – примечания выгружаются как примечания таблицы Excel.
- Получение полей детальных записей в агрегатных функциях для иерархических группировок:
 - Используется режим совместимости с версией 8.3.5 – использовались поля иерархического набора данных, если в этом наборе данных доступны соответствующие поля.
 - Не используется режим совместимости – используются поля детального набора данных. Для получения внутри агрегатных функций значения иерархической записи можно использовать функцию `ВычислитьВыражение()`.
- Подстановка параметра шаблона ограничения доступа, указанного в комментариях:
 - Используется режим совместимости с версией 8.3.5 – заменяется на фактическое значение.
 - Не используется режим совместимости – не заменяется.
- Применение механизма условного оформления системы компоновки данных к полям группировок и детальных записей:
 - Используется режим совместимости с версией 8.3.5 – при оформлении ячейки, в которую выводится поле родительской группировки, используются выражения полей, полученных для родительской группировки.
 - Не используется режим совместимости – при оформлении ячейки, в которую выводится поле родительской группировки, используются выражения полей, полученных для текущей группировки.
- Загрузка свойства расширения таблицы формы для динамического списка `Период` и свойства `Отображение` таблицы формы, связанной с динамическим списком, при открытии формы:
 - Используется режим совместимости с версией 8.3.5 – свойство расширения таблицы формы для динамического списка `Период` загружается из пользовательских настроек до выполнения обработчика события управляемой формы `ПриСозданииНаСервере`.

Свойство `Отображение` таблицы формы, связанной с динамическим списком, загружается из настроек на клиенте при открытии формы.

 - Не используется режим совместимости – загружаются из настроек одновременно с загрузкой пользовательских настроек динамического списка.
- Поведение системы в том случае, если в обработчике события управляемой формы `ПриОткрытии` параметру `Отказ` установлено значение `Истина`:
 - Используется режим совместимости с версией 8.3.5 – обработчик события `ПриЗакрытии` и оповещение, указанное в свойстве управляемой формы `ОписаниеОповещенияОЗакрытии`:
 - В тонком клиенте вызываются;
 - В веб-клиенте не вызываются.
 - Не используется режим совместимости – не вызывается обработчик события `ПриЗакрытии` и оповещение, установленное в свойстве управляемой формы `ОписаниеОповещенияОЗакрытии`.
- История выбора при вводе (см. [здесь](#)) для перечислений:
 - Используется режим совместимости с версией 8.3.5 – история выбора не отображается и не сохраняется.

- Не используется режим совместимости – отображение и сохранение истории выбора управляется свойством `ИсторияВыбораПриВводе`.
- Обращение (во встроенных механизмах платформы) к свойствам метода, написанного на встроенном языке, в том случае, если изменяется состав свойств объекта, переданного в качестве параметра этого метода:
 - Используется режим совместимости с версией 8.3.5 – обращение выполняется по номеру свойства, определенного в момент начал вызова. Если номер свойства изменяется – обращение не будет выполнено корректно.
 - Не используется режим совместимости – обращение выполняется по имени свойства. Если изменяется имя свойства – обращение не будет выполнено корректно.

7.3. Особенности работы веб-клиента

- В веб-браузерах Google Chrome и Safari не поддерживается:
 - Программная вставка из буфера обмена, а начиная с версии Google Chrome 6 и Safari 5 – любые операции программного доступа к буферу обмена. Доступны только команды работы с буфером обмена с помощью клавиатуры (а не контекстного меню).
 - Автоматическая аутентификация операционной системы.
 - Возможность изменения колонтитулов при печати.
- Не поддерживается расширение работы со средствами криптографии в веб-браузере Safari.
- ОС MacOS X поддерживают ввод паролей только с помощью букв латинского алфавита и цифр. ОС iOS, до версии 5.x (включая), поддерживает ввод паролей только с помощью букв латинского алфавита и цифр. В версии ОС iOS 6 допускается использовать в паролях символы национальных алфавитов. Для обеспечения совместимости между различными версиями ОС, а также при необходимости использовать прикладное решение с помощью веб-браузера Safari, рекомендуется задавать пароли пользователей с использованием цифр и букв только латинского алфавита.
- Работа с COM-объектами поддерживается только при работе в веб-браузере Microsoft Internet Explorer.
- При добавлении страницы с веб-приложением в [Закладки](#) веб-браузера Mozilla Firefox с помощью команды [Добавить страницу в закладки](#) следует вручную исправить добавленную ссылку. Для этого необходимо открыть окно редактирования свойств добавленной закладки и в поле адреса удалить текст от начала строки до `http://`, либо воспользоваться перетаскиванием иконки приложения (слева от адреса страницы) в меню [Закладки](#).
- Не поддерживается прерывание пользователем работы модуля. Во время выполнения модуля вызов метода `ОбработкаПрерыванияПользователя()` игнорируется.
- Не поддерживается метод `ЗаблокироватьРаботуПользователя()`.
- В функции `Вычислить()` не диагностируется ошибка [Обращение к процедуре как к функции](#).
- При возникновении ошибки в выражении, переданном в функцию `Вычислить()`, ошибка диагностируется как ошибка вызова метода, а не как ошибка в переданном выражении.
- В веб-браузере Google Chrome не поддерживается отладка кода на встроенном языке в обработчиках событий `ПередЗавершениемРаботыСистемы` и `ПриЗавершенииРаботыСистемы`.
- Во время выполнения длительных операций не происходит изменения вида курсора.
- Не поддерживается горизонтальная прокрутка (с помощью колеса мыши и нажатой клавиши [Shift](#)) во всех элементах формы.

- В веб-браузерах Microsoft Internet Explorer и Mozilla Firefox не обрабатываются клавиши, нажатые во время серверного вызова. В веб-браузерах Google Chrome и Safari эти нажатия обрабатываются после завершения серверного вызова.
- Не поддерживается определение горячей клавиши с помощью символа & в текстах команд, пунктов меню и т. д. Символ &, указанный в тексте, игнорируется и не отображается в интерфейсе.
- При работе в веб-браузере Safari при длительном выполнении запроса сервером может происходить ошибка.
- Не рекомендуется выполнять программное открытие и закрытие одной или нескольких форм в одном обработчике. Такие действия должны быть разнесены по времени. Рекомендуется закрытие форм выполнять в обработчике ожидания.
- Если форма, которая открыта во вспомогательном окне, закрывается нажатием системной кнопки закрытия окна («крестик» в правом верхнем углу формы) или нажатием **Alt + F4**, то не происходит вызов обработчика `ПередЗакрытием` модуля закрываемой формы.
- Если форму, которая открывается во вспомогательном окне, закрыть нажатием системной кнопки закрытия окна («крестик» в правом верхнем углу формы) до того, как она полностью отображена на экране, возможно возникновение ошибки веб-браузера.
- При изменении значений реквизитов формы вне модуля этой формы, сделанные изменения не отображаются элементами формы. Для отображения изменений необходимо вызвать метод `ОбновитьОтображениеДанных()`.
- В веб-браузере Google Chrome не производится активизация свернутого окна при выдаче предупреждения, относящегося к содержимому данного окна.
- Если в веб-браузере Google Chrome свернуть форму измененного объекта, а затем закрыть основное окно приложения, то будет выдано предупреждение о несохраненных данных, но форма с этими данными не будет активизирована.
- Если необходимо нормально отобразить текст элемента формы при указании большого размера шрифта, следует указывать не только размер, но и гарнитуру используемого векторного шрифта (TrueType-шрифт).
- Формирование представления значения (для отображения в поле ввода), которое передается в параметре `ВыбранноеЗначение` обработчика `ОбработкаВыбора`, происходит после того, как переданное значение приводится к типу значения реквизита, который отображается полем ввода.
- Если высота кнопки превышает 5 единиц (при стандартном размере шрифта), то кнопка отображается без перехода цветов (градиента).
- Если в перечне кнопок функции глобального контекста `Вопрос()` нет кнопки со значением `КодВозвратаДиалога.Отмена`, то попытка закрыть диалог с помощью системной кнопки закрытия («крестик» в правом верхнем углу формы) приведет к повторному открытию диалога.
- Функциональность панели состояния ограничена. Во время выполнения модуля обновление состояния зависит от используемого веб-браузера (подробнее см. [здесь](#)).
- Отображение поля индикатора отличается от тонкого клиента. В частности отличается внешний вид поля индикатора, у которого свойство `Отображение` установлено в значение `Прерывистый` или `ПрерывистыйНаклонный`.
- Функциональность объекта `Дендрограмма` ограничена.
- Функциональность объекта `ГрафическаяСхема` ограничена.
- Функциональность объекта `ГеографическаяСхема` ограничена.
- Для объекта `ДиаграммаГанта` ограничены возможности интерактивной настройки.

- Не поддерживается получение системной информации о клиентском компьютере.
- Не поддерживается работа с объектами `ЧтениеТекста`, `ЗаписьТекста`, `ЧтениеZipФайла`, `ЗаписьZipФайла`, `XBase`. Рекомендуется выполнять работу с такими файлами на стороне сервера.
- Не поддерживается работа с XML и XDTO. Рекомендуется выполнять работу с такими файлами на стороне сервера.
- В веб-браузерах Google Chrome и Safari реализован внутренний буфер обмена:
 - Данные во внутреннем буфере обмена актуальны во время сеанса.
 - Все значения, помещаемые в буфер обмена операционной системы, также дублируются во внутренний буфер обмена.
 - Команды `M+` и `M-` изменяют значение, находящееся во внутреннем буфере обмена. Полученный результат помещается в буфер обмена операционной системы.
 - Внутренний буфер обмена в калькуляторе отображается в ячейке «1». Команды калькулятора `MR`, `M+` и `M-` работают со значением внутреннего буфера обмена. Полученный результат помещается в буфер обмена операционной системы.
- Не поддерживается диалог настройки калькулятора.
- Функциональность интерактивного изменения табличного документа ограничена.
- Интерактивные команды, настраивающие отображение табличного документа, доступны в ограниченном объеме.
- Правила переноса текста в ячейке табличного документа зависят от используемого веб-браузера.
- Не поддерживается фоновая картинка табличного документа.
- Не поддерживается узор ячейки табличного документа.
- Не поддерживается просмотр и печать табличного документа в черно-белом режиме.
- Табличный документ, напечатанный из различных веб-браузеров, может иметь разные размеры.
- В ячейку таблицы управляемой формы выводятся только те строки многострочного текста, которые целиком помещаются в ячейку по высоте. Если строка отображается не полностью, в конце строки отображается символ "...". В веб-браузере Microsoft Internet Explorer 8 многоточие в конце строки не отображается.
- Если в результате изменения источника данных таблицы в таблице изменяется текущая строка и одновременно изменяется значение свойства элемента формы, которое приводит к перестроению формы, то порядок вызова обработчика `ПриИзмененииСтроки` и момент перезагрузки формы не определены.
- В формах, в которых предполагается вертикальная прокрутка, у таких элементов, как таблица, табличный документ, диаграмма и т. д., необходимо выключать растягивание по вертикали.
- Метод `ПолучитьHTML()` объекта `ФорматированныйДокумент` возвращает пустой список картинок.
- Выделение текста, выполненное в поле форматированного документа, скрывает цветовое оформление текста (цвет текста и цвет фона) при работе в веб-браузере Microsoft Internet Explorer и тонком клиенте.
- Не поддерживается открытие из файла табличного документа, содержащего отчет, в расшифровках которого присутствуют ссылочные значения.
- В табличном документе, поле табличного документа и таблице формы не поддерживается прокрутка

с помощью нажатия колеса мыши.

- Если у ячейки табличного документа установлено выравнивание по правому краю и текст полностью не помещается в ячейке, то помещающаяся часть текста выравнивается по левому краю.
- В веб-браузерах Microsoft Internet Explorer 6.0 и 7 не поддерживается отображение объединенных ячеек табличного документа, у которых указана рамка слева и не указана рамка снизу. Рекомендуется указывать рамку снизу для таких ячеек.
- В веб-браузере Microsoft Internet Explorer 7 для табличного документа, в котором используются ячейки разной ширины в одной колонке и используются текст и рисунок в одной ячейке, вывод на печать выполняется с искажениями.
- В веб-браузерах Google Chrome и Safari при определенном сочетании объединений ячеек и рамок ячеек в табличном документе возможно неправильное отображение рамки.
- Для двух соседних (по горизонтали) ячеек табличного документа соприкасающаяся граница будет оформлена такой же линией, что и правая граница левой ячейки, если табличный документ отображается в веб-браузере Microsoft Internet Explorer 6.0.
- Для оформления ячейки табличного документа доступны только следующие типы линий: **Сплошная**, **Точечная** и **НетЛинии**. Другие типы линий заменяются доступными следующим образом:
 - тип линии **Двойная** заменяется на **Сплошная**;
 - типы линий **РедкийПунктир**, **ЧастыйПунктир** и **БольшойПунктир** заменяются типом **Точечный**.
- В веб-браузерах Microsoft Internet Explorer и Google Chrome точечная рамка смежных ячеек в некоторых случаях может отображаться как сплошная.
- Для правильной печати из веб-браузеров Google Chrome и Mozilla Firefox (версии 20 и выше) табличных документов, содержащих точечные линии, рекомендуется устанавливать у этих табличных документов свойство **ТочностьПечати** в значение **Точная**. Аналогичная рекомендация применяется в случае печати из веб-браузеров Google Chrome и Mozilla Firefox (версии 20 и выше) табличных документов, для которых нужно обеспечить точную размерность всех элементов, выводимых на печать.
- Не поддерживается режим сравнения файлов.
- Не поддерживается список последних открытых файлов.
- Не поддерживается перетаскивание файлов в формы веб-клиента.
- В веб-клиенте используется ограниченный набор шрифтов, доступных в используемых операционных системах. Другие шрифты не могут выбираться в диалоге выбора шрифта в веб-клиенте, а при использовании замещаются на поддерживаемые шрифты.
- Доступ к главному меню выполняется с помощью клавиши **F10**, доступ с помощью одиночного нажатия клавиши **Alt** не поддерживается.
- Нажатие комбинации **Ctrl + W** в веб-браузере Google Chrome приводит к закрытию вспомогательного окна, а не к выделению слова.
- Нажатие комбинации **Ctrl + F4** в веб-браузере Google Chrome приводит к закрытию текущей вкладки. Если вкладка последняя, выполняется закрытие текущего окна веб-браузера.
- Нажатие комбинации **Ctrl + N** в веб-браузере Google Chrome приводит к созданию нового окна веб-браузера, а не к открытию меню создания нового документа.
- Не поддерживается установка в веб-браузере масштаба отображения, отличного от 100 %, или использование нестандартного dpi.

- Переключение окон с помощью клавиатуры имеет ряд особенностей (в зависимости от веб-браузера):

- **Microsoft Internet Explorer:**

- Версия 6.0 – без особенностей.

- Версии 7 и 8 – не работает переключение по **Ctrl + Tab**, **Ctrl + Shift + Tab**, если в веб-браузере включены закладки. Переключение по **F6**, **Ctrl + F6** работает без ограничений. В веб-браузере Microsoft Internet Explorer 8 в некоторых случаях при переходе между окнами будет выдаваться предупреждение **Нажмите ОК для перехода к окну** с кнопкой **ОК**. Переход к окну будет выполнен после нажати помощью клавиатуры требуется дополнительная настройка, выполнить которую можно с помощью диалога **Главное меню – Сервис – Настройки веб-браузера**.

- **Google Chrome:**

- Для любых версий – не работает переключение по **Ctrl + Tab**, **Ctrl + Shift + Tab**. Переключение по **F6**, **Ctrl + F6** работает без ограничений.

- Для версии 7.0 и младше – при использовании горячих клавиш (**F6**, **Ctrl + F6**) для переключения окон веб-клиента, при попытке повторного открытия уже открытой формы или при использовании для переключения метода **Активизировать()**, после переключения на новое окно выдается предупреждение **Выполнен переход к окну. Нажмите ОК для продолжения**. После нажатия кнопки **ОК** можно продолжить работу с окном, в которое произошло переключение.

- Модальное окно в веб-браузере Microsoft Internet Explorer блокирует все родительские окна, в веб-браузере Mozilla Firefox блокируется только родительское окно данного окна, в веб-браузере Google Chrome окна не блокируются. При этом остальные окна приложения затемняются, в эти окна можно переключиться, но в них нельзя выполнять никаких действий.

- При открытии модального окна перестает выполняться обновление отображения главного окна в веб-браузере Microsoft Internet Explorer.

- Из-за ограничения веб-браузеров, невозможно открытие немодальной формы из формы, открытой модально. В связи с этим, все формы, открываемые из модальной формы, также открываются в модальном режиме.

- Окно веб-клиента не может быть меньше некоторого размера, определяемого используемым веб-браузером.

- При работе на нескольких мониторах, если они имеют разное установленное разрешение, окна, открытые не на первом мониторе, при повторном открытии могут иметь другие координаты.

- Не поддерживается отображение командной панели с вертикальным отображением кнопок.

- Отображение отсутствующей картинки в веб-клиенте отличается от отображения в тонком клиенте.

- В веб-браузере Microsoft Internet Explorer 8 некорректно отображаются картинки, содержащие цвет с уровнем прозрачности, отличным от 0 или 100%.

- Не поддерживается отображение в поле HTML-документа страниц, имеющих в адресе то же имя хоста веб-сервера, что и веб-клиент. Рекомендуется хранить страницы в информационной базе.

- Не поддерживается обработка событий поля HTML-документа для страниц, имеющих в адресе то же имя хоста веб-сервера, что и веб-клиент. Страницы, для которых требуется обработка событий, следует хранить в информационной базе.

