

ОСОБЕННОСТИ ПЛАТФОРМЫ "1С:ПРЕДПРИЯТИЕ 8.2"

В платформе "1С:Предприятие 8.2" есть ряд неочевидных особенностей, о которых лучше знать заранее, чем внезапно натолкнуться в реальной практике. Обратите внимание, речь пойдет не об ошибках платформы, а именно о "фичах", о неоднозначном поведении, о тех "граблях", на которые часто наступают технические специалисты.

Все описанные особенности воспроизводятся на релизе 1С:Предприятие 8.2 (8.2.13.202). В будущих версиях возможно изменение поведения системы.

ОСОБЕННОСТЬ №1. ОЧИСТКА ЗНАЧЕНИЙ В КОЛОНКАХ ТАБЛИЦЫ ФОРМЫ

Рассмотрим решение задачи. Требуется для каждой строки документа вывести определенный статус, который целиком зависит от реквизитов документа. Например, если Количество > 100 единиц или Сумма > 10000 руб., то статус – "Опт", иначе – "Розница".

Для решения можно создать дополнительный реквизит табличной части и заполнять его при изменении цены и количества. Однако зачем хранить эти данные? Дополнительные реквизиты – определенная нагрузка на систему, например, более длительное время записи документа. Тем более "Статус" целиком зависит от реквизитов "Сумма" и "Количество", а алгоритм вычисления достаточно простой. В платформе существует возможность добавить дополнительную колонку для реквизита формы (см. рис. 1). Далее она может быть размещена на форме.

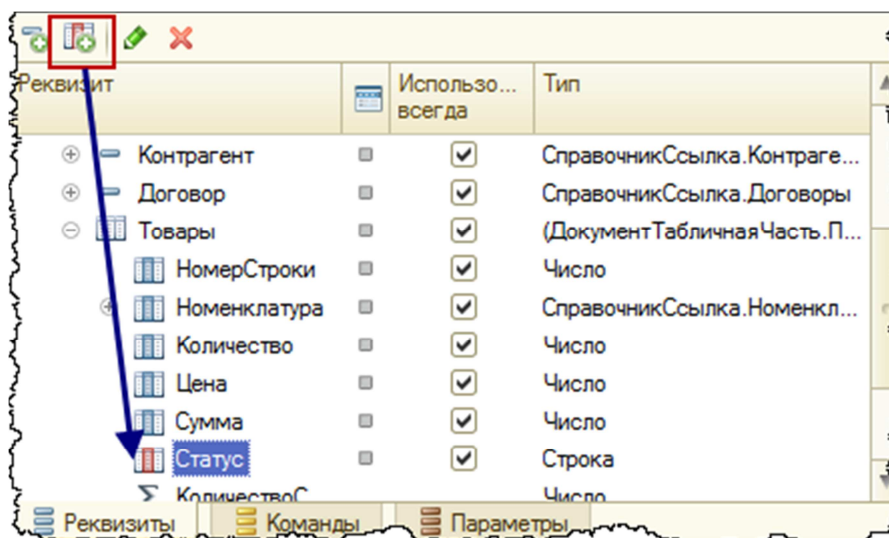


Рис. 1. Добавление колонки реквизита формы

Осталось заполнить значение колонки. Это нужно сделать программным образом, ведь колонка реквизита существует пока открыта форма и ее значения не сохраняются в базе данных. Таким образом, заполнение нужно выполнять как при открытии формы (а если быть более точным, при заполнении данными реквизитов формы), так и при изменении связанных данных (в нашем случае это поля "Количество" и "Сумма").

Алгоритм заполнения является тривиальным, и в комментариях не нуждается (см. рис. 2).

```

&НаСервере
Процедура ПриЧтенииНаСервере (ТекущийОбъект)
    Для каждого Строка Из Объект.Товары Цикл
        Если Строка.Количество>100 ИЛИ Строка.Сумма>10000 Тогда
            Строка.Статус = "Опт";
        Иначе
            Строка.Статус = "Розница";
        КонечЕсли;
    КонечЦикла;
КонечПроцедуры

```

Рис. 2. Алгоритм заполнения колонки реквизита

Важно правильно выбрать обработчик события. Обработчик "ПриЧтенииНаСервере" вызывается при любом заполнении данных в реквизиты формы. Такое заполнение происходит при открытии формы существующего документа и при вызове команды "Перечитать" из меню "Все действия" документа (см. рис 3).