- Окно выбора из списка значений при использовании метода **ВыбратьИзСписка()** открывается в виде отдельного окна веб-браузера, а не в виде списка.

- Меню выбора из списка значений при использовании метода **ВыбратьИзМеню()**, а также меню расшифровки системы компоновки данных открываются в виде отдельного окна веб-браузера, а не в виде меню.

- При вызове метода `ПоместитьФайл()` возможны ситуации, когда ошибка помещения файла корректно не определяется. При этом в качестве возвращаемого значения выступает значение `Истина`, а параметр `Адрес` не содержит адрес во временном хранилище.
- В веб-браузере Safari окна калькулятора и календаря всегда открываются в окнах, размер которых не меньше минимального размера окна для этого браузера
- Цвет полос прокрутки поля HTML-документа соответствует цвету, заданному в операционной системе, кроме случая использования веб-браузера Microsoft Internet Explorer, когда в поле HTML-документа отображается HTML-страница, заданная строкой текста.
- Не поддерживается вызов обработчиков событий `ДокументСформирован` и `ПриНажатии`, а также выполнение команд `Печать` и `Сохранить` поля HTML-документа в том случае, если в поле отображается документ, в адресе которого имя хоста веб-сервера отличается от имени хоста веб-клиента.
- Отличается от других клиентов поведение команд отмены и возврата действия в многострочных полях ввода. В веб-клиенте объем отменяемых изменений определяется веб-браузером.
- В поле ввода результат щелчка левой кнопкой мыши по слову (с нажатой клавишей `Ctrl`) зависит от веб-браузера, а не приводит к выделению слова, по которому выполняется щелчок.
- В поле текстового документа нажатие стрелок влево/вправо в первой/последней позиции строки приводит к переходу на предыдущую/последующую строку.
- В многострочном поле ввода и поле текстового документа не поддерживается ввод текста в режиме замены.
- В многострочном поле ввода и ячейке таблицы формы перенос текста выполняется только по словам. Сплошной текст по границе элемента не переносится.
- При нажатии `Shift + Del` в многострочном поле ввода удаляется вся строка, без учета автоматического разбиения строки по границе поля ввода.
- Если символ, недопустимый с точки зрения спецификации XML версии 1.0 (<http://www.w3.org/TR/xml/>), программно помещается в поле ввода, то в веб-браузерах Google Chrome и Safari этот символ будет заменен символом "?" и поле ввода станет недоступным для редактирования.
- Если в тексте подсказки присутствуют разделители строк, то в веб-браузере Mozilla Firefox вместо них отображаются пробелы.
- Язык интерфейса веб-клиента в веб-браузере Safari соответствует языку интерфейса веб-браузера, если язык интерфейса явно не задан с помощью параметра запуска `L`.
- Если при отладке веб-клиента локальные переменные или параметры методов используются в окне `Выражение`, табло или условиях точек останова, то их имена должны задаваться в точности так, как эти имена заданы при определении (с учетом регистра символов).
- В веб-клиенте не поддерживаются следующие возможности:
 - Общие:
 - печать на текущий принтер.
 - Поле графической схемы:
 - поиск и замена.
 - Табличный документ:
 - переход к ячейке.

- Текстовый документ:
 - переход к строке;
 - работа с закладками;
 - подменю **Текст** главного меню.
- В веб-клиенте не поддерживается вызов следующих команд с помощью горячих клавиш:
 - Поле картинки:
 - увеличить масштаб (**Num+**);
 - уменьшить масштаб (**Num-**);
 - прокрутка картинки с помощью клавиатуры.
 - Поле графической схемы:
 - прокрутка схемы.
 - Таблица, связанная с иерархическим списком:
 - свернуть узел (**Ctrl + Num-**);
 - свернуть все узлы (**Ctrl + Shift + Num-**);
 - развернуть узел (**Ctrl + Num+**);
 - развернуть все узлы (**Ctrl + Shift + Num+**).
 - Табличный документ:
 - свернуть группу (**Ctrl + Num-**);
 - свернуть все группы (**Ctrl + Shift + Num-**);
 - развернуть группу (**Ctrl + Num+**);
 - развернуть все группы (**Ctrl + Shift + Num+**);
 - прокрутка страницы влево (**Alt + PgUp**);
 - прокрутка страницы вправо (**Alt + PgDn**);
 - выделение ячеек с помощью клавиатуры.
 - Текстовый документ:
 - удалить текущую строку (**Ctrl + L**): не поддерживается в веб-браузерах Mozilla Firefox, Google Chrome, Safari.
- В операционной системе MacOS X при включенной неанглийской раскладке клавиатуры не поддерживается использование горячих клавиш, если в сочетаниях используются следующие клавиши:
 - в веб-браузере Google Chrome: клавиши **Alt** и **Alt + Shift**;
 - в веб-браузере Safari: клавиши **Alt**, **Alt + Shift**, **Cmd** и **Cmd + Shift**.

7.4. Особенности работы веб-клиента на iPad

Эти особенности расширяют перечень особенностей поведения веб-клиента, описание которых изложено в предыдущем разделе.

При работе на устройстве iPad:

- Вместо двойного щелчка левой кнопкой мыши используется быстрое двойное нажатие на одном и том же месте экрана. Вместо щелчка правой кнопкой мыши используется длительное нажатие (более 1 сек.).
- Поддерживается только режим работы в закладках и интерфейс [Такси](#).
- Экранная клавиатура может не появляться автоматически при редактировании ячейки таблицы или табличного документа (в появляющемся поле ввода). Для того чтобы она появилась, следует повторно нажать редактируемое поле ввода в ячейке.
- Не поддерживается переход к главному окну приложения с помощью диалога [Все окна](#). При попытке перейти к нему выводится предупреждение с текстом [Не может быть выполнен переход к главному окну](#).
- Рекомендуется для завершения работы с веб-клиентом пользоваться кнопкой закрытия в верхнем правом углу главного окна.
- Не реализовано перетаскивание.
- Не выполняется переход к главному окну при открытых вспомогательных окнах.
- Не поддерживается печать.
- Поле форматированного документа позволяет только просмотр документа (без возможности редактирования).
- Панель состояния отображается таким же окном, как и окно оповещения пользователя.
- Недоступен метод `ПоместитьФайл()`.
- Недоступно расширение работы с файлами, расширение работы с криптографией и внешние компоненты.

Следующие методы объектов и глобального контекста не могут быть использованы в прикладном решении, работающем на iPad:

- Глобальный контекст:
 - `Вопрос()`,
 - `ОткрытьЗначение()`,
 - `ОткрытьФормуМодально()`,
 - `Предупреждение()`,
 - `ВвестиДату()`,
 - `ВвестиЗначение()`,
 - `ВвестиСтроку()`,
 - `ВвестиЧисло()`.

- Объект **ДиалогВыбораШрифта**:
 - **Выбрать ()**.
- Объект **ДиалогВыбораЦвета**:
 - **Выбрать ()**.
- Объект **ДиалогРедактированияСтандартногоПериода**:
 - **Редактировать ()**.
- Объект **КонструкторФорматнойСтроки**:
 - **ОткрытьМодально ()**.
- Объект **ОбработкаРасшифровкиКомпоновкиДанных**:
 - **ВыбратьДействие ()**.
- Объект **ДиалогРасписанияРегламентногоЗадания**:
 - **ОткрытьМодально ()**.
- Объект **СписокЗначений**:
 - **ОтметитьЭлементы ()**,
 - **ВыбратьЭлемент ()**.
- Объект **УправляемаяФорма**:
 - **ВыбратьИзМеню ()**,
 - **ВыбратьИзСписка ()**,
 - **ОткрытьМодально ()**.

7.5. Особенности поведения объектов системы с включенным режимом разделения данных

7.5.1. Изменение значений разделителей

В процессе работы сеанса использование разделителей и значения разделителей могут быть изменены посредством изменения параметров сеанса. Значения параметров сеанса, на которые ссылается общий реквизит (см. [здесь](#)), могут быть изменены в произвольный момент времени в соответствии с ограничениями, наложенными на права пользователя, от лица которого производится попытка изменения параметров сеанса.

Замена текущих значений разделителей позволяет, например, выполнять необходимые административные действия в случаях, когда требуется работа в разделенном режиме.

Однако следует помнить, что если доступ к информационной базе выполняется посредством веб-сервера, то программное изменение значений объектов, связанных с механизмом разделения, может быть заблокировано с помощью файла **default.vrd** (подробнее см. [здесь](#)).

«1С:Предприятие» не гарантирует логическую целостность данных приложения (не информационной базы) после изменения параметров сеанса, аналогично изменению параметров сеанса, влияющих на ограничения доступа (см. [здесь](#)).

Изменение из встроенного языка значения параметра сеанса, на который ссылается хотя бы один разделитель в режиме **Независимо**, приводит к следующим действиям:

- очищается кеш объектов;
- удаляются повторно используемые значения (см. [здесь](#)).

После изменений значений и использования разделителей в сеансе рекомендуется вызвать метод `ОбновитьИнтерфейс()`.

7.5.2. Нумерация объектов

Механизм нумерации работает в рамках каждой уникальной комбинации значений разделителей. Для такой комбинации работает автоматическая нумерация объектов и реализуется контроль уникальности.

При определении номера используются значения разделителей, установленных в сеансе. В том случае, если разделитель не используется в сеансе, для этого разделителя используется значение, установленное у объекта информационной базы.

7.5.3. Оперативная отметка времени

Оперативная отметка времени разделяется в рамках каждой уникальной комбинации независимых разделителей только в том случае, если не существует документов, для которых разрешено проведение в оперативном режиме и которые не входят в состав этих разделителей.

Например, в прикладном решении существует два независимых разделителя: **Абонент** и **Организация**, а также три документа (с разрешенным проведением в оперативном режиме) **ПриходнаяНакладная**, **РасходнаяНакладная** и **Перемещение**. В состав разделителя **Организация** входят документы **ПриходнаяНакладная** и **Перемещение**, а в состав разделителя **Абонент** входят все документы. Разделение оперативной отметки времени будет выполняться только для разделителя **Абонент**, т. к. для разделителя **Организация** существуют документы, которые им не разделены (документ **РаходнаяНакладная**):

	Абонент	Организация
ПриходнаяНакладная	Входит	Входит
РасходнаяНакладная	Входит	Не входит
Перемещение	Входит	Входит
Разделение оперативной отметки	Разделяется	Не разделяется

7.5.4. Предопределенные данные

Предопределенные данные можно использовать для объектов, входящих в состав любых разделителей.

Создание предопределенных данных при первом обращении к таблице объекта будет выполнено только в том случае, если в сеансе используются все разделители (в том числе разделители в режиме **Независимо** и **совместно**), в состав которых входит объект, содержащий предопределенные данные.

Для создания предопределенных данных в новой области данных можно воспользоваться методом `ИнициализироватьПредопределенныеДанные()`. В зависимости от того, как в сеансе используются разделители, в состав которых входит объект конфигурации, можно выделить следующие варианты:

1. В сеансе используются все разделители, в состав которых входит объект конфигурации – предопределенные элементы будут созданы.
2. В сеансе не используются все разделители, в состав которых входит объект конфигурации – предопределенные элементы не будут созданы и объект конфигурации будет пропущен.

Уникальные идентификаторы элементов данных, связанных с предопределенными данными, не совпадают для данных, расположенных в разных областях. Другими словами, если в справочнике **Товары**

есть предопределенный элемент `П1`, и есть две области: с кодами `165` и `567`, то уникальный идентификатор элемента данных, связанный с элементом `П1` в области `165`, не будет равен уникальному идентификатору элемента данных, связанному с элементом `П1` в области `567`.

В распределенной и разделенной информационной базе изменения предопределенных элементов регистрируются следующим образом:

- Если план обмена и объект конфигурации, для которого выполняется регистрация изменения предопределенных данных, входят в состав одного разделителя, то регистрируются изменения только тех предопределенных элементов, у которых значение разделителя совпадает со значением этого же разделителя узла плана обмена. Таким образом, не регистрируются изменения предопределенных элементов, если в планах обмена нет узлов, у которых значение разделителя совпадает со значением разделителя предопределенного элемента. Не выполняется регистрация в узлах, у которых значение разделителя отличается от значения разделителя предопределенного элемента.
- Если план обмена и объект метаданных не имеют общих разделителей, то выполняется регистрация всех предопределенных элементов во всех узлах этого плана обмена.
- Никогда не выполняется регистрация изменений для узла `ЭтотУзел`.

7.5.5. Права доступа

Административные действия, которые выполняются в целом над информационной базой и над конкретной областью данных, регулируются разными правами.

- **Администрирование** – данное право регулирует административные действия над всей информационной базой и редактирование списка пользователей.
- **Администрирование данных** – данное право регулирует административные действия над данными. К этим действиям относятся:
 - просмотр записей журнала регистрации и получение значений отбора без ограничений;
 - установка часового пояса информационной базы (области данных) – вызов метода `УстановитьЧасовойПоясИнформационнойБазы()`;
 - создание начального образа подчиненного узла распределенной информационной базы (вызов метода `СоздатьНачальныйОбраз()`);
 - обновление нумерации объектов (вызов метода `ОбновитьНумерациюОбъектов()`);
 - выполнение методов объекта `СтандартноеХранилищеНастроекМенеджер` в тех случаях, если выполняется работа с настройками пользователя, отличными от текущего.

ПРИМЕЧАНИЕ. Право **Администрирование** требует наличия права **Администрирование данных**. Проверка на наличие в списке пользователей информационной базы пользователя с правом **Администрирование** учитывает только пользователей, в которых не задан ни один разделитель.

Для общих реквизитов, для которых включено разделение данных, права по умолчанию не устанавливаются, даже если в форме редактирования роли установлен флажок **Устанавливать права для реквизитов и табличных частей по умолчанию**.

7.5.6. Пользователи

При создании пользователя информационной базы (объект `ПользовательИнформационнойБазы`) имеется возможность указать значения разделителя для этого пользователя.

Для этого служит свойство `РазделениеДанных` объекта `ПользовательИнформационнойБазы`. В данном свойстве хранится структура, в которой:

- Ключ элемента структуры содержит имя разделителя (как оно задано в конфигураторе).

- Значение элемента структуры содержит строковое представление значения разделителя. Подробнее о формировании строкового представления значения разделителя см. [здесь](#).

Если в структуре присутствует запись с именем какого-либо разделителя, значит, для пользователя значение этого разделителя задано.

Значение свойства **РазделениеДанных** определяет:

- пространство уникальности имен пользователей, если разделитель разделяет аутентификацию (см. [здесь](#));
- разделение списка пользователей, если разделитель разделяет пользователей (см. [здесь](#));
- начальную установку значений разделителей сеанса пользователя для любого разделителя, определенного в конфигурации (см. [здесь](#)).

7.5.7. Планы обмена

Любой разделитель, в состав которого входит план обмена, с точки зрения плана обмена считается находящимся в режиме **Независимо и совместно**. При этом обеспечивается уникальность ссылок элементов плана обмена, в том числе и для тех элементов, которые описывают predetermined узлы в каждой области данных (метод `ЭтотУзел()` менеджера плана обмена).

Поведение системы при регистрации изменений зависит от следующих факторов:

- разделен или нет план обмена;
- разделен или нет регистрируемый объект всеми разделителями плана обмена;
- используется или нет в текущем сеансе хотя бы один разделитель, в состав которого входит план обмена.

Описание того, как будет выполняться автоматическая регистрация изменений в зависимости от вышеуказанных факторов, приведено в таблице:

План обмена	Объект метаданных разделен всеми разделителями плана обмена	Используется хотя бы один разделитель плана обмена	Автоматическая регистрация
Не разделен	–	–	Все узлы
Разделен	Нет	Нет	Все узлы
Разделен	Нет	Да	1
Разделен	Да	Нет	2
Разделен	Да	Да	2

В приведенной таблице:

1. Вызывается исключение при выполнении автоматического заполнения списка узлов получателя. Если план обмена входит в состав хотя бы одного разделителя, в состав которого не входит объект метаданных с включенной авторегистрацией в этом плане обмена, то при обновлении конфигурации базы данных будет выдано предупреждение о том, что объект конфигурации не входит в состав разделителя плана обмена.

2. При автоматическом заполнении списка узлов получателей используется область данных, которая определяется набором разделителей, каждый из которых разделяет и план обмена, и объект метаданных. При этом если используются не все разделители, в состав которых входит и план обмена, и объект, для которого выполняется регистрация изменений, то для недостающих разделителей используются значения разделителей из этого объекта. Если при регистрации изменений обнаруживается, что изменены значения разделителей, то:

- Для узлов плана обмена области данных, определяемых «старой» комбинацией значений разделителей, регистрируется удаление объекта данных.
- Для узлов плана обмена области данных, определяемых «новой» комбинацией значений разделителей, регистрируется изменение объекта данных.

Изменение конфигурации регистрируется для всех узлов всех планов обмена вне зависимости от разделения этих планов обмена каким-либо разделителем (только для планов обмена с установленным флажком [Распределенная информационная база](#)).

При записи узла плана обмена выполняется проверка, что в области данных, описанной значениями всех разделителей, в состав которых входит записываемый объект, должен существовать ровно один элемент плана обмена, для которого значение свойства [ЭтотУзел](#) установлено в значение [Истина](#). Если условие не выполняется – генерируется исключение. Если запись выполняется в режиме загрузки данных (свойство [ОбменДанными.Загрузка](#) установлено в значение [Истина](#)), то проверка не выполняется.