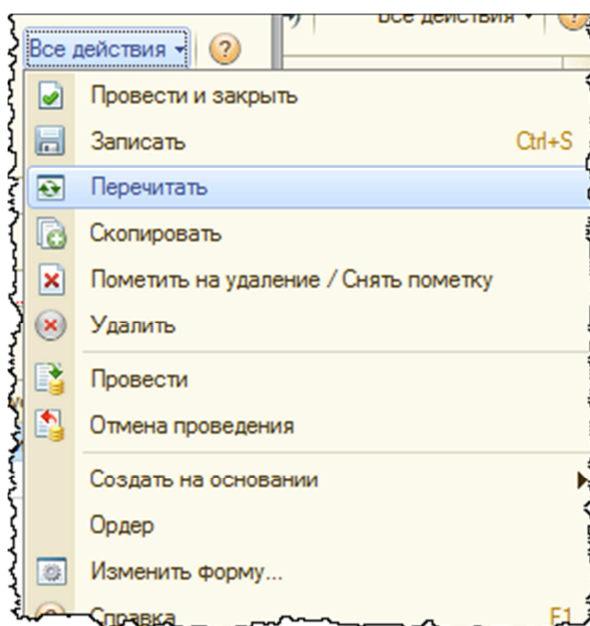


Рис. 3. Меню "Все действия" документа

Приведенный алгоритм прекрасно работает, но начиная с релиза 8.2.13, появилась одна особенность. При записи объекта (включая проведение документа) происходит очистка реквизитов, не связанных с данными (в нашем случае "Статус"). Строго говоря, очистка происходит не только при записи, но и при любом преобразовании объекта в данные формы. Например, при использовании методов "ЗначениеВРеквизитФормы()", "ДанныеВРеквизитФормы()".

Поэтому для того, чтобы приведенное решение было работоспособным, необходимо выполнять повторное заполнение колонок после записи и после любого его преобразования из объекта. Например, для этого можно использовать обработчик события формы "ПослеЗаписиНаСервере()".

ОСОБЕННОСТЬ №2. ИСПОЛЬЗОВАНИЕ ПАРАМЕТРИЗОВАННЫХ КОМАНД

В платформе 8.2 появился замечательный объект конфигурации "Команда". Мы не будем останавливаться на свойствах команд, опишем лишь одну особенность.

Создаем параметризованную команду, тип параметра – ссылка на соответствующий документ (см. рис. 4). Расположение команды указывается в реквизите "Группа", задаем значение "Командная панель формы".

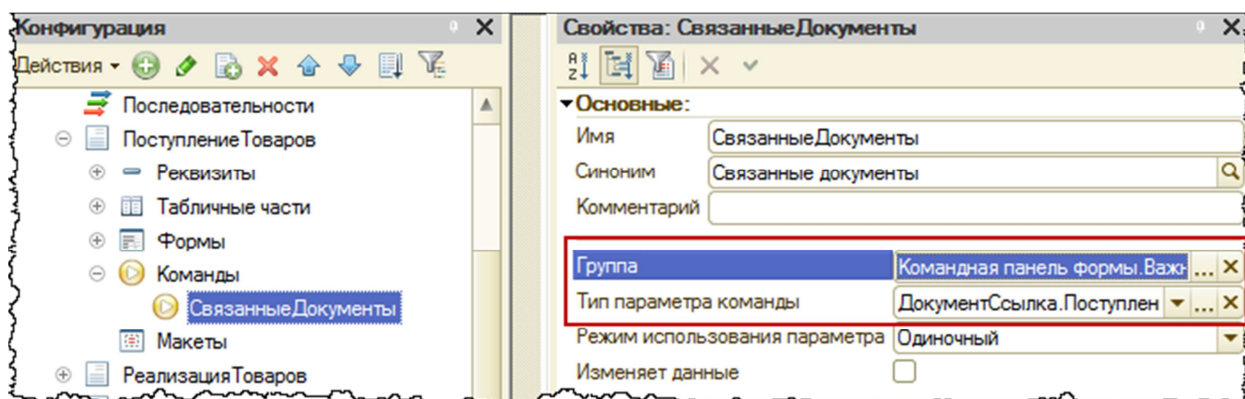


Рис. 4. Настройка параметризованной команды документа

Особенность таких команд заключается в том, что они выполняют неявную запись объектов. Для документов ситуация обостряется, тем что они перепроводятся. Поскольку запись проведенного документа выполняет его перепроведение. Таким образом, возможен следующий сценарий:

- Пользователь открыл документ, случайно в нем удалил строку. Но это не страшно, ведь пользователь знает, что при закрытии документа система выполнит запрос о сохранении данных, и он сможет отказаться от записи изменений;
- Пользователь решает посмотреть связанные документы с текущим, нажав на соответствующую кнопку (по сути, выполнив команду);
- Платформа выполняет запись объекта. Причем, если документ был новый, то происходит запрос на запись. А если ранее был записан, то это действие выполняется безусловно, без предупреждений;
- Итог очевиден. Нежелательные изменения будут внесены в базу данных.

Следует отметить, что такой особенностью обладают именно команды, расположенные в командной панели формы. Если указать расположение "Панель навигации формы", то неявной записи не будет происходить.

ОСОБЕННОСТЬ №3. ПОВТОРНЫЙ ВЫЗОВ КОМАНДЫ ПАНЕЛИ НАВИГАЦИИ ИЗ ОКНА ПРИЛОЖЕНИЯ

Речь опять пойдет о командах. Но сейчас рассмотрим команды не формы, а главного окна приложения. Настройки такой команды показаны на рис. 5. В модуле команды открывается форма отчета (см. рис 6).

Такая команда действительно будет работать, но есть одна проблема: если открыть отчет, далее закрыть его, а после обратиться к нему еще раз, форма не откроется! То есть повторное исполнение команды возможно, только если вначале выполнить любую другую команду.

Объяснение такой особенности очень простое. Платформа предполагает, что формы, открываемые из панели навигации, будут располагаться в рабочей области, то есть не будут открываться в независимых окнах. Соответственно такие формы нельзя закрыть, но можно выполнить другую команду, которая откроет новую форму, взамен прежней. Поэтому любые действия, которые выполняют открытие независимых окон нужно располагать не в панели навигации, а в панели действий командного интерфейса.

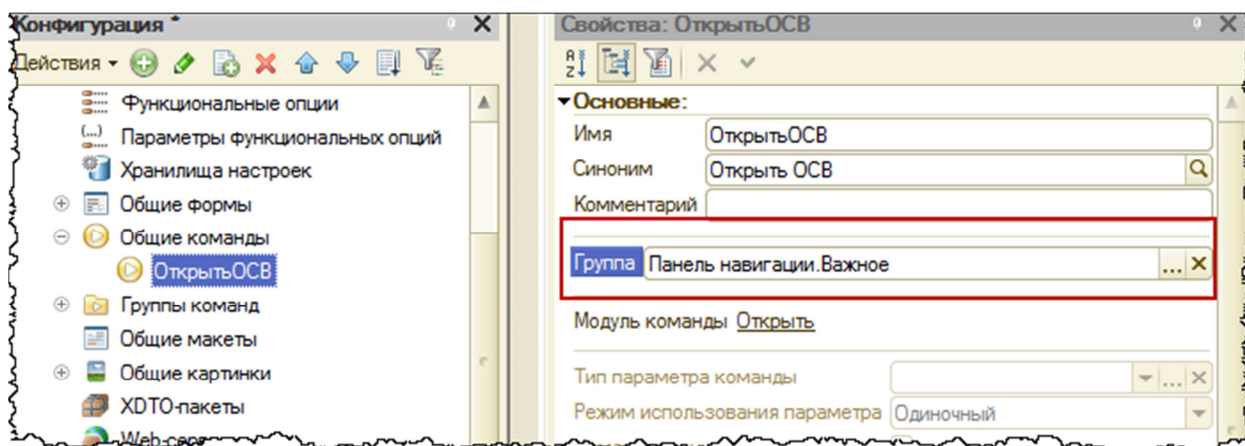


Рис. 5. Настройка общей команды

```

&НаКлиенте
├─ Процедура ОбработкаКоманды(ПараметрКоманды, ПараметрыВыполненияКоманды)
│   └─ ОткрытьФорму ("Отчет.ОСВ.форма");
└─ КонецПроцедуры
  
```

Рис. 6. Модуль общей команды

ОСОБЕННОСТЬ №4. ИЕРАРХИЧЕСКИЙ ПРОСМОТР СПРАВОЧНИКА

В конфигурации создано 2 справочника: "Контрагенты" и "Контактные лица". Контактные лица подчинены контрагентам и являются иерархическим справочником.

Если обратиться к справочнику в пользовательском режиме, то можно увидеть, что группы отображаются, но просмотра по иерархии нет (см. рис. 7). Все элементы выведены линейным списком. Более того, если попытаться включить иерархический просмотр, то можно убедиться, что режимы просмотра недоступны (см. рис. 8).