7.5.8. Функциональные опции

В случае если объект конфигурации, использующийся для хранения значения функциональной опции, входит в состав разделителей, то получение значения функциональной опции будет выполняться следующим образом:

- Если не установлен хотя бы один независимый разделитель, который разделяет объект конфигурации, использующийся для хранения значения функциональной опции, то значение функциональной опции, хранимой в реквизите типа [Булево](#), будет равно значению [Истина](#), а для реквизитов других типов будет вызвано исключение.
- Если используются все независимые разделители и не используется хотя бы один разделитель с установленным режимом разделения [Независимо и совместно](#), то значение функциональной опции, хранимой в реквизите типа [Булево](#), будет определяться сложением «по ИЛИ» всех значений с заданными значениями разделителя, а для реквизитов других типов будет вызвано исключение.
- Анализируется использование только тех разделителей, в состав которых входит объект конфигурации, в котором хранится значение функциональной опции.

7.5.9. Регламентные задания

Регламентное задание может входить в состав разделителя.

Запуск регламентного задания выполняется со значениями разделителей, которые заданы для этого задания с помощью свойства [РазделениеДанных](#) объекта [РегламентноеЗадание](#). С помощью этого свойства также можно получить набор разделителей, соответствующих сеансу, в котором запускается регламентное задание (доступны только значения разделителей, находящихся в режиме [Независимо и совместно](#), см. [здесь](#)).

Для объекта [РегламентноеЗадание](#) возможно изменение значений свойства [РазделениеДанных](#) по общим правилам, кроме предопределенных регламентных заданий, для которых изменение свойства [РазделениеДанных](#) невозможно. Регламентное задание может быть изменено из сеанса, в котором используются только разделители, разделяющие это регламентное задание.

При получении регламентных заданий (по уникальному идентификатору или списка регламентных заданий) выполняется неявная установка отбора по разделителям в соответствии со значениями разделителей, установленных в текущем сеансе. В зависимости от режима разделителя будет доступен разный набор регламентных заданий:

- Разделитель в режиме [Независимо](#).
 - Разделитель используется – доступны только регламентные задания, для которых значения разделителей совпадают со значениями разделителей в текущем сеансе.
 - Разделитель не используется – нет доступных регламентных заданий.

- Разделитель в режиме **Независимо и совместно**:
 - Разделитель используется – доступны только регламентные задания, для которых значения разделителей совпадают со значениями разделителей в текущем сеансе.
 - Разделитель не используется – доступны регламентные задания с произвольным значением этого разделителя, а также задания, не разделенные этим разделителем.

Если регламентное задание отмечено в конфигураторе как предопределенное, то при первом входе в режиме 1С:Предприятие с новой комбинацией разделителей происходит создание списка регламентных заданий с текущей комбинацией значений разделителей и их использования. Если при этом в сеансе разделитель не используется, а регламентное задание входит в состав такого разделителя, то в соответствующее свойство регламентного задания будет записано значение по умолчанию для типа разделителя и включено использование.

СОВЕТ. Не рекомендуется создавать разделенные регламентные задания, которые выполняются достаточно часто при большом количестве областей данных. Рекомендуется создавать неразделенное регламентное задание, которое будет само обрабатывать разные значения разделителей.

7.5.10. Фоновые задания

Запуск фоновых заданий выполняется со значениями разделителей, соответствующих сеансу, из которого выполняется запуск фоновых заданий. Во время работы фоновых заданий (или после его завершения) значения разделителей можно получить с помощью свойства **РазделениеДанных** объекта **ФоновоеЗадание** (доступны только значения разделителей, находящихся в режиме **Независимо и совместно**, см. [здесь](#)).

При получении фоновых заданий (по уникальному идентификатору или списка фоновых заданий) выполняется неявная установка отбора по разделителям в соответствии со значениями разделителей, установленных в текущем сеансе. В зависимости от типа разделителя будет доступен разный набор фоновых заданий:

- Разделитель в режиме **Независимо**: доступны только фоновые задания, запущенные из сеансов с таким же использованием и значениями разделителей, что и для текущего сеанса.
- Разделитель в режиме **Независимо и совместно**: доступны только фоновые задания, запущенные из сеансов с такими же значениями разделителей, используемых в текущем сеансе. Неиспользуемые разделители не учитываются.

7.5.11. Web-сервисы

При вызове Web-сервиса значения разделителей могут быть указаны так же, как и для веб-клиента (см. [здесь](#)).

7.5.12. Константы

В случае, если в сеансе используются все разделители, в состав которых входят константы, то в языке запросов можно обращаться к константам посредством таблицы **Константы** (аналогично «1С:Предприятию» версии 8.2.13 и младше).

Поля, соответствующие разделителям, находящимся в режиме **Независимо и совместно**, существуют у объекта **КонстантаМенеджерЗначения** и таблицы **Константа**.

7.5.13. Регистры

Разделитель, находящийся в режиме **Независимо и совместно**, выступает в регистре в качестве измерения, логически предшествующего измерениям, заданным прикладным разработчиком.

В одном наборе записей могут находиться только записи с одинаковым набором значений разделителей.

7.5.13.1. Регистры сведений

Разделители входят в состав основного отбора регистра сведений.

В свойство **Отбор** набора записей регистра сведений могут быть добавлены элементы отбора, соответствующие разделителям, в состав которых входит регистр сведений.

Проверка уникальности записи выполняется в рамках области данных, соответствующей текущей уникальной комбинации значений разделителей.

При вызове метода **Прочитать()** набора записей следует помнить о следующих особенностях:

- Можно не устанавливать значения разделителей в отборе набора записей, если в текущем сеансе используются все разделители, в состав которых входит регистр;
- Если в качестве значений разделителей в отборе набора записей будут установлены значения, не совпадающие со значениями, указанными для текущего сеанса, результатом выполнения метода будет пустой набор записей;
- Если в отборе набора записей не будут установлены значения для разделителей, которые не используются в текущем сеансе, результатом выполнения метода будет набор записей без отбора по значениям этих разделителей.

При вызове метода **Прочитать()** менеджера записи следует помнить о следующих особенностях:

- Можно не устанавливать значения разделителей в менеджере записи, если в текущем сеансе используются все разделители, в состав которых входит регистр;
- Если в качестве значений разделителей в менеджере записи будут установлены значения, не совпадающие со значениями, указанными для текущего сеанса, результатом выполнения метода будет пустая запись;
- Если в менеджере записи не будут установлены значения для разделителей, которые не используются в текущем сеансе, результатом выполнения метода будет пустая запись.

При вызове методов менеджера регистра **Выбрать()**, **Получить()**, **ПолучитьПервое()**, **ПолучитьПоследнее()**, **СрезПервых()** и **СрезПоследних()** нужно учитывать следующие особенности:

- Если в параметре вызова этих методов **Отбор** указано значение для разделителя, используемого в сеансе, то это значение должно совпадать со значением разделителя. В противном случае будет вызвано исключение.
- При вызове метода **Получить()** в параметре **Отбор** должны быть указаны значения для всех разделителей, которые не используются в текущем сеансе.

7.5.13.2. Регистры накопления и бухгалтерии

Параметры регистров (признак разделения итогов, признак использования итогов и т. д.) являются разделяемыми и хранятся для каждой области данных. Получение и установку настройки итогов можно выполнять только для сеанса, в котором используются все значения разделителей, в состав которых входит регистр. При удалении разделителей или модификации их типов, во время обновления конфигурации базы данных, в таблицах параметров регистров происходит объединение записей по новым комбинациям установленных разделителей.

Операцию пересчета итогов (методы **ПересчитатьИтоги()** и **ПересчитатьИтогиЗаПериод()**) и текущих итогов (метод **ПересчитатьТекущиеИтоги()**) можно выполнять при любом состоянии использования разделителей, в состав которых входит соответствующий регистр. Пересчет итогов будет выполняться в пределах области данных, определенной текущими значениями используемых разделителей.

7.5.13.2.1. Регистр бухгалтерии

При редактировании регистров бухгалтерии и планов счетов нужно помнить о следующих особенностях:

- Состав независимых разделителей, в состав которых входит план счетов и связанный с ним регистр

бухгалтерии, должен быть идентичным.

- Регистр бухгалтерии должен входить в состав всех разделителей, находящихся в режиме **Независимо и совместно**, в которые входит план счетов, связанный с данным регистром.

При записи элемента плана счетов, если были изменены настройки учета, влияющие на движения регистров бухгалтерии, проверяется, что в сеансе не используются разделители регистра бухгалтерии в режиме **Независимо и совместно**, которых нет в плане счетов.

При записи набора записей регистра бухгалтерии проверяется, что все счета, использованные в наборе записей, имеют значения разделителей, идентичные значениям разделителей в записях набора записей.

7.5.13.2.2. Агрегаты регистра накопления

Настройки агрегатов являются разделяемыми и хранятся для каждой области данных.

При работе с агрегатами необходимо помнить следующие особенности:

- Следующие методы можно вызывать только в таком сеансе, в котором используются все разделители, включающие регистр:
 - `ПолучитьРежимАгрегатов()`,
 - `ПолучитьИспользованиеАгрегатов()`,
 - `ПолучитьАгрегаты()`,
 - `УстановитьРежимАгрегатов()`,
 - `УстановитьИспользованиеАгрегатов()`.
- Следующие методы можно вызывать в любом случае:
 - `ОпределитьОптимальныеАгрегаты()`. Выполняется над данными области, определенной значениями используемых разделителей.
 - `ПерестроитьИспользованиеАгрегатов()`. Если в сеансе используются все разделители, в состав которых входит регистр, то метод выполняется только для определенной области данных. В противном случае устанавливает одинаковые настройки (состав и периодичность) использования агрегатов для используемых разделителей и всех комбинаций значений неиспользуемых разделителей.
 - `ОбновитьАгрегаты()`, `ОчиститьАгрегаты()`, `АгрегатыЗаполнены()`. Если в сеансе используются все разделители, в состав которых входит регистр, методы выполняются только для определенной области, в противном случае – для каждой области данных в рамках значений используемых разделителей.

Если разделитель находится в режиме **Независимо и совместно** и в сеансе этот разделитель не используется, то существует возможность использования агрегатов, кроме следующих случаев:

- Если для некоторых значений разделителя включен режим итогов.
- Если для некоторых значений разделителя выключено использование агрегатов.
- Если для разных значений разделителя выбрана разная минимальная периодичность используемых агрегатов. Если метод `ПерестроитьИспользованиеАгрегатов()` вызывался в неразделенном режиме, то минимальная периодичность используемых агрегатов будет одинаковой.
- Если для разных значений разделителей наборы используемых агрегатов для разных значений разделителей не пересекаются. Если метод `ПерестроитьИспользованиеАгрегатов()` вызывался в случае, когда разделители не используются, то наборы используемых агрегатов будут одинаковыми.

- Для запросов в транзакции – если для некоторых значений разделителей процесс обновления агрегатов не завершен и находится на этапе переноса данных в агрегат из буфера.

7.5.13.3. Регистры расчета

При получении виртуальной таблицы данных графика сопоставление записей регистра расчета и записей регистра сведений графика работы выполняется не только по реквизитам регистра расчета с заполненным свойством [Связь с графиком](#), но и по разделителям в режиме [Независимо и совместно](#), являющимся общими для регистра сведений графика и регистра расчета.

При получении виртуальной таблицы базовых данных сопоставление записей регистров расчета выполняется не только по полям, переданным как параметры виртуальной таблицы, но и по разделителям в режиме [Независимо и совместно](#), которые являются общими для регистра сведений графика и регистра расчета.

При расчете фактического периода действия разделение периода действия между вытесняющими видами расчетов выполняется с учетом совпадения значений разделителей (в режиме [Независимо и совместно](#)).

Также нужно помнить о следующих особенностях:

- при записи набора записей регистра расчета не контролируется совпадение значений разделителей (в режиме [Независимо и совместно](#)) в записи регистра расчета и в виде расчета этой записи;
- при записи вида расчета не контролируется совпадение значений разделителей (в режиме [Независимо и совместно](#)) в виде расчета и во всех записях регистра расчетов с таким видом расчета;
- не контролируется совпадение значений разделителей (в режиме [Независимо и совместно](#)) в виде расчета и в тех видах расчета, которые включены в табличные части вытесняющих, базовых и ведущих видов расчета.

7.5.13.3.1. Перерасчеты

Перерасчеты автоматически входят в состав того же набора разделителей, что и соответствующие регистры расчета. При записи перерасчетов используются значения разделителей, установленных для соответствующих записей регистра расчета.

7.5.14. Последовательность документов

Разделитель, находящийся в режиме [Независимо и совместно](#), выступает для последовательности в качестве измерения.

Последовательность документов является разделенной только в следующем случае:

- разделены все документы и регистры, входящие в нее;
- документы и регистры, входящие в последовательность, разделены одинаковым набором разделителей.

Параметр [Отбор методов УстановитьГраницу\(\)](#) и [Проверить\(\)](#) является необязательным.

Если свойство [Заполнение последовательностей](#) документа, участвующего в последовательности, установлено в значение [Заполнять автоматически](#), то значения общих реквизитов записи последовательности будут установлены в значения, соответствующие документу, который порождает эту запись.

7.5.15. Параметры информационной базы

Часовой пояс информационной базы разделяется независимыми разделителями. Если используются не все разделители, то при попытке получения или установки часового пояса информационной базы вызывается исключение. В противном случае используется значение часового пояса в разрезе значений разделителей.

Об изменении часового пояса информационной базы другие сеансы оповещаются не позднее чем через 20 секунд после установки. Не требуется монопольный доступ к информационной базе, если установка часового пояса выполняется в сеансе, в котором используется хотя бы один разделитель.

Метод `УстановитьЧасовойПоясИнформационнойБазы()` требует установки монопольной блокировки области в том случае, если в текущем сеансе не используется ни один разделитель.

7.5.16. Список сеансов

В список сеансов попадут сеансы, в которых значения разделителей совпадают со значениями в текущем сеансе (из которого выполняется получение списка с помощью метода `ПолучитьСеансыИнформационнойБазы()`). В этом списке сеансов имена пользователей (и имена компьютеров) будут недоступны, если пользователь не принадлежит области данных, доступной текущему сеансу. Например, в информационной базе существует разделитель `Абонент` (типа `Число`). Для пользователя `Иванов` значение разделителя установлено равным `5`. Если в процессе работы прикладное решение изменит значение разделителя `Абонент` на любое другое значение, то в списке сеансов имя пользователя будет недоступным.

7.5.17. Список соединений

Список соединений (полученный с помощью метода `ПолучитьСоединенияИнформационнойБазы()`) будет непустым только в том случае, если в текущем сеансе не используется ни один разделитель.

7.5.18. Журнал регистрации

7.5.18.1. Визуальное отображение

Если журнал регистрации открыт в сеансе, где существует хотя бы один неиспользуемый разделитель, то в следующие формы будет добавлено поле `Разделение данных сеанса`:

- форма списка журнала регистрации,
- форма настройки отбора журнала регистрации,
- форма записи журнала регистрации.

Представление имен разделителей определяются следующим образом:

- Для существующего разделителя – его представление.
- Для переименованного разделителя, т. е. разделителя, имя которого было изменено после внесения записей с его участием в журнал, – текущее представление разделителя.
- Для удаленного разделителя – последнее имя удаленного разделителя, дополненное пробелами перед не первыми прописными буквами. При этом прописные буквы преобразуются в строчные (кроме первой буквы в имени разделителя).

7.5.18.2. Отбор

При формировании (для просмотра или при выгрузке с помощью метода `ВыгрузитьЖурналРегистрации()`) журнала регистрации всегда выполняется фильтрация по тем разделителям, которые используются в текущем сеансе. Такой отбор (принудительный) невозможно отключить или изменить. Если формирование журнала регистрации выполняется в сеансе, в котором разделитель условно выключен, то отбор по этому разделителю не выполняется.

Для тех разделителей, которые не используются в текущем сеансе, возможна пользовательская установка значений отбора. При этом принудительный отбор и пользовательский отбор по значениям разделителей объединяются «по И».

При получении значений отбора для полей `Пользователь` и `Компьютер` (с помощью метода `ПолучитьЗначенияОтбораЖурналаРегистрации()`) результат формируется следующим образом:

- Разделитель используется – возвращаются значения полей из тех записей журнала, которые были зафиксированы для текущего значения разделителя.
- Разделитель не используется – возвращаются все значения полей, зафиксированные журналом регистрации.

При программном получении записей журнала регистрации (с помощью метода `ВыгрузитьЖурналРегистрации()`) отбор формируется путем добавления в структуру фильтра элемента `РазделениеДанныхСеанса`, имеющего тип `Структура`. Каждый элемент структуры в качестве ключа содержит имя разделителя, а в качестве значения – набор значений разделителя, по которым следует отбирать записи журнала регистрации. Если в структуре `РазделениеДанныхСеанса` задано больше одного элемента, запись будет удовлетворять отбору, если она удовлетворяет всем значениям, указанным в структуре (отбор «по И»).

Этот набор можно задавать либо с помощью структуры (в этом случае отбор выполняется по одному значению), либо с помощью массива структур (в этом случае отбор задается набором значений).

Структура, используемая для отбора, содержит следующие элементы:

- `Использование` (типа `Булево`). Значение по умолчанию – `Истина`. Задаёт использование общего реквизита, разделяющего данные. Если `Ложь`, элемент `Значение` игнорируется и может быть опущен. Таким образом, можно отбирать записи журнала, в которых данный общий реквизит не установлен.
- `Значение` (произвольного типа). Значение общего реквизита, по умолчанию – `Неопределено`.

Если в элементе структуры `РазделениеДанныхСеанса` указан массив структур, то запись журнала регистрации будет удовлетворять этому условию, если она удовлетворяет хотя бы одному значению (отбор «по ИЛИ»).