Контактные лица		
Наименование	Код	Владелец
Администрация	000000001	Белявский
Архип	000000006	Белявский
Директор	000000003	Белявский
Зам. директора. Кротов	000000004	Белявский
Менеджеры	000000002	Белявский
Руслан	000000005	Белявский

Рис. 7. Просмотр иерархического подчиненного справочника

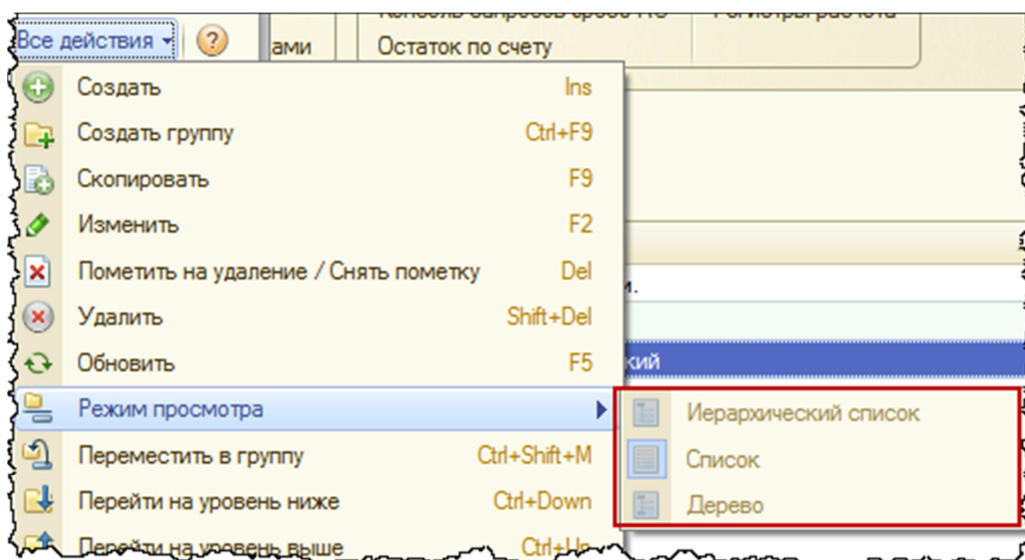


Рис. 8. Недоступность изменения режима просмотра

Такое поведение является очередной особенностью платформы. Дело в том, что иерархический просмотр в подчиненном справочнике возможен только при включенном отборе по владельцу (в настройке динамического списка). Этот отбор можно включать как интерактивно, так и программно.

Обратите внимание, что при открытии списка контактных лиц из формы контрагента отбор по владельцу установлен, и режим просмотра справочника – иерархический (см. рис. 9).