Рассмотрим пример.

В прикладном решении заданы два разделителя:

- `Абонент`, тип `Число`;
- `Филиал`, тип `Число`.

Пример 1:

- Тип `Структура`.
 - Ключ `Абонент`, значение `Структура`:
 - Элемент 1: ключ `Использование`, значение `Истина`.
 - Элемент 2: ключ `Значение`, значение `1`.
 - Ключ `Филиал`, значение `Структура`:
 - Элемент 1: ключ `Использование`, значение `Ложь`.

В этом случае в отбор попадут записи журнала регистрации, которые одновременно удовлетворяют следующим условиям:

- Разделитель `Абонент` используется и установлен в значение `1`.
- Разделитель `Филиал` не используется.

Пример 2:

- Тип `Структура`.

- Ключ **Абонент**, значение **Массив**:
 - Элемент 1: **Структура**:
 - Элемент 1: ключ **Использование**, значение **Истина**.
 - Элемент 2: ключ **Значение**, значение **1**.
 - Элемент 2: **Структура**:
 - Элемент 1: ключ **Использование**, значение **Истина**.
 - Элемент 2: ключ **Значение**, значение **5**.
- Ключ **Филиал**, значение **Структура**:
 - Элемент 1: ключ **Использование**, значение **Истина**.
 - Элемент 2: ключ **Значение**, значение **8215**.

В этом случае в отбор попадут записи журнала регистрации, которые одновременно удовлетворяют следующим условиям:

- Разделитель **Абонент** используется и может принимать значение **1** или **5**.
- Разделитель **Филиал** используется и может принимать значение **8215**.

7.5.18.3. Получение записей журнала регистрации

При выгрузке журнала регистрации (в таблицу значений или в документ формата XML) в соответствующую структуру данных добавляются следующие колонки:

- **РазделениеДанныхСеанса**, тип **Структура**, где ключ структуры – имя разделителя, значение – значение разделителя;
- **ПредставлениеРазделенияДанныхСеанса**, тип **Массив**, состоящий из строк вида **<Представление разделителя>**: **<Представление значения разделителя>**.

Эти колонки добавляются только в том случае, если в текущем сеансе есть хотя бы один разделитель, который не используется в данном сеансе. Данные, содержащиеся в колонках **РазделениеДанныхСеанса** и **ПредставлениеРазделенияДанныхСеанса**, содержат информацию только по неиспользуемым разделителям.

Например, в прикладном решении существуют два разделителя:

- **Абонент**,
- **Филиал**.

Если в сеансе используются оба разделителя, то в выгружаемых данных не будет колонок **РазделениеДанныхСеанса** и **ПредставлениеРазделенияДанныхСеанса**. Если в сеансе не используется только разделитель **Абонент**, то в данных колонок **РазделениеДанныхСеанса** и **ПредставлениеРазделенияДанныхСеанса** будет информация только по разделителю **Абонент**, а по разделителю **Филиал** информация будет отсутствовать.

Представление имен разделителей определяется следующим образом:

- Для существующего разделителя – его представление.
- Для переименованного разделителя, т. е. разделителя, имя которого было изменено после внесения записей с его участием в журнал, – текущее представление разделителя.

- Для удаленного разделителя – последнее имя удаленного разделителя, дополненное пробелами перед не первыми прописными буквами. При этом прописные буквы преобразуются в строчные (кроме первой буквы в имени разделителя).

Для получения набора допустимых значений общих реквизитов, разделяющих данные, используется метод `ПолучитьЗначенияОтбораЖурналаРегистрации()`.

Для получения набора общих реквизитов, разделяющих данные, в первом параметре метода следует указать строку `РазделениеДанныхСеанса`.

В структуре, которую возвращает метод, набор общих реквизитов записывается в элемент с ключом `РазделениеДанныхСеанса`. Значением элемента является соответствие, где:

- **Ключ** – имя общего реквизита,
- **Значение** – представление общего реквизита.

Для получения набора значений разделителей в первом параметре метода следует указать строку `РазделениеДанныхСеанса.<Имя общего реквизита>`, где `<Имя общего реквизита>` – имя общего реквизита (см. выше).

В структуре, которую возвращает метод, набор допустимых значений общего реквизита записывается в элемент с ключом `РазделениеДанныхСеансаЗначения`. Значением элемента является соответствие, где:

- **Ключ** – имя общего реквизита.
- **Значение** – значение типа `Соответствие`, в котором:
 - **Ключ** – значение общего реквизита;
 - **Значение** – представление значения общего реквизита.

7.5.18.4. Прочее

Если пользователь недоступен в текущем сеансе (принадлежит другой области данных, удален и т. д.), то при просмотре журнала, его выгрузке и в диалоге установки отбора такой пользователь отображается в виде строки `Имя пользователя <не найден>`.

7.5.19. Настройки и избранное

Настройки хранятся в разрезе уникальной комбинации значений разделителей и имен пользователей. Разные пользователи с одинаковым именем неразличимы. Отключенный разделитель представляет собой особое значение, для которого также выполняется хранение настроек.

При изменении значений разделителей в процессе работы системы не происходит автоматического считывания настроек для нового состава разделителей. Для этого необходимо закрыть и заново открыть ту форму, настройки которой требуется перечитать.

При реструктуризации информационной базы (после изменения состава разделителей) выполняются следующие действия:

- При добавлении разделителя считается, что все настройки, которые были ранее в базе данных, были записаны в ситуации, когда новый разделитель используется и его значением является значение по умолчанию для типа добавленного общего реквизита.
- Если при удалении разделителя обнаружено несколько настроек, сохраненных для одинаковых объектов конфигурации и различных значений разделителя, то будут сохранены только те настройки, которые сохранялись для минимального значения удаляемого разделителя. При возникновении такой ситуации при реструктуризации будет выдано предупреждение.

Работа с избранным происходит по вышеописанным правилам.

7.5.20. История

История действий пользователя хранится в разрезе уникальной комбинации значений разделителей и имен пользователей. Разные пользователи с одинаковым именем неразличимы. Отключенный разделитель представляет собой особое значение, для которого также выполняется хранение истории.

При реструктуризации информационной базы (после изменения состава разделителей) выполняются следующие действия:

- При добавлении разделителя считается, что вся история действия, которая была ранее в базе данных, была записана в ситуации, когда новый разделитель используется и его значением является значение по умолчанию для типа добавленного общего реквизита.
- При удалении разделителя выполняется очистка истории таким образом, чтобы в каждой уникальной комбинации значений остающихся разделителей было не более 200 записей истории.

7.5.21. Настройки стандартного интерфейса OData

Настройки стандартного интерфейса OData хранятся в разрезе уникальной комбинации значений разделителей. Отключенный разделитель представляет собой особое значение, для которого также выполняется хранение настройки стандартного интерфейса OData.

При реструктуризации информационной базы (после изменения состава разделителей) выполняется удаление настройки стандартного интерфейса OData и ее необходимо выполнить заново (см. [здесь](#)).

7.5.22. Запросы

Если объект входит в состав разделителя и в запросе к данным используется соединение с его объектной таблицей, то в условии соединения (секция запроса **ПО**) не допускается использование табличной части объекта.

Следует учитывать, что эффективность запросов, которые выполняются с низкой эффективностью (из-за большого количества соединений) в системе без разделителей, будет еще более снижена, если в системе существует более двух разделителей или разделитель типа **Строка**. В связи с этим следует избегать использование большого количества соединений в системах с указанными особенностями.

7.5.23. Управляемые транзакционные блокировки

При блокировке из языка без указания каких-либо полей блокируется все пространство по используемым разделителям.

Установка блокировки из языка с явным указанием разделителей доступна только для разделителей в режиме **Независимо и совместно**, только для пространств измерений (не наборов записей) регистров.

Если объект записывается со значениями разделителей, равными значениям по умолчанию своих типов и эти разделители используются в текущем сеансе, то блокировка устанавливается по значениям разделителей, указанным в сеансе.

Если блокировка выполняется по значению ссылки, то пространство блокировки определяется этим значением и используемыми значениями независимых разделителей. Таким образом, одновременная блокировка по значению ссылки из сеанса, в котором используется разделитель в режиме **Независимо и совместно**, и другого сеанса, в котором этот же разделитель не используется, невозможна.

В том случае, когда сеанс устанавливает монопольный доступ к каким-либо областям, все попытки установить управляемые блокировки на эти области (выполняемые из этого же сеанса) будут игнорироваться.

7.5.24. Монопольный режим

Метод `УстановитьМонопольныйРежим()` оперирует с областью, описанной разделителями, которые используются в данном сеансе. Таким образом, в одной информационной базе может одновременно существовать несколько областей, для которых установлен монопольный режим (с учетом ограничений,

описанных далее). Если в сеансе не используется ни один разделитель, монопольный режим будет установлен не на область, а на всю базу данных.

Монопольный режим области не совместим с любой блокировкой другим сеансом той же области данных или ее любой вложенной области, а также с монопольным режимом, который пытается установить другой сеанс для любой объемлющей области. При этом область 02 является вложенной в область 01 в том случае, если существует разделитель, который используется в области 02 и не используется в области 01, а разделители, одновременно используемые в областях 01 и 02, имеют одинаковые значения. Соответственно, аналогичным образом определяется объемлющая область.

При переходе из области в область, список областей, на которые данный сеанс установил монопольный доступ, не изменяется.

Монопольный режим, заданный для области данных, запрещает:

- Создание новых сеансов для области и ее вложенных областей;
- Изменение данных области из объемлющих областей;
- Установку монопольного режима в объемлющих областях.

При этом (для вышеописанных ограничений) можно считать всю информационную базу областью, являющейся объемлющей для любой области данных.

7.5.25. Удаление помеченных объектов

Метод `НайтиПомеченныеНаУдаление()` находит все помеченные на удаление объекты, доступные в текущем сеансе.

При попытке удаления объектов (метод `УдалитьОбъекты()`) с контролем ссылочной целостности этот контроль выполняется среди тех объектов, которые доступны в текущем сеансе в соответствии с установленными разделителями. Возможны ситуации, когда на удаляемые объекты существуют ссылки из других объектов, недоступных в рамках текущего сеанса (например, из другой области данных). В этом случае ответственность за удаление объектов лежит на разработчике прикладного решения и пользователе, выполняющем такое удаление.

СОВЕТ. В прикладных решениях рекомендуется ограничивать удаление объектов (например, удаление неразделенных объектов в разделенных сеансах) с помощью средств конфигурирования (права доступа и т. д.).

7.5.26. Полнотекстовый поиск

В прикладном решении, использующем разделители, индекс полнотекстового поиска (как совокупность служебных файлов) является общим для всех значений разделителей информационной базы. Обновление индекса происходит сразу для всех значений разделителей и может выполняться только в сеансе, где все разделители не используются. Обновление полнотекстового индекса не поддерживается, если в сеансе используется хотя бы один разделитель.

Для поиска будут использоваться все неразделенные объекты и разделенные объекты, для которых выполняются следующие условия:

- Разделитель в режиме **Независимо**:
 - Разделитель используется – в поиске участвуют объекты, для которых значение разделителя, установленное для сеанса, совпадает со значением, которое установлено для объекта.
 - Разделитель не используется – разделенные объекты не участвуют в поиске.
- Разделитель в режиме **Независимо и совместно**:
 - Разделитель используется – в поиске участвуют объекты, для которых значение разделителя, установленное для сеанса, совпадает со значением, которое установлено для объекта.

- Разделитель не используется – в поиске участвуют все разделенные объекты.

Если в сеансе используются все разделители прикладного решения, то время поиска информации эквивалентно времени поиска в информационной базе без разделителей, аналогичной по характеристикам области данных. Если в сеансе используются не все разделители прикладного решения, то поиск может занимать несколько большее время, чем в информационной базе без деления данных, аналогичной по параметрам области данных по заданным разделителям.

СОВЕТ. Регламентные задания, используемые для обновления и слияния индекса полнотекстового поиска, необходимо делать неразделенными.

В силу того, что индекс полнотекстового поиска один для всей информационной базы (а не для каждой комбинации разделителей), метод `ИндексАктуален()` может вернуть `Ложь` даже в том случае, если в текущем сеансе не было никаких изменений, которые могут попасть в индекс. Такие изменения могут быть выполнены в сеансах с другими значениями разделителей.

Методы `ОбновитьИндекс()`, `ОчиститьИндекс()`, `УстановитьРежимПолнотекстовогоПоиска()` могут использоваться только в том случае, когда в сеансе не используется ни один разделитель. В противном случае вызывается исключение.

7.6. Особенности работы клиентского приложения под ОС Linux

Клиентское приложение, работающее под ОС Linux, обладает некоторыми особенностями и ограничениями:

- Не поддерживается технология COM и весь перечень возможностей, связанных с этой технологией:
 - Не поддерживается работа с объектом `COMОбъект`;
 - Запуск клиента «1С:Предприятия» в режиме Automation-сервера;
 - Внешние компоненты, созданные с использованием COM-технологии.
- Не поддерживается консоль администрирования кластера серверов «1С:Предприятия». Администрирование возможно с помощью сервера администрирования (`ras`) и утилиты администрирования (`rac`).
- Не поддерживается аутентификация пользователя средствами операционной системы.
- Не поддерживается работа с объектом `Почта`. Вместо этого объекта можно использовать почтовое приложение по умолчанию, используя функцию `ЗапуститьПриложение()`, передавая ей в качестве параметра URL, сформированный в соответствии с описанием схемы `mailto:` (RFC 2368, <http://tools.ietf.org/html/rfc2368>).
- Не поддерживается размещение на одном компьютере нескольких версий приложения и возможность автоматического «подбора» версии системы для работы с информационной базой.
- Не поддерживается использование метафайлов Windows (`WMF`, `EMF`).
- При разработке прикладных решений следует учитывать, что элемент управления `ПолеHTMLДокумента` (в обычном режиме) и поле формы вида `Поле HTML документа` (в управляемом режиме) реализовано на основе библиотеки `WebKit` вместо `Internet Explorer`. В связи с этим следует учитывать, что внешний вид и объектная модель HTML-документа (DOM-модель), доступ к которому осуществляется с помощью свойства `ПолеHTMLДокумента.Документ`, могут отличаться от таковых при работе в клиентском приложении, работающего на ОС Windows.

7.7. Особенности работы мобильного приложения

При разработке мобильного приложения поддерживаются следующие объекты конфигурации и механизмы системы:

- Подсистемы;
- Константы;
- Справочники;
- Документы;
- Журналы документов;
- Регистры накопления (кроме разделения итогов и режима агрегатов);
- Регистры сведений;
- Обработки;
- Перечисления;
- Права доступа (с ограничениями), роли;
- Пользователи информационной базы (с ограничениями);
- Функциональные опции;
- Параметры сеанса;
- Планы обмена (за исключением планов обмена с установленным признаком [Распределенная информационная база](#));
- Подписки на события;
- Запросы;
- Динамический список (с ограничениями);
- Использование Web-сервисов (без возможности создания Web-сервиса в мобильном прикладном решении);
- Общие картинки;
- Общие команды и группы команд;
- Общие макеты;
- Общие модули;
- Общие формы;
- Объекты настройки системы компоновки данных (без интерфейса пользователя);
- Условное оформление;
- Механизм XDTO (без возможности создания пакетов XDTO);
- Полнотекстовый поиск;
- Языки;
- Фоновые задания.

Не поддерживаются механизмы и объекты конфигурации:

- Переключение интерфейсов (и режим совместимости интерфейсов)
- Механизмы бухгалтерского учета;
- Механизмы периодических расчетов;
- Механизмы бизнес-процессов и задач;
- Система компоновки данных;
- Механизмы отладки прикладных решений;
- Механизм общих реквизитов;
- Ограничения доступа к данным;
- Управляемые блокировки;
- Работа с внешними источниками данных;
- Механизм автоматизированного тестирования;
- Механизм перетаскивания;
- Стандартные функции;
- Некоторые элементы управляемых форм;
- Расширенное редактирование в элементах формы;
- Пользовательская настройка форм;
- Сохранение и восстановление данных форм в настройках;
- Информационная панель, история работы пользователя и оповещения (метод `ПоказатьОповещениеПользователя()`);
- Отображение состояния длительных процессов (метод `Состояние()`);
- Справочная система;
- Механизм регламентных заданий;
- Возможности и изменения, реализованные в платформе для персонального компьютера, автоматически не переносятся в мобильную платформу. Новые возможности мобильной платформы отражаются соответствующими записями в файле [V8Update.htm](#), в разделе [Мобильная платформа](#).

Некоторые объекты конфигурации и механизмы работают с особенностями:

- В ОС iOS не поддерживается работа оператора `Выполнить()` и функции `Вычислить()`;
- ОС iOS, до версии 5.x (включительно), поддерживает ввод паролей только с помощью букв латинского алфавита и цифр. В версии ОС iOS 6 допускается использовать в паролях символы национальных алфавитов. Для обеспечения совместимости между различными версиями ОС рекомендуется задавать пароли пользователей с использованием цифр и букв только латинского алфавита.
- При работе динамического списка не поддерживается интерактивное управление настройками динамического списка. Управление настройками доступно только из встроенного языка.
- На мобильных устройствах под управлением ОС Android, анимация в мобильном приложении может быть отключена, если платформа оценивает данное устройство как недостаточно мощное.

Также необходимо помнить, что для поддерживаемых объектов реализованы не все возможности, доступные в прикладных решениях для персональных компьютеров. В связи с этим следует обращать внимание на раздел [Доступность](#) синтакс-помощника для определения доступности того или иного метода или свойства в мобильной платформе.

Приложение 8. Особенности работы с различными СУБД

8.1. Общие особенности

- В индекс может входить не более 16 полей базы данных, кроме файлового варианта, где ограничением является 256 полей.
- Операции сравнения строк в языке запросов не учитывают концевые пробелы, в отличие от сравнения строк на встроенном языке, где концевые пробелы участвуют в операции сравнения. Например, сравнение строк «bb» и «bb » на встроенном языке вернет **Ложь** (строки не равны) и **Истина** в языке запросов (строки равны).

8.2. Файловая база данных

- Файл базы данных (.1CD) внутри организован как множество так называемых внутренних файлов. Каждой из таблиц базы данных соответствует до четырех внутренних файлов:
 - Файл описания таблицы – в нем находится описание таблицы.
 - Файл записей данных – содержит данные всех записей таблицы, за исключением данных, содержащихся в полях неограниченной длины.
 - Файл индексов – размещены все индексы, определенные для таблицы. Если не определено ни одного индекса, то этот файл отсутствует.
 - Файл значений неограниченной длины – хранятся значения неограниченной длины, содержащиеся в полях таблицы.
- Размер каждого из вышеперечисленных внутренних файлов не может превышать 4 Гбайт.
- Длина ключа в индексе не может превышать 1920 байт.

8.3. Сервер IBM DB2

- Правым операндом операции сравнения **ПОДОБНО (LIKE)** может быть только литерал (параметр) или выражение над литералами. Шаблонными символами являются только «_» – любой символ и «%» – последовательность любых символов.
- Нетипизированный **NULL**. Тип колонки, являющейся результатом запроса **ВЫБРАТЬ NULL**, – самый общий составной тип, а не простой тип, совместимый с любым другим. Поэтому такая колонка не может использоваться в операциях, в которых не могут использоваться поля составных типов. Например, **ВЫБРАТЬ ЕСТЬNULL(f1 + 1, 1) FROM (ВЫБРАТЬ NULL f1) t1** при работе с DB2 приведет к ошибке, поскольку операндом операции **<+>** не может быть поле составного типа.
- Максимальная длина числовых данных – 31 знак (а не 38).
- Максимальная длина ресурсов регистров накопления и бухгалтерии – 25 знаков (а не 32).
- Максимальный размер данных неограниченной длины равен 1 Гбайт.
- Максимальное количество колонок в списке выборки оператора не может превышать 1012. При определении количества колонок нужно учитывать, что для полей составного типа «1С:Предприятие» создает несколько колонок в таблице СУБД.
- Отличные от других СУБД правила определения точности результата при выполнении арифметических операций.
- Использование подзапроса на языке запросов в разделе **ПО** может приводить к замедлению исполнения запросов. При многократном применении подзапроса в разделе **ПО** (например,

соединение нескольких таблиц по условиям, содержащим подзапрос) возможна ситуация, что запрос не будет исполнен.

- Если запрос содержит операцию конкатенации 10 или более строковых значений, включающих поля и литералы и результат конкатенации участвует в сравнении, то возможно возникновение ошибки [Ошибка СУБД: SQL0401N. Операнды, заданные для операции '...', имеют несовместимые типы данных.](#)

- Не допускается соединение в запросе, если выполняется соединение двух таблиц, с условием, включающим сравнение полей табличной части.

- Если в настройках СУБД параметр `COLLATION_SEQUENCE` не равен значению `UCA500R1_LROOT_AN_CX_EX_FX_HX_NX_S2`, то при работе прикладного решения будут присутствовать следующие особенности:

- Не поддерживается работа фонового обновления конфигурации базы данных;
- Чувствительность к регистру букв при сравнении строк. В неявных сравнениях строк, выполняемых в процессе отработки `РАЗЛИЧНЫЕ (DISTINCT)`, `СГРУППИРОВАТЬ ПО (GROUP BY)`, `ОБЪЕДИНИТЬ (UNION)` языка запросов, большие и маленькие буквы различаются. Поэтому исполнение запросов имеет следующие особенности:
 - При использовании конструкции `РАЗЛИЧНЫЕ` и `ОБЪЕДИНИТЬ` (без `ВСЕ`) значения полей результата запроса типа `Строка` считаются различными, если они отличаются только регистром букв (при использовании других СУБД они считаются одинаковыми).
 - При использовании конструкции `УПОРЯДОЧИТЬ ПО` значения типа `Строка`, отличающиеся только регистром букв, считаются различными:
 - если запрос содержит конструкцию `ОБЪЕДИНИТЬ`, `ОБЪЕДИНИТЬ ВСЕ`, `РАЗЛИЧНЫЕ`;
 - список выборки запроса содержит выражение `ВЫБОР`, в котором имеются вложенные запросы;
 - текущий пользователь имеет ограничения доступа к данным, а запрос не содержит ключевого слова `РАЗРЕШЕННЫЕ`.
 - В других случаях сравнение строковых данных нечувствительно к регистрам букв (значения типа `Строка`, отличающиеся только регистром букв, считаются одинаковыми):
 - при явном сравнении полей типа `Строка`;
 - при использовании конструкции `СГРУППИРОВАТЬ ПО`;
 - в конструкции `УПОРЯДОЧИТЬ ПО`, за исключением перечисленных случаев.

Параметр СУБД `COLLATION_SEQUENCE` устанавливается при создании базы данных. Если база данных создавалась «1С:Предприятием» версии 8.1.12 и младше, то этот параметр **не** равен значению `UCA500R1_LROOT_AN_CX_EX_FX_HX_NX_S2` и будут наблюдаться вышеописанные особенности. База данных, созданная «1С:Предприятием» версии 8.1.13 и старше, устанавливается этот параметр в значение `UCA500R1_LROOT_AN_CX_EX_FX_HX_NX_S2` и описанных ограничений наблюдаться не будет.

- В случае обнаружения одной из следующих ситуаций:
 - Большое время выполнения запроса;
 - Большое количество операций с диском во время выполнения запроса;
 - Наличие в плане запроса конструкции `HASH JOIN`;
 - Запрос начинает выполняться существенно быстрее при изменении уровня оптимизации (`current query optimization`) с 5 до 0,

Рекомендуется выполнить команду `db2set DB2_OVERRIDE_BPF=5000` и выполнить перезапуск СУБД IBM DB2. Данную рекомендацию не нужно выполнять на версиях СУБД IBM DB2 9.7 FixPack 5 и выше.

8.4. Сервер Microsoft SQL Server

- В запросе можно использовать не более 256 таблиц (для версий Microsoft SQL Server 2000 и Microsoft SQL Server 2005).
- Одна запись в таблице не может занимать более 8060 байт;
- Одна запись индекса не может занимать более 900 байт. Если запись индекса занимает более 900 байт, то ошибка будет происходить не в момент обновления конфигурации базы данных, а во время попытки вставки значения, в результате которого в индекс попадет запись, превышающая ограничение.
- При использовании Microsoft SQL Server 2000, при группировке по выражению (а не полю), при использовании в разделе **ИМЕЮЩИЕ** агрегатной функции, не входящей в раздел **ВЫБРАТЬ**, может происходить ошибка СУБД. При создании решений, ориентированных на Microsoft SQL Server 2000 рекомендуется включать выражения, использующиеся в разделе **ИМЕЮЩИЕ** в раздел **ВЫБРАТЬ**. На СУБД Microsoft SQL Server 2005 и выше эта проблема не проявляется.
- В случаях, когда прикладное решение рассчитано на работу на Microsoft SQL Server 2000, не рекомендуется использовать функции работы с датами в разделе запроса **СТРУППИРОВАТЬ ПО**, т. к. это может приводить к неверным результатам. Вместо использования выражений, содержащих функции, можно использовать группировку по всем полям таблиц, от которых зависело выражение.

8.5. Сервер Oracle Database

- Одна запись индекса не может превышать приблизительно 80% от размера блока:
 - Для блока размером 4K – 3318 байт;
 - Для блока размером 8K – 6398 байт;
 - Для блока размером 16K – 12958 байт.
- В автоматическом режиме Oracle Database использует табличные блокировки. Это означает, что транзакция, захватившая одну запись в таблице, блокирует эту таблицу целиком, что может помешать конкурирующим транзакциям работать с данными этой таблицы. Рекомендуется разработку прикладных решений вести в режиме управляемых блокировок с целью максимального использования возможностей СУБД. Использование прикладного решения в автоматическом режиме рекомендуется для совместимости с предыдущими версиями прикладных решений и не рекомендуется для реальной эксплуатации.
- Отсутствует возможность использования внутри оператора **В** вложенного запроса с модификатором **ПЕРВЫЕ** и разделом **УПОРЯДОЧИТЬ ПО**, если внутри вложенного запроса есть обращения к полям внешнего запроса.
- При сортировке по возрастанию поля со значениями **NULL** оказываются последними в выборке.
- Использование подзапроса на языке запросов в разделе **ПО** может приводить к замедлению исполнения запросов. При многократном применении подзапроса в разделе **ПО** (например, соединение нескольких таблиц по условиям, содержащим подзапрос) возможна ситуация, при которой запрос не будет исполнен.
- Оптимизируется запись первого (по порядку в таблице) реквизита типа **ХранилищеЗначений**, а если таких нет, то с первого (по порядку в таблице) реквизита с типом **Строка неограниченной длины**.
- Следует учитывать, что при работе с Oracle Database актуальность статистики оказывает влияние на производительность. Рекомендуется поддерживать статистику СУБД в актуальном состоянии (по умолчанию это делается самой СУБД, однако можно самостоятельно запускать сбор статистики с помощью процедуры `dbms_stats.gather_schema_stats`). В том числе имеет смысл выполнять сбор

статистики после загрузки информационной базы из файла выгрузки информационной базы (файлы *.dt). После перезагрузки сервера баз данных сбор статистики также может оказывать влияние на производительность.

- Если у регистров более трех измерений типа **Строка**, то во время обновления конфигурации базы данных, а также во время загрузки информационной базы может понадобиться табличное пространство для индексов с большим размером ключа. Система «1С:Предприятие» не создает такого табличного пространства, однако может использовать пространство с именем **V81C_INDEX_BIG**.

Создание табличного пространства должно выполняться администратором базы данных с помощью команды **CREATE TABLESPACE**, с указанием длины блока в параметре **BLOCKSIZE**. Чтобы создать табличное пространство с длиной блока, превышающей длину блока базы данных, может потребоваться дополнительная настройка экземпляра базы данных. Например, создание табличного пространства с размером блока 16384 байт можно выполнить, установив параметр **DB_16K_CACHE_SIZE** с помощью команды **ALTER SYSTEM**.

- В запросах, использующих операцию сравнения **ПОДОБНО**, квадратные скобки в шаблоне воспринимаются как специальные символы только в том случае, если они используются в текстовом литерале, а не в выражении.

В следующем примере квадратные скобки имеют специальный смысл – обозначают множество символов:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ * Справочник.Товары ГДЕ Артикул ПОДОБНО "123[АБ]%"
```

В следующем примере квадратные скобки, которые могут находиться в переменной **Шаблон**, не имеют специального символа:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ * Справочник.Товары ГДЕ Артикул ПОДОБНО Шаблон+"%"
```

8.6. Сервер PostgreSQL

- В автоматическом режиме PostgreSQL использует табличные блокировки. Это означает, что транзакция, захватившая одну запись в таблице, блокирует эту таблицу целиком, что может помешать конкурирующим транзакциям работать с данными этой таблицы. Рекомендуется разработку прикладных решений вести в режиме управляемых блокировок с целью максимального использования возможностей СУБД. Использование прикладного решения в автоматическом режиме рекомендуется для совместимости с предыдущими версиями прикладных решений и не рекомендуется для реальной эксплуатации.

- При сортировке по возрастанию поля со значениями **NULL** оказываются последними в выборке, а при сортировке по убыванию – первыми.

- При выполнении операций, связанных с интенсивным удалением и добавлением записей в таблицы базы данных (например, перепроведение документов), может наблюдаться деградация производительности выполняемой операции. Для восстановления исходной производительности рекомендуется регулярно выполнять операцию **REINDEX** (или **VACUUM**) для интенсивно модифицируемых таблиц базы данных. Частота выполнения указанных операций зависит от интенсивности работы с этими таблицами.

- На производительность PostgreSQL оказывает существенное влияние производительность дисковой системы, поскольку по умолчанию параметр **fsync** включен. Это означает, что при выполнении операции **COMMIT** данные сразу переписываются из кеша операционной системы на диск; тем самым гарантируется консистентность при возможном аппаратном сбое. Обратной стороной этого является снижение производительности операций записи на диск, поскольку в данном случае не используются возможности отложенной записи данных операционной системы.

- Повышение производительности возможно при использовании многодисковых **RAID**-массивов, созданных на основе кеширующих **RAID**-контроллеров с энергонезависимой кеш-памятью, и при использовании источников бесперебойного питания (**UPS**). В этом случае задачу по обеспечению консистентности данных при аппаратном сбое берут на себя описанные выше устройства, поэтому

появляется возможность отключения параметра `fsync` и увеличения производительности операций записи на диск. Следует отметить, что увеличение количества дисков в RAID-массиве и объема кеша RAID-контроллера само по себе позволяет компенсировать снижение производительности, обусловленное включением параметра `fsync`.

- Имеются некоторые различия в работе функций даты/времени. Так, в Microsoft SQL `select datediff(hour, datetime(2006, 10, 29, 0, 0, 0), datetime(2006, 10, 30, 0, 0, 0))` вернет 24, а в PostgreSQL – 25 из-за того, что между этими двумя датами был переход на зимнее время. Аналогичная ситуация в отношении учета летнего времени существует для функции `DATEADD`.
- В запросах не рекомендуется использовать **ПОЛНОЕ ВНЕШНЕЕ СОЕДИНЕНИЕ**, т. к. данная конструкция реализована в СУБД недостаточно эффективно. В большинстве случаев без использования этой конструкции можно обойтись, переписав исходный запрос.
- Не допускается одновременно использовать **ПОЛНОЕ ВНЕШНЕЕ СОЕДИНЕНИЕ** и обращение к табличным частям в списке полей выборки.
- В запросах, использующих операцию сравнения **ПОДОБНО**, квадратные скобки в шаблоне воспринимаются как специальные символы только в том случае, если они используются в текстовом литерале, а не в выражении.

В следующем примере квадратные скобки имеют специальный смысл – обозначают множество символов:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ * Справочник.Товары ГДЕ Артикул ПОДОБНО "123[АВ]%"
```

В следующем примере квадратные скобки, которые могут находиться в переменной **Шаблон**, не имеют специального символа:

[Копировать в буфер обмена](#)

```
ВЫБРАТЬ * Справочник.Товары ГДЕ Артикул ПОДОБНО Шаблон+"%"
```

Приложение 9. Работа с XBase

В данном разделе приводится пояснение терминов, использованных при описании средства встроенного языка для работы с базами данных – объекта **XBase**.

9.1. Поля и записи

Хорошим примером базы данных может служить телефонный справочник организации. Он содержит фамилии, номера телефонов и номера комнат всех сотрудников организации. Каждая строка справочника соответствует одной записи, а каждая колонка – полю. Каждое поле имеет наименование и характеристики информации, для хранения которой оно предназначено: тип, длина, точность. Содержимое поля для конкретной записи называется значением поля. Например, телефонный справочник может быть организован в виде таблицы с колонками (полями) **Фамилия** **Имя** **Отчество**, **№ комнаты**, **Телефон**. Каждая строка (запись) содержит информацию об одном абоненте.

9.2. Таблица, структура таблицы, файл базы данных

Весь справочник целиком в терминах баз данных называется таблицей. Состав входящих в таблицу полей определяет структуру таблицы, а состав входящих в таблицу записей – ее содержание. Каждая запись в таблице состоит из того же набора полей, что и таблица целиком, поэтому иногда употребляется термин **структура записи**. Это понятие адекватно структуре таблицы, хотя употребление первого термина представляется более корректным, т. к. таблица имеет структуру независимо от того, имеется ли в ней хотя бы одна запись.

Реализация баз данных формата **DBF** подразумевает, что каждая таблица хранится в отдельном файле. Поэтому в дальнейшем мы будем применять термин «файл базы данных» или «файл БД», имея в виду таблицу базы данных.

9.3. Индексы, выражения индекса и фильтра, индексный файл

Для организации упорядочивания содержимого файла базы данных и поиска в ней по значению одного или нескольких полей применяется механизм индексов. Его применение можно сравнить с сортировкой картотеки по определенному признаку (совокупности признаков). Однако, в отличие от картотеки, файл базы данных может иметь сразу несколько индексов и, соответственно, являться упорядоченным одновременно по нескольким признакам. Каждый индекс имеет наименование, признак уникальности, выражение индекса и фильтр. Наименование индекса используется для идентификации индекса. Выражение индекса и фильтр представляют собой написанные на специальном языке выражения, вычисление значения которых для каждой записи позволяют определить ее место при упорядочивании и необходимость помещения ее в упорядоченный список (индекс может содержать упоминание не обо всех записях таблицы, а только об удовлетворяющих выражению фильтра). Уникальный индекс (имеющий установленный признак уникальности) позволяет иметь в индексе ссылки на записи только с различным значением индексного выражения.

Индексы хранятся в индексном файле. Индексный файл может содержать информацию более чем об одном индексе.

9.4. Запись изменений в базу данных

Каждый объект представляет собой структуру данных, расположенных в памяти компьютера. И изменение содержимого свойств объекта не вызывает немедленного изменения в файлах базы данных. При включенном режиме автосохранения запись содержимого объекта в файлы базы данных происходит при изменении позиционирования (переход к следующей записи, поиск по ключу и т. д.). При выключенном режиме автосохранения запись изменений происходит только при вызове соответствующего метода объекта.

9.5. Работа с индексными файлами

Следует иметь в виду, что объект **XBase** одновременно может быть связан не более чем с одним индексным файлом. Все изменения в базе данных, сделанные в сеансе работы с одним индексным файлом, никак не отражаются на остальных. Поэтому не рекомендуется иметь более одного индексного файла для базы данных. В противном случае после каждого открытия базы данных с индексным файлом,

отличным от открытого в предыдущем сеансе работы с базой, следует производить переиндексацию (обновление содержимого индексного файла).

9.6. Удаление записей

Удаление записи из базы данных не приводит к физическому уничтожению ее на диске. В этом случае в специальном служебном поле записи, недоступном обычными средствами, ставится пометка об удалении. На записи, помеченной удаленной, позиционирования не происходит, если не включен специальный режим просмотра удаленных записей. Имеется свойство, управляющее специальным режимом просмотра, а также набор методов для определения, является ли спозиционированная запись удаленной, и восстановления удаленной записи.

Метод сжатия базы вызывает физическое уничтожение записей, помеченных как удаленные. Метод очистки базы вызывает физическое уничтожение всех записей. После применения этих методов восстановление удаленных записей становится невозможным.

9.7. Создание базы данных, индекса, индексного файла

Помимо работы с существующими базами данных, объект **XBase** имеет набор методов, позволяющих создать новую базу данных произвольной структуры, новые индексы и новый индексный файл. Следует отметить, что если использование методов, изменяющих структуру базы данных, возможно только для объектов, не связанных с существующей базой данных (т. е. для вновь создаваемых баз данных), то создание новых индексов и индексного файла возможно как для создаваемых баз данных, так и для уже существующих и открытых.

9.8. Ограничения

Основное назначение объектов **XBase** – организация экспорта-импорта информации в/из внешних файлов формата **DBF**.

Объекты **XBase** не поддерживают поля типа **memo**. Объекты **XBase** поддерживают только монопольный доступ к файлам. Объекты **XBase** поддерживают индексные файлы в формате **CDX**. Однако использование внешними программами (например, **FoxBase**) индексных файлов, созданных с помощью объектов **XBase**, так же как и использование объектами индексных файлов, созданных внешними программами, не рекомендуется из-за возможной несовместимости версий.

Приложение 10. Особенности использования отдельных механизмов

Работа с HTML-документом

- Элемент `<TABLE>` не может быть добавлен в качестве дочернего элемента в элемент `<P>`. Таким образом, конструкция `<P><TABLE></TABLE></P>` после обработки системой «1С:Предприятие» станет `<P></P><TABLE></TABLE>`. Правила обработки элемента `<P>` указаны в стандарте HTML (<http://www.w3.org/TR/html401/struct/text.html>), в разделе 9.3.1 Paragraphs: the P element.

Форма

- Во время исполнения программного кода, находящегося в обработчике ожидания, не обрабатывается сочетание **Ctrl + Break**, а также не выполняется перерисовка экрана. Также не следует размещать в обработчике ожидания программный код, который будет препятствовать комфортной работе пользователя.
- Оповещение сервера о закрытии формы (а также снятие блокировки с объекта, если он заблокирован) происходит в следующих случаях:
 - закрытие формы, если в форме для редактирования был заблокирован объект;
 - получение формы с сервера с помощью методов `ОткрытьФорму()`, `ПолучитьФорму()`;
 - вызов серверных (контекстных и внеконтекстных) методов формы;
 - закрытие 20 форм;
 - система находится в состоянии ожидания более 20 секунд.
 - При выполнении следующих команд динамического списка:
 - удаление элемента;
 - пометка на удаление элемента;
 - проведение или отмена проведения документа;
 - перенос элемента справочника в другую группу.
- При использовании метода `ОткрытьЗначение()` или при нажатии кнопки `Открыть` (лупа) поля ввода в случае, если прикладное решение выполняется в толстом клиенте, работающем в управляемом режиме, будет открыта форма списка при одновременном выполнении следующих условий:
 - Для объекта конфигурации выбран режим редактирования **В списке** (для справочников, планов обмена, планов видов характеристик, планов счетов, планов видов расчетов, независимых регистров сведений, бизнес-процессов и задач);
 - В прикладном решении существует обычная форма списка, выбранная в качестве основной или вспомогательной;
 - В прикладном решении разрешено использование обычных форм в управляемом приложении;
 - Отсутствует управляемая форма списка или объекта, выбранная в качестве основной или вспомогательной.

Работа с файлами

- При работе с файлами в локальной файловой системе под ОС Windows следует помнить, что длина полного пути к файлу не может превышать 260 символов.

Табличный документ

- Если в ячейке табличного документа находится многострочный текст, который по горизонтали выровнен по центру и выходит за границу ячейки, то при экспорте такого табличного документа в формат Microsoft Excel этот текст будет обрезан левой и правой границами ячейки. Чтобы избежать такого поведения, следует либо создать объединенную ячейку так, чтобы в ней помещался весь текст целиком, либо каждую строку многострочного текста размещать в своей строке табличного документа. При этом не рекомендуется делать отдельный формат для строки, содержащей ячейку с многострочным текстом.
- Если у непустой ячейки табличного документа свойство `Размещение` установлено в значение `Авто`, то ячейки объединяются до следующей заполненной ячейки либо до конца табличного документа по горизонтали.

Работа с почтой

- При работе под управлением ОС Windows, отключение использования протокола OpenSSL при работе с объектом ИнтернетПочта в «1С:Предприятии» версий 8.3.3-8.3.5, а также для последующих версий в режиме совместимости с версией 8.3.5 и ниже, выполняется с помощью редактирования системного реестра. Для этого ключ `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 3.0\Client\Enabled` следует установить в значение `0`. Если ключ отсутствует, то его следует создать. Тип ключа – `DWORD`.

Приложение 11. Правила автоматического формирования имен элементов формы

При обращении к элементам управляемой формы контекстное меню и автоматическая командная панель как к именованным элементам свойства **Элементы** можно использовать стандартные префиксы **КоманднаяПанель**, **КонтекстноеМеню** и **Форма** на обоих вариантах встроеного языка, вне зависимости от того, каким был выбранный вариант на момент создания формы.

СОВЕТ. В случае необходимости вручную формировать имена элементов формы (например, при программном создании) рекомендуется делать это по тем же правилам, как это делает система автоматически. Также не рекомендуется изменять имена элементов формы, которые формирует система при разработке формы (например, при создании элемента формы с помощью перетаскивания реквизита).

В тех случаях, когда имя элемента формируется из пути к связанным данным, формирование происходит следующим образом: имя формируется как путь к связанным данным с исключенными символами «.».

1. Элементы, связанные с данными (поля и таблицы), в том числе связанные с основным реквизитом формы. Имя формируется из полного пути к данным. Если связанные данные принадлежат основному реквизиту, то из имени удаляется имя основного реквизита.

Пример:

Табличная часть **Товары** основного реквизита. Имя таблицы: **Товары**.

В форме редактируется еще один объект, размещенный в реквизите **ДопОбъект**. Таблица, отображающая табличную часть **Товары** этого объекта: **ДопОбъектТовары**.

2. Элементы, связанные с данными, размещенные в таблице (колонки) или связанные с текущими данными таблиц. За исключением таблиц, связанных с основным реквизитом. Имя формируется по следующему правилу **<ИмяЭлементаТаблица><ПутьКДаннымРеквизитаТаблицы - ПутьКДаннымТаблицы>**.

Пример:

Реквизит **Количество** табличной части **Товары** основного реквизита формы: **ТоварыКоличество**.

Реквизит **Количество** табличной части **Товары** не основного реквизита формы **ДопОбъект**: **ДопОбъектТоварыКоличество**.

3. Элементы, связанные с данными, размещенные в таблице (колонки) или связанные с текущими данными таблиц. Для таблиц, связанных с основным реквизитом. Если таблица непосредственно отображает основной реквизит, то имена формируются аналогично описанному в пункте 1. Если таблица связана с реквизитом основного реквизита формы, то имена формируются аналогично описанному в пункте 2.

Пример:

В таблице **Список**, связанной с динамическим списком, являющимся основным реквизитом формы, существует колонка, отображающая реквизит динамического списка **Дата**. Имя колонки: **СписокДата**.

На форме располагается таблица, связанная с табличной частью объектов, отображаемых динамическим списком. Табличная часть имеет имя **Товары**, а реквизит табличной части имеет имя **Количество**. Имя элемента формы таблицы: **Товары**. Имя колонки этого элемента формы: **ТоварыКоличество**.

4. Предопределенные элементы (контекстные меню и автоматические командные панели): **<ИмяЭлементаРодителя>КонтекстноеМеню** и **<ИмяЭлементаРодителя>КоманднаяПанель**.

Пример:

Командная панель формы: `ФормаКоманднаяПанель`.

Контекстное меню колонки `Количество`, связанной с табличной частью `Товары` основного реквизита формы: `ТоварыКоличествоКонтекстноеМеню`.

5. Имя группы, связанной с элементом, являющимся источником команд:

`<ИмяЭлементаИсточника>Действия`.

6. Имя кнопки, размещенной в форме (не в командных панелях или контекстных меню):

`<ИмяКоманды>`. Для команд, порожденных элементом, префикс в виде имени элемента не добавляется.

Пример:

Команда формы `Записать`: `Записать`.

Команда `Изменить` табличной части `Товары`: `Изменить`.

7. Имя кнопки, размещенной в автоматической командной панели с включенным свойством

`Автозаполнение`: `<ИмяЭлементаВладельцаКоманднойПанели><ИмяКоманды>`. Для команд, порожденных элементом, префикс в виде имени элемента не добавляется.

Пример:

Команда формы `Записать`: `ФормаЗаписать`.

8. Имя кнопки, размещенной в контекстном меню с включенным свойством `Автозаполнение`:

`<ИмяЭлементаВладельцаКонтекстногоМеню>КонтекстноеМеню<ИмяКоманды>`. Для команд, порожденных элементом, префикс в виде имени элемента не добавляется.

Пример:

Команда `Подобрать` контекстного меню поля `Дата`: `ДатаКонтекстноеМенюПодобрать`.

9. Имя группы, размещенной в командной панели или контекстном меню без собственного источника команд: `<ИмяИсточникаКомандРодительскогоЭлемента><ИмяГруппы>`. Имена групп не влияют на имена дочерних элементов.

10. Имя группы, размещенной в таблице, не связанной с основным реквизитом формы:

`<ИмяТаблицыФормы><ИмяГруппы>`. Имена групп не влияют на имена дочерних элементов.

Пример:

Группа `ЦенаИКоличество` в таблице `Таблица`: `ТоварыЦенаИКоличество`.

11. Имя кнопок и групп, размещенных в неавтоматических командных панелях с источником, – аналогично пункту 7. Имена кнопок и групп, размещенных в неавтоматических командных панелях без источника команд: имена не модифицируются.

12. Имя группы кнопок, размещенных в командных панелях или контекстных меню с источником команд: `<ИмяЭлементаРодителяКоманднойПанели><ИмяГруппы>`. Каждое имя формируется по правилам, описанным выше. Наличие источника команд не оказывает влияния на формирование имени. Для дочерних элементов они становятся родительскими группами по описанным выше правилам. Префикс `КоманднаяПанель` не добавляется.

Пример:

В контекстное меню колонки `УслугиКоличество` добавлена группа кнопок с источником данных `Форма`. Стандартное имя группы: `УслугиКоличествоКонтекстноеМенюФормаДействия`.

Приложение 12. Описание сущностей, предоставляемых через стандартный интерфейс OData

Подробное описание представления данных см. [здесь](#). При описании свойства описание представления дополнительно не приводится.

12.1. Константы

Имя сущности	<code>Constant_<Имя></code>
Имя набора сущностей	<code>Constant_<Имя></code>

Состав свойств:

Имя	Тип	Комментарий
<code>Value</code>	Соответствует типу значения	Значение константы.
<code><Общие реквизиты></code>	Соответствует типу разделителя	Кроме общих реквизитов являющихся независимыми разделителями.
<code>Surrogate_Key</code>	<code>Edm.Int32</code>	Суррогатный ключ в том случае, если отсутствуют разделители. Всегда равен 0.

Особенности поведения:

- Удаление не поддерживается (запрос DELETE);
- Создание поддерживается (запрос POST) с особенностями.

Если значение константы в информационной базе имеет значение **Неопределено**, то выполняется установка значения константы. Если значение константы в информационной базе имеет значение, отличное от **Неопределено** – генерируется ошибка.

- Не поддерживается обращение «через точку».

12.2. Планы обмена

Имя сущности	<code>ExchangePlan_<Имя></code>
Имя набора сущностей	<code>ExchangePlan_<Имя></code>

Состав свойств:

Имя	Тип	Комментарий
<code>Code</code>	<code>Edm.String</code> <code>Edm.Int32</code>	Код.
<code>Description</code>	<code>Edm.String</code>	Наименование.
<code>Ref</code>	Ссылочный тип	Ссылка.
<code>DeletionMark</code>	<code>Edm.Boolean</code>	Пометка удаления.
<code>Version</code>	<code>Edm.String</code>	Версия данных.
<code>ReceivedNo</code>	<code>Edm.Int32</code>	Номер последнего полученного сообщения.
<code>SentNo</code>	<code>Edm.Int32</code>	Номер последнего отправленного сообщения.
<code><Реквизиты></code>	Соответствует типу реквизита	
<code><Общие реквизиты></code>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.
<code><Табличные части></code>	<code>Collection(СоставнойТип)</code>	Каждая табличная часть выступает в виде отдельной сущности с именем <code>ExchangePlan_<ИмяОбъекта>_<ИмяТабличнойЧасти></code> .

12.2.1. Табличная часть плана обмена

Имя сущности	ExchangePlan_<Имя объекта>_<Имя табличной части>_RecordType
Имя набора сущностей	ExchangePlan_<Имя объекта>_<Имя табличной части>_RecordType

Состав свойств:

Имя	Тип	Комментарий
Ref	Ссылочный тип	Ссылка.
LineNumber	Edm.Int64	Номер строки.
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

12.2.2. Строка табличной части

Имя составного типа: [ExchangePlan_<Имя объекта>_<Имя табличной части>_RowType](#).

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки табличной части
<Реквизиты>	Соответствует типу реквизита	

12.3. Справочники

Имя сущности	Catalog_<Имя>
Имя набора сущностей	Catalog_<Имя>

Состав свойств:

Имя	Тип	Комментарий
Ref	Ссылочный тип	Ссылка.
Code	Edm.String, Edm.Int32	При наличии у справочника кода.
Description	Edm.String	При наличии у справочника наименования.
IsFolder	Edm.Boolean	Для иерархического справочника.
Owner	Ссылочный или составной тип	Владелец для подчиненного справочника.
Parent	Ссылочный тип	Родитель для иерархического справочника.
Version	Edm.String	Версия данных.
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.
<Табличные части>	Collection(СоставнойТип)	Каждая табличная часть выступает в виде отдельной сущности с именем Catalog_<ИмяОбъекта>_<ИмяТабличнойЧасти> .

12.3.1. Табличная часть справочника

Имя сущности	Catalog_<Имя объекта>_<Имя табличной части>_RecordType
Имя набора сущностей	Catalog_<Имя объекта>_<Имя табличной части>_RecordType

Состав свойств:

--	--	--

Имя	Тип	Комментарий
Ref	Ссылочный тип	Ссылка.
LineNumber	Edm.Int64	Номер строки.
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

12.3.2. Строка табличной части

Имя составного типа: `Catalog_<Имя объекта>_<Имя табличной части>_RowType`.

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки табличной части.
<Реквизиты>	Соответствует типу реквизита	

12.4. Документы

Имя сущности	<code>Document_<Имя></code>
Имя набора сущностей	<code>Document_<Имя></code>

Состав свойств:

Имя	Тип	Комментарий
Ref	Ссылочный тип	Ссылка.
Number	Edm.String, Edm.Int32	Номер документа, в зависимости от установленного типа номера.
Date	Edm.Date	Если включено.
DeletionMark	Edm.Boolean	Пометка удаления.
Version	Edm.String	Версия данных.
Posted	Edm.Boolean	Признак проведения документа.
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.
<Табличные части>	<code>Collection(СоставнойТип)</code>	Каждая табличная часть выступает в виде отдельной сущности с именем <code>Document_<ИмяОбъекта>_<ИмяТабличнойЧасти></code> .

12.4.1. Методы

Post

Описание:

Выполняет проведение документа.

Параметры:

`PostingModeOperational` Edm.Boolean

Определяет необходимость проведения документа в оперативном режиме. При необходимости такого проведения значение свойства должно принимать значение `true`.

Значение по умолчанию: `false`.

Unpost

Выполняет отмену проведения документа.

Параметры:

PostingModeOperational Edm.Boolean

Определяет необходимость проведения документа в оперативном режиме. При необходимости такого проведения значение свойства должно принимать значение **true**.

Значение по умолчанию: **false**.

12.4.2. Табличная часть справочника

Имя сущности	Document_<Имя объекта>_<Имя табличной части>_RecordType
Имя набора сущностей	Document_<Имя объекта>_<Имя табличной части>_RecordType

Состав свойств:

Имя	Тип	Комментарий
Ref	Ссылочный тип	Ссылка.
LineNumber	Edm.Int64	Номер строки.
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

12.4.3. Строка табличной части

Имя составного типа: [Document_<Имя объекта>_<Имя табличной части>_RowType](#).

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки табличной части
<Реквизиты>	Соответствует типу реквизита	

12.5. Журналы документов

Имя сущности	DocumentJournal_<Имя>
Имя набора сущностей	DocumentJournal_<Имя>

Состав свойств:

Имя	Тип	Комментарий
Ref	Ссылочный тип	
Number	Edm.String, Edm.Int32	Номер документа, в зависимости от установленного типа номера.
Date	Edm.Date	Если включено.
DeletionMark	Edm.Boolean	Пометка удаления.
Type	Edm.String	Представление типа документа.
Posted	Edm.Boolean	Признак проведения документа.
<Графы>	Соответствует типу графы	
<Реквизиты>	Соответствует типу реквизита	

<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.
-------------------	------------------------------	---

Особенности поведения:

- Поддерживается только получение списка строк
- Не поддерживаются:
 - Создание документа (запрос POST);
 - Изменение документа (запрос PATCH);
 - Удаление документа (запрос DELETE);
 - Обращение «через точку».

12.6. Планы видов характеристик

Имя сущности	ChartOfCharacteristicTypes_<Имя>
Имя набора сущностей	ChartOfCharacteristicTypes_<Имя>

Состав свойств:

Имя	Тип	Комментарий
Ref	Ссылочный тип	Ссылка.
Code	Edm.String, Edm.Int32	При наличии у плана видов характеристик кода.
Description	Edm.String	При наличии у плана видов характеристик наименования.
IsFolder	Edm.Boolean	Для иерархического плана видов характеристик.
Owner	Ссылочный или составной тип	Владелец (для подчиненного плана видов характеристик).
Parent	Ссылочный тип	Родитель (для иерархических планов видов характеристик).
DeletionMark	Edm.Boolean	Пометка удаления.
Version	Edm.String	Версия данных.
ValueType	EnterpriseV8.TypeDescription	Тип значения характеристики.
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.
<Табличные части>	Collection(СоставнойТип)	Каждая табличная часть выступает в виде отдельной сущности с именем ChartOfCharacteristicTypes_<ИмяОбъекта>_<ИмяТабличнойЧасти> .

12.6.1. Табличная часть плана видов характеристик

Имя сущности	ChartOfCharacteristicTypes_<Имя объекта>_<Имя табличной части>_RecordType
Имя набора сущностей	ChartOfCharacteristicTypes_<Имя объекта>_<Имя табличной части>_RecordType

Состав свойств:

Имя	Тип	Комментарий
Ref	Ссылочный тип	Ссылка.
LineNumber	Edm.Int64	Номер строки.
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

12.6.2. Строка табличной части

Имя составного типа: [ChartOfCharacteristicTypes_<Имя объекта>_<Имя табличной части>_RowType](#).

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки табличной части.
<Реквизиты>	Соответствует типу реквизита	

12.7. Планы счетов

Имя сущности	ChartOfAccounts_<Имя>
Имя набора сущностей	ChartOfAccounts_<Имя>

Состав свойств:

Имя	Тип	Комментарий
Ref	Ссылочный тип	Ссылка.
Code	Edm.String, Edm.Int32	При наличии у плана счетов кода.
Description	Edm.String	При наличии у плана счетов наименования.
OffBalance	Edm.Boolean	Признак забалансового счета.
Order	Edm.String	Порядок сортировки.
Type	Edm.String	Тип счета: активный, пассивный, активно/пассивный.
Parent	Ссылочный тип	Родитель (для иерархического плана счетов).
DeletionMark	Edm.Boolean	Пометка удаления.
Version	Edm.String	Версия данных.
<ПризнакиУчета>	Edm.Boolean	Признаки учета.
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.
<Табличные части>	Collection(СоставнойТип)	Каждая табличная часть выступает в виде отдельной сущности с именем ChartOfAccounts_<ИмяОбъекта>_<ИмяТабличнойЧасти> .
ExtDimensionTypes	Collection(СоставнойТип)	Стандартная табличная часть «Виды субконто».

12.7.1. Табличная часть плана счетов

Имя сущности	ChartOfAccounts_<Имя объекта>_<Имя табличной части>_RecordType
Имя набора сущностей	ChartOfAccounts_<Имя объекта>_<Имя табличной части>_RecordType

Состав свойств:

Имя	Тип	Комментарий
Ref	Ссылочный тип	Ссылка.
LineNumber	Edm.Int64	Номер строки.
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

12.7.2. Строка табличной части

Имя составного типа: [ChartOfAccounts_<Имя объекта>_<Имя табличной части>_RowType](#).

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки табличной части.
<Реквизиты>	Соответствует типу реквизита	

12.7.3. Строка стандартной табличной части «Виды субконто»

Имя составного типа: [ChartOfAccounts_<Имя объекта>_ExtDimensions](#).

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки табличной части
ExtDimensionType_Key	Edm.Guid	Идентификатор типа плана видов характеристик, хранящий виды субконто
Predefined	Edm.Boolean	Признак predetermined вида субконто
<Реквизиты>	Соответствует типу реквизита	
<ПризнакиУчета>	Edm.Boolean	Признаки учета.

12.8. Планы видов расчета

Имя сущности	ChartOfCalculationTypes_<Имя>
Имя набора сущностей	ChartOfCalculationTypes_<Имя>

Состав свойств:

Имя	Тип	Комментарий
Ref	Ссылочный тип	Ссылка.
Code	Edm.String, Edm.Int32	При наличии у плана видов расчета кода.
Description	Edm.String	При наличии у плана видов расчета наименования.
ActionPeriodIsBasic	Edm.Boolean	Признак того, что базовым периодом является период действия (только для планов видов расчета с поддержкой периода действия).
DeletionMark	Edm.Boolean	Пометка удаления.
Version	Edm.String	Версия данных.
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.
<Табличные части>	Collection(СоставнойТип)	Каждая табличная часть выступает в виде отдельной сущности с именем ChartOfCalculationTypes_<ИмяОбъекта>_<ИмяТабличнойЧасти> .
LeadingCalculationTypes	Collection(СоставнойТип)	Стандартная табличная часть «Ведущие виды расчета».
DisplacingCalculationTypes	Collection(СоставнойТип)	Стандартная табличная часть «Вытесняющие виды расчета» (только для планов видов расчета с поддержкой периода действия).
BaseCalculationTypes	Collection(СоставнойТип)	Стандартная табличная часть «Базовые виды расчета» (только для плана видов расчета с поддержкой базового периода действия).

12.8.1. Табличная часть плана расчетов

Имя сущности	ChartOfCalculationTypes_<Имя объекта>_<Имя табличной части>_RecordType
Имя набора сущностей	ChartOfCalculationTypes_<Имя объекта>_<Имя табличной части>_RecordType

Состав свойств:

Имя	Тип	Комментарий
Ref	Ссылочный тип	Ссылка.
LineNumber	Edm.Int64	Номер строки.
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

12.8.2. Строка табличной части

Имя составного типа: [ChartOfCalculationTypes_<Имя объекта>_<Имя табличной части>_RowType](#).

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки табличной части.
<Реквизиты>	Соответствует типу реквизита	

12.8.3. Строка стандартной табличной части «Ведущие виды расчета»

Имя составного типа: [ChartOfCalculationTypes_<Имя объекта>_LeadingCalculationTypes](#).

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки табличной части.
CalculationType	Edm.Guid	Идентификатор типа плана видов расчета, хранящий виды расчета.
Predefined	Edm.Boolean	Признак predetermined вида субконто.

12.8.4. Строка стандартной табличной части «Вытесняющие виды расчета»

Имя составного типа: [ChartOfCalculationTypes_<Имя объекта>_BaseCalculationTypes](#).

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки табличной части.
CalculationType	Edm.Guid	Идентификатор типа плана видов расчета, хранящий виды расчета.
Predefined	Edm.Boolean	Признак predetermined вида субконто.

12.8.5. Строка стандартной табличной части «Базовые виды расчета»

Имя составного типа: [ChartOfCalculationTypes_<Имя объекта>_DisplacingCalculationTypes](#).

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки табличной части.
CalculationType_Key	Edm.Guid	Идентификатор типа плана видов расчета, хранящий виды расчета.
Predefined	Edm.Boolean	Признак predetermined вида субконто.

12.9. Регистры сведений

12.9.1. Независимый

Имя сущности	InformationRegister_<Имя>
Имя набора сущностей	InformationRegister _<Имя>

Состав свойств:

Название в схеме	Тип	Комментарий
Period	Edm.Date	Период записи. Только для периодического регистра сведений.
<Измерения>	Соответствует типу измерения	Измерения включаются в ключ сущности.
<Ресурсы>	Соответствует типу ресурса	
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

Особенности поведения:

Измерения регистра сведений образуют ключ записи. При попытке изменить ключ записи будет генерироваться исключение с HTTP-статусом 404. Рекомендуется в таком случае вначале удалять существующую запись, а затем добавлять новую.

12.9.1.1. Функции

SliceLast

Описание:

Набор сущностей – срез последних периодического регистра сведений.

Параметры:

Period Edm.Date

Период получения среза.

Condition Edm.String

Отбор записей в формате OData. Подробнее см. [здесь](#).

SliceFirst

Описание:

Набор сущностей – срез первых периодического регистра сведений.

Параметры:

Period Edm.Date

Период получения среза.

Condition Edm.String

Отбор записей в формате OData. Подробнее см. [здесь](#).

12.9.2. Подчиненный регистратору

Имя сущности	InformationRegister_<RegisterName>
Имя набора сущностей	InformationRegister _<RegisterName>

Состав свойств:

Имя	Тип	Комментарий
Recorder	Edm.String	Регистратор.
Recorder_Type	Edm.String	Ссылка на регистратор.
RecordSet	Collection(СоставнойТип)	Набор записей.

Особенности поведения:

Измерения регистра сведений образуют ключ записи. При попытке изменить ключ записи будет генерироваться исключение с HTTP-статусом 404. Рекомендуется в таком случае вначале удалять существующую запись, а затем добавлять новую.

12.9.2.1. Функции

SliceLast

Описание:

Набор сущностей – срез последних периодического регистра сведений.

Параметры:

Period Edm.Date

Период получения среза.

Condition Edm.String

Отбор записей в формате OData. Подробнее см. [здесь](#).

SliceFirst

Описание:

Набор сущностей – срез первых периодического регистра сведений.

Параметры:

Period Edm.Date

Период получения среза.

Condition Edm.String

Отбор записей в формате OData. Подробнее см. [здесь](#).

12.9.2.2. Набор записей регистра

Имя составного типа: [InformationRegister_<Имя>_RowType](#)

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки в наборе записей
Active	Edm.Boolean	Признак активности строки
<Измерения>	Соответствует типу измерения	
<Ресурсы>	Соответствует типу ресурса	
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

12.9.2.3. Строка набора записей регистров сведений

Имя сущности	InformationRegister_<Имя>_RecordType
Имя набора сущностей	InformationRegister_<Имя>_RecordType

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки в наборе записей.
Active	Edm.Boolean	Признак активности строки.
<Измерения>	Соответствует типу измерения	
<Ресурсы>	Соответствует типу ресурса	
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

Особенности поведения:

Сущность доступна только на чтение (запрос GET).

12.10. Регистр накопления

Имя сущности	AccumulationRegister_<RegisterName>
Имя набора сущностей	AccumulationRegister_<RegisterName>

Состав свойств:

Имя	Тип	Комментарий
Recorder	Edm.String	Регистратор.
Recorder_Type	Edm.String	Ссылка на регистратор.
RecordSet	Collection(СоставнойТип)	Набор записей.

Особенности поведения:

Измерения регистра сведений образуют ключ записи. При попытке изменить ключ записи будет генерироваться исключение с HTTP-статусом 404. Рекомендуется в таком случае вначале удалять существующую запись, а затем добавлять новую.

12.10.1. Функции

Balance

Описание:

Обеспечивает доступ к виртуальной таблице остатков регистра накопления.

Параметры:

Period Edm.Date

Дата, на которую необходимо получить остатки регистра накопления.

Condition Edm.String

Отбор записей в формате OData. Подробнее см. [здесь](#).

Dimensions Edm.String

Список измерений через запятую.

Turnovers

Описание:

Обеспечивает доступ к виртуальной таблице оборотов регистра накопления. Разворот итогов по периодам не поддерживается.

Параметры:

Condition Edm.String

Отбор записей в формате OData. Подробнее см. стр. 1103.

EndPeriod Edm.Date

Конец периода, за который требуется получить обороты.

StartPeriod Edm.Date

Начало периода, за который требуется получить обороты.

BalanceAndTurnovers**Описание:**

Обеспечивает доступ к виртуальной таблице остатков и оборотов регистра накопления. Разворот итогов по периодам не поддерживается.

Параметры:

Condition Edm.String

Отбор записей в формате OData. Подробнее см. [здесь](#).

EndPeriod Edm.Date

Конец периода, за который требуется получить обороты.

StartPeriod Edm.Date

Начало периода, за который требуется получить обороты.

12.10.2. Набор записей регистра

Имя составного типа: [AccumulationRegister_<RegisterName>_RowType](#).

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки в наборе записей.
Active	Edm.Boolean	Признак активности строки.
<Измерения>	Соответствует типу измерения	
<Ресурсы>	Соответствует типу ресурса	
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

Особенности поведения:

Измерения регистра сведений образуют ключ записи. При попытке изменить ключ записи будет выдаваться исключение, код HTTP 404. Правильный способ – удалить запись и создать новую.

12.10.3. Строка набора записей регистров сведений

Имя сущности	AccumulationRegister_<Имя>_RecordType
Имя набора сущностей	AccumulationRegister_<Имя>_RecordType

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки в наборе записей.
Active	Edm.Boolean	Признак активности строки.
<Измерения>	Соответствует типу измерения	
<Ресурсы>	Соответствует типу ресурса	
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

Особенности поведения:

Сущность доступна только на чтение (запрос GET).

12.11. Регистры бухгалтерии

Имя сущности	AccountingRegister_<RegisterName>
Имя набора сущностей	AccountingRegister_<RegisterName>

Состав свойств:

Имя	Тип	Комментарий
Recorder	Edm.String	Регистратор.
Recorder_Type	Edm.String	Ссылка на регистратор.
RecordSet	Collection(СоставнойТип)	Набор записей.

Особенности поведения:

Измерения регистра сведений образуют ключ записи. При попытке изменить ключ записи будет генерироваться исключение с HTTP-статусом 404. Рекомендуется в таком случае вначале удалять существующую запись, а затем добавлять новую.

12.11.1. Функции

Balance

Описание:

Обеспечивает доступ к виртуальной таблице остатков регистра бухгалтерии.

Параметры:

Period Edm.Date

Дата, на которую необходимо получить остатки регистра бухгалтерии.

Condition Edm.String

Отбор записей в формате OData. Подробнее см. [здесь](#).

Dimensions Edm.String

Список измерений через запятую.

AccountCondition Edm.String

Условие отбора по счетам в формате OData. Подробнее см. [здесь](#).

ExtraDimensions Edm.String

Перечень уникальных идентификаторов видов субконто, по которым необходимо установить отбор.

Turnovers**Описание:**

Обеспечивает доступ к виртуальной таблице оборотов регистра бухгалтерии. Разворот итогов по периодам не поддерживается.

Параметры:

Condition Edm.String

Отбор записей в формате OData. Подробнее см. [здесь](#).

EndPeriod Edm.Date

Конец периода, за который требуется получить обороты.

StartPeriod Edm.Date

Начало периода, за который требуется получить обороты.

AccountCondition Edm.String

Условие отбора по счетам в формате OData. Подробнее см. [здесь](#).

BalanceAccountCondition Edm.String

Условие отбора по корреспондирующим счетам в формате OData. Подробнее см. [здесь](#).

Dimensions Edm.String

Список измерений через запятую.

ExtraDimensions Edm.String

Перечень уникальных идентификаторов видов субконто, по которым необходимо установить отбор.

BalancedExtraDimensions Edm.String

Перечень уникальных идентификаторов корреспондирующих видов субконто, по которым необходимо установить отбор.

BalanceAndTurnovers**Описание:**

Обеспечивает доступ к виртуальной таблице остатков и оборотов регистра бухгалтерии. Разворот итогов по периодам не поддерживается.

Параметры:

Condition Edm.String

Отбор записей в формате OData. Подробнее см. [здесь](#).

EndPeriod Edm.Date

Конец периода, за который требуется получить обороты.

StartPeriod Edm.Date

Начало периода, за который требуется получить обороты.

ExtDimensions**Описание:**

Обеспечивает доступ к виртуальной таблице субконто регистра бухгалтерии.

RecordsWithExtDimensions**Описание:**

Обеспечивает доступ к виртуальной таблице движений с субконто регистра бухгалтерии.

Параметры:

Condition Edm.String

Отбор записей в формате OData. Подробнее см. [здесь](#).

EndPeriod Edm.Date

Конец периода, за который требуется получить обороты.

StartPeriod Edm.Date

Начало периода, за который требуется получить обороты.

Top Edm.Int32

Ограничение максимального количества записей.

OrderBy Edm.String

Строка, содержащую имена колонок (через запятую), по которым будет выполняться сортировка проводок.

DrCrTurnover

Описание:

Обеспечивает доступ к виртуальной таблице получать обороты по регистру бухгалтерии за произвольный период с определенной агрегацией. Разворот итогов по периодам не поддерживается.

Параметры:

Condition Edm.String

Отбор записей в формате OData. Подробнее см. [здесь](#).

EndPeriod Edm.Date

Конец периода, за который требуется получить обороты.

StartPeriod Edm.Date

Начало периода, за который требуется получить обороты.

AccountCondition Edm.String

Условие отбора по счетам в формате OData. Подробнее см. [здесь](#).

BalanceAccountCondition Edm.String

Условие отбора по корреспондирующим счетам в формате OData. Подробнее см. [здесь](#).

Dimensions Edm.String

Список измерений через запятую.

ExtraDimensions Edm.String

Перечень уникальных идентификаторов видов субконто, по которым необходимо установить отбор.

BalancedExtraDimensions Edm.String

Перечень уникальных идентификаторов корреспондирующих видов субконто, по которым необходимо установить отбор.

12.11.2. Набор записей регистра

Имя составного типа: [AccountingRegister_<RegisterName>_RowType](#).

Состав свойств:

--	--	--

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки в наборе записей.
Active	Edm.Boolean	Признак активности строки.
Period	Edm.Date	Момент времени, в который выполнена запись.
RecordType	Edm.String	Вид движения записи регистра бухгалтерии: Debit или Credit .
AccountDr_Key	Edm.Guid	Счет дебета (только для регистра с корреспонденцией).
AccountCr_Key	Edm.Guid	Счет кредита (только для регистра с корреспонденцией).
Account_Key	Edm.Guid	Счет (только для регистра без корреспонденции).
<Измерения>	Соответствует типу измерения	Для небалансовых измерений два свойства: <ul style="list-style-type: none"> • <ИмяИзмерения>Dr • <ИмяИзмерения>Cr.
<Ресурсы>	Соответствует типу ресурса	Для небалансовых измерений два свойства: <ul style="list-style-type: none"> • <ИмяРесурса>Dr • <ИмяРесурса>Cr.
<Реквизиты>	Соответствует типу реквизита	•
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

12.11.3. Строка набора записей регистров сведений

Имя сущности	AccountingRegister_<Имя>_RecordType
Имя набора сущностей	AccountingRegister_<Имя>_RecordType

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки в наборе записей.
Active	Edm.Boolean	Признак активности строки.
Period	Edm.Date	Момент времени, в который выполнена запись.
RecordType	Edm.String	Вид движения записи регистра бухгалтерии: Debit или Credit .
AccountDr_Key	Edm.Guid	Счет дебета (только для регистра с корреспонденцией).
AccountCr_Key	Edm.Guid	Счет кредита (только для регистра с корреспонденцией).
Account_Key	Edm.Guid	Счет (только для регистра без корреспонденции).
<Измерения>	Соответствует типу измерения	Для небалансовых измерений два свойства: <ul style="list-style-type: none"> • <ИмяИзмерения>Dr • <ИмяИзмерения>Cr.
<Ресурсы>	Соответствует типу ресурса	Для небалансовых измерений два свойства: <ul style="list-style-type: none"> • <ИмяРесурса>Dr • <ИмяРесурса>Cr.
<Реквизиты>	Соответствует типу реквизита	•
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

Особенности поведения:

Сущность доступна только на чтение (запрос GET).

12.12. Регистры расчета

Имя сущности	CalculationRegister_<RegisterName>
Имя набора сущностей	CalculationRegister_<RegisterName>

Состав свойств:

Имя	Тип	Комментарий
Recorder	Edm.String	Регистратор.
Recorder_Type	Edm.String	Ссылка на регистратор.
RecordSet	Collection(СоставнойТип)	Набор записей.

Особенности поведения:

Измерения регистра сведений образуют ключ записи. При попытке изменить ключ записи будет генерироваться исключение с HTTP-статусом 404. Рекомендуется в таком случае вначале удалять существующую запись, а затем добавлять новую.

12.12.1. Функции

ScheduledData

Описание:

Обеспечивает доступ к виртуальной таблице данных графика регистра расчета. Только для регистра расчета с заданным графиком.

Параметры:

Condition Edm.String

Отбор записей в формате OData. Подробнее см. стр. 1103.

ActualActionPeriod

Описание:

Обеспечивает доступ к виртуальной таблице фактического периода действия. Только для регистра с поддержкой периода действия.

Параметры:

Condition Edm.String

Отбор записей в формате OData. Подробнее см. стр. 1103.

<ИмяПерерасчета>

Описание:

Обеспечивает доступ к виртуальной таблице записей перерасчета.

Base<ИмяБазовогоРегистраРасчета>

Описание:

Обеспечивает доступ к виртуальной таблице базовых данных регистра расчета.

Параметры:

Condition Edm.String

Отбор записей в формате OData. Подробнее см. [здесь](#).

MainRegisterDimensions Edm.String

Имена измерений основного регистра расчета, по которому строится таблица базовых данных, через запятую.

Имена измерений базового регистра расчета, по которому строится таблица базовых данных, через запятую.

ViewPoints Edm.String

Имена полей базового регистра расчета, по которым производится суммирование базовых данных, через запятую

12.12.2. Набор записей регистра

Имя составного типа: [CalculationRegister_<RegisterName>_RowType](#).

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки в наборе записей.
Active	Edm.Boolean	Признак активности строки.
Period	Edm.Date	Момент времени, в который выполнена запись.
BegOfBasePeriod	Edm.Date	Дата начала базового периода учетной записи. Только для регистров с базовым периодом.
EndOfBasePeriod	Edm.Date	Дата окончания базового периода учетной записи. Только для регистров расчета с базовым периодом.
BegOfActionPeriod	Edm.Date	Дата начала периода действия. Только для регистров расчета с периодом действия.
EndOfAcrionPeriod	Edm.Date	Дата окончания периода действия. Только для регистров расчета с периодом действия.
CalculationType_Key	Edm.Guid	Ссылка на вид расчета учетной записи.
ReversingEntry	Edm.Boolean	Признак сторно-записи.
<Измерения>	Соответствует типу измерения	Для небалансовых измерений два свойства: <ul style="list-style-type: none"> • <ИмяИзмерения>Dr • <ИмяИзмерения>Cr.
<Ресурсы>	Соответствует типу ресурса	Для небалансовых измерений два свойства: <ul style="list-style-type: none"> • <ИмяРесурса>Dr • <ИмяРесурса>Cr.
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

12.12.3. Строка набора записей регистров сведений

Имя сущности	CalculationRegister_<Имя>_RecordType
Имя набора сущностей	CalculationRegister_<Имя>_RecordType

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки в наборе записей.
Active	Edm.Boolean	Признак активности строки.
Period	Edm.Date	Момент времени, в который выполнена запись.
BegOfBasePeriod	Edm.Date	Дата начала базового периода учетной записи. Только для регистров с базовым периодом.
EndOfBasePeriod	Edm.Date	Дата окончания базового периода учетной записи. Только для регистров расчета с базовым периодом.

BegOfActionPeriod	Edm.Date	Дата начала периода действия. Только для регистров расчета с периодом действия.
EndOfAcrionPeriod	Edm.Date	Дата окончания периода действия. Только для регистров расчета с периодом действия.
CalculationType_Key	Edm.Guid	Ссылка на вид расчета учетной записи.
ReversingEntry	Edm.Boolean	Признак сторно-записи.
<Измерения>	Соответствует типу измерения	Для небалансовых измерений два свойства: <ul style="list-style-type: none"> • <ИмяИзмерения>Dr • <ИмяИзмерения>Cr.
<Ресурсы>	Соответствует типу ресурса	Для небалансовых измерений два свойства: <ul style="list-style-type: none"> • <ИмяРесурса>Dr • <ИмяРесурса>Cr.
<Реквизиты>	Соответствует типу реквизита	•
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

Особенности поведения:

Сущность доступна только на чтение (запрос GET).

12.13. Бизнес-процессы

Имя сущности	BusinessProcess_<Имя>
Имя набора сущностей	BusinessProcess_<Имя>

Состав свойств:

Имя	Тип	Комментарий
Ref	Ссылочный тип	Ссылка.
Code	Edm.String, Edm.Int32	При наличии у бизнес-процесса кода.
Description	Edm.String	При наличии у бизнес-процесса наименования.
DeletionMark	Edm.Boolean	Пометка удаления.
Version	Edm.String	Версия данных.
Started	Edm.Boolean	Бизнес-процесс стартован.
Completed	Edm.Boolean	Бизнес-процесс завершен.
HeadTask		Два поля: <ul style="list-style-type: none"> • <Имя реквизита> – содержит значение ссылки в виде строки; • <Имя реквизита>_Type – ссылка на задачу, обслуживающую данный бизнес-процесс.
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.
<Табличные части>	Collection(СоставнойТип)	Каждая табличная часть выступает в виде отдельной сущности с именем BusinessProcess_<ИмяОбъекта>_<ИмяТабличнойЧасти>.

12.13.1. Функции

Start

Стартует бизнес-процесс.

Параметры:

RoutePoint

Имя точки маршрута, с которой стартует бизнес-процесс. Необязательный параметр.

12.13.2. Табличная часть бизнес-процесса

Имя сущности	BusinessProcess_<Имя объекта>_<Имя табличной части>_RecordType
Имя набора сущностей	BusinessProcess_<Имя объекта>_<Имя табличной части>_RecordType

Состав свойств:

Имя	Тип	Комментарий
Ref	Ссылочный тип	Ссылка.
LineNumber	Edm.Int64	Номер строки.
<Реквизиты>	Соответствует типу реквизита	•
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

12.13.3. Строка табличной части

Имя составного типа: [BusinessProcess_<Имя объекта>_<Имя табличной части>_RowType](#).

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки табличной части.
<Реквизиты>	Соответствует типу реквизита	

12.14. Задачи

Имя сущности	Task_<Имя>
Имя набора сущностей	Task_<Имя>

Состав свойств:

Имя	Тип	Комментарий
Ref	Ссылочный тип	Ссылка.
Number	Edm.String , Edm.Int32	При наличии у задачи номера.
Description	Edm.String	При наличии у задачи наименования.
DeletionMark	Edm.Boolean	Пометка удаления.
Version	Edm.String	Версия данных.
Executed	Edm.Boolean	Задача выполнена.
BusinessProcess		Описывает составной тип, если задача связана с несколькими бизнес-процессами: <ul style="list-style-type: none"> <Имя реквизита> – содержит значение ссылки в виде строки; <Имя реквизита>_Type – ссылка на задачу, обслуживающую данный бизнес-процесс.

		Если задача связана с одним бизнес-процессом, то реквизит описывается двумя свойствами: <ul style="list-style-type: none"> • Навигационная ссылка; • BusinessProcess_Key – значение ссылки.
RoutePoint	Edm.String	Имя точки маршрута, для исполнения которой создана задача. Только для задачи, привязанной к бизнес-процессу.
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.
<Табличные части>	Collection(СоставнойТип)	Каждая табличная часть выступает в виде отдельной сущности с именем Task_<ИмяОбъекта>_<ИмяТабличнойЧасти> .

12.14.1. Функции

ExecuteTask

Описание:

Выполняет задачу.

12.14.2. Табличная часть бизнес-процесса

Имя сущности	Task_<Имя объекта>_<Имя табличной части>_RecordType
Имя набора сущностей	Task_<Имя объекта>_<Имя табличной части>_RecordType

Состав свойств:

Имя	Тип	Комментарий
Ref	Ссылочный тип	
LineNumber	Edm.Int64	Номер строки.
<Реквизиты>	Соответствует типу реквизита	
<Общие реквизиты>	Соответствует типу реквизита	Кроме общих реквизитов являющихся независимыми разделителями.

12.14.3. Строка табличной части

Имя составного типа: **Task_<Имя объекта>_<Имя табличной части>_RowType**.

Состав свойств:

Имя	Тип	Комментарий
LineNumber	Edm.Int64	Номер строки табличной части.
<Реквизиты>	Соответствует типу реквизита	

Приложение 13. Соответствие синхронных методов асинхронным аналогам

При переработке существующих алгоритмов, использующих синхронные вызовы, на асинхронную работу, будет полезна следующая таблица, в которой приведено соответствие синхронных и асинхронных методов глобального контекста и различных объектов:

Объект	Синхронный метод	Асинхронный метод
Глобальный контекст	ОткрытьФормуМодально ()	ОткрытьФорму ()
	Вопрос ()	ПоказатьВопрос ()
	ОткрытьЗначение ()	ПоказатьЗначение ()
	Предупреждение ()	ПоказатьПредупреждение ()
	ВвестиДату ()	ПоказатьВводДаты ()
	ВвестиЗначение ()	ПоказатьВводЗначения ()
	ВвестиСтроку ()	ПоказатьВводСтроки ()
	ВвестиЧисло ()	ПоказатьВводЧисла ()
	УстановитьВнешнююКомпоненту ()	НачатьУстановкуВнешнейКомпоненты ()
	УстановитьРасширениеРаботыСФайлами ()	НачатьУстановкуРасширенияРаботыСФайлами ()
	УстановитьРасширениеРаботыСКриптографией ()	НачатьУстановкуРасширенияРаботыСКриптографией ()
	ПодключитьРасширениеРаботыСКриптографией ()	НачатьПодключениеРасширенияРаботыСКриптографией ()
	ПодключитьРасширениеРаботыСФайлами ()	НачатьПодключениеРасширенияРаботыСФайлами ()
	ПоместитьФайл ()	НачатьПомещениеФайла ()
	КопироватьФайл ()	НачатьКопированиеФайла ()
	ПереместитьФайл ()	НачатьПеремещениеФайла ()
	НайтиФайлы ()	НачатьПоискФайлов ()
	СоздатьКаталог ()	НачатьСозданиеКаталога ()
	КаталогВременныхФайлов ()	НачатьПолучениеКаталогаВременныхФайлов ()
	КаталогДокументов ()	НачатьПолучениеКаталогаДокументов ()
РабочийКаталогДанныхПользователя ()	НачатьПолучениеРабочегоКаталогаДанныхПользователя ()	
ПолучитьФайлы ()	НачатьПолучениеФайлов ()	
ПоместитьФайлы ()	НачатьПомещениеФайлов ()	
ЗапроситьРазрешениеПользователя ()	НачатьЗапросРазрешенияПользователя ()	
ЗапуститьПриложение ()	НачатьЗапускПриложения ()	
ДиалогРедактированияСтандартногоПериода	Редактировать ()	Показать ()
КонструкторФорматнойСтроки	ОткрытьМодально ()	Показать ()
ОбработкаРасшифровкиКомпоновкиДанных	ВыбратьДействие ()	ПоказатьВыборДействия ()
ДиалогВыбораШрифта	Выбрать ()	Показать ()
ДиалогВыбораЦвета	Выбрать ()	Показать ()
ДиалогРасписанияРегламентногоЗадания	ОткрытьМодально ()	Показать ()
ДиалогВыбораФайла	Выбрать ()	Показать ()
СписокЗначений	ОтметитьЭлементы ()	ПоказатьОтметкуЭлементов ()
	ВыбратьЭлемент ()	ПоказатьВыборЭлемента ()
УправляемаяФорма	ОткрытьМодально ()	Открыть ()
	ВыбратьИзМеню ()	ПоказатьВыборИзМеню ()
	ВыбратьИзСписка ()	ПоказатьВыборИзСписка ()
МенеджерКриптографии	Конструктор	НачатьИнициализацию ()
	Зашифровать ()	НачатьШифрование ()
	Расшифровать ()	НачатьРасшифровку ()
	Подписать ()	НачатьПодписывание ()
	ПроверитьПодпись ()	НачатьПроверкуПодписи ()
	ПолучитьИнформациюМодуляКриптографии ()	НачатьПолучениеИнформацииМодуляКриптографии ()
	ПолучитьСертификатыИзПодписи ()	НачатьПолучениеСертификатовИзПодписи ()
	ПолучитьХранилищеСертификатов ()	НачатьПолучениеХранилищаСертификатов ()
ПроверитьСертификат ()	НачатьПроверкуСертификата ()	
СертификатКриптографии	Конструктор	НачатьИнициализацию ()
	Выгрузка ()	НачатьВыгрузку ()
СредстваКриптографии	ПолучитьИнформациюМодуляКриптографии ()	НачатьПолучениеИнформацииМодуляКриптографии ()
ХранилищеСертификатовКриптографии	Добавить ()	НачатьДобавление ()
	НайтиПоОтпечатку ()	НачатьПоискПоОтпечатку ()
	НайтиПоСерийномуНомеру ()	НачатьПоискПоСерийномуНомеру ()
	НайтиПоСубъекту ()	НачатьПоискПоСубъекту ()
	ПолучитьВсе ()	НачатьПолучениеВсех ()
	Удалить ()	НачатьУдаление ()

Файл	Конструктор	НачатьИнициализацию ()
	ПолучитьВремяИзменения ()	НачатьПолучениеВремениИзменения ()
	ПолучитьНевидимость ()	НачатьПолучениеНевидимости ()
	ПолучитьТолькоЧтение ()	НачатьПолучениеТолькоЧтения ()
	ПолучитьУниверсальноеВремяИзменения ()	НачатьПолучениеУниверсальногоВремениИзменения ()
	Размер ()	НачатьПолучениеРазмера ()
	Существует ()	НачатьПроверкуСуществования ()
	УстановитьВремяИзменения ()	НачатьУстановкуВремениИзменения ()
	УстановитьНевидимость ()	НачатьУстановкуНевидимости ()
	УстановитьТолькоЧтение ()	НачатьУстановкуТолькоЧтения ()
	УстановитьУниверсальноеВремяИзменения ()	НачатьУстановкуУниверсальногоВремениИзменения ()
	ЭтоКаталог ()	НачатьПроверкуЭтоКаталог ()
	ЭтоФайл ()	НачатьПроверкуЭтоФайл ()

Также при переработке модулей будет полезен механизм рефакторинга, в котором предусмотрены специальные возможности для облегчения переработки (см. [здесь](#)).