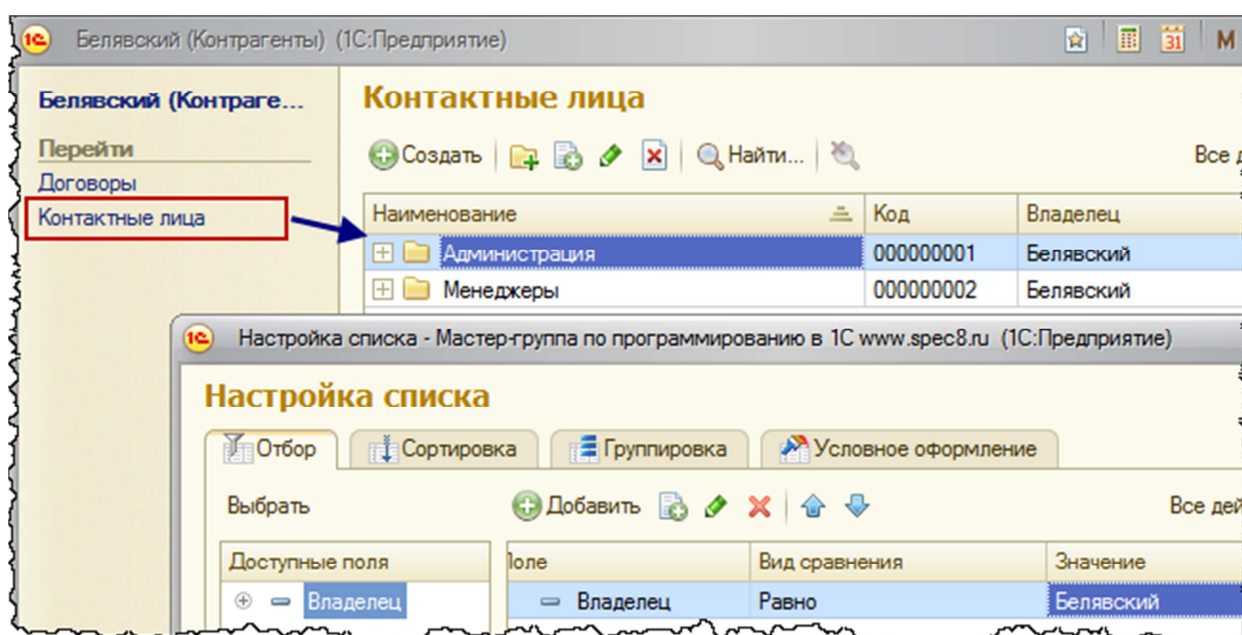


Рис. 9. Просмотр подчиненного справочника из формы владельца

ОСОБЕННОСТЬ №5. КОНТЕКСТ ИСПОЛНЕНИЯ КОДА. КЛИЕНТ И СЕРВЕР

Программный код "1С:Предприятие 8.2" может исполняться на клиентской машине, на одном из серверов (разумеется, это возможно только в клиент-серверном варианте работы платформы), на компьютере, откуда произведено внешнее подключение к ИБ (например, используя технологию Com-connection).

Для всех свойств и методов платформы определена доступность.

- Некоторые методы могут вызываться только на клиенте. В основном это методы работы с интерфейсами, открытие форм, организация диалога с пользователем;
- Другие методы могут выполняться исключительно на стороне сервера. Например, это работа с базой данных;
- Существует ряд методов доступных как на стороне клиента, так и на сервере. Где конкретно они будут исполняться зависит от места компиляции программного кода.

Именно о последних мы поговорим. Есть ли разница где выполнять программный код на клиенте или сервере? Разумеется.

Во-первых, сервер априори более мощная машина, чем клиент. Поэтому все длительные и ресурсоемкие операции лучше выполнять на стороне сервера. Но следует учитывать, что есть определенное время, которое будет затрачено на обращение к серверу, и на возврат результатов на клиента.

Во-вторых, результат некоторых операций зависит от контекста. Приведем два примера.

1. Работа с файлами, например, обращение к каталогу "C:\Temp", в зависимости от места выполнения кода будет обращаться либо, к каталогу текущего компьютера, либо к данным на сервере.
2. Получение даты и времени. На клиенте получается локальная дата, на сервере – дата одного из серверов, входящих в кластер, который может располагаться, например, в другом полушарии. Разумеется, время (а иногда и дата) будет получено разное.

Также следует иметь в виду, что в 8.2 в кластере серверов может быть несколько машин, и к какой из них будет выполняться подключение заранее не известно. Более того, в процессе работы соединение может быть "переброшено" с одного рабочего сервера на другой.

ОСОБЕННОСТЬ №6. НАСТРОЙКИ ВНЕШНИХ ОБРАБОТОК И ОТЧЕТОВ

При создании внешних обработок удобно использовать сохранение настроек (в том числе автоматическое сохранение). Применение этого механизма включается в свойствах формы (см. рис. 10). Также для реквизитов нужно указать возможность сохранения в настройках (см. рис. 11).

Отчеты, разработанные на системе компоновки данных, по умолчанию могут сохранять настройки вариантов.

Все настройки хранятся в информационной базе. При этом сохранение происходит в привязке к идентификатору формы (либо схемы компоновки для отчетов). В этом и заключается проблема. Дело в том, что при копировании внешних ответов и обработок идентификаторы также копируются. Это логично. Но возможен следующий сценарий.

1. Создана некоторая внешняя обработка по загрузке данных в справочник "Номенклатура". На форме располагаются несколько реквизитов с возможностью сохранения. Пользователь сохраняет настройки.
2. Копированием создается еще одна обработка, которая предназначена для загрузки контрагентов. Обработка незначительно изменяется, пользователь вновь сохраняет настройки реквизитов.
3. Получилось две формы с одинаковыми идентификаторами. И сохранение настроек в одной обработке приводит к их замещению в другой.

Выход из данной коллизии следующий. Необходимо обновить идентификатор формы. Для этого форму обработки можно скопировать, а старую удалить. Идентификатор у формы будет изменен.

Для отчетов необходимо выгрузить схему компоновки данных в xml, и выполнить обратную загрузку. Идентификатор схемы обновится.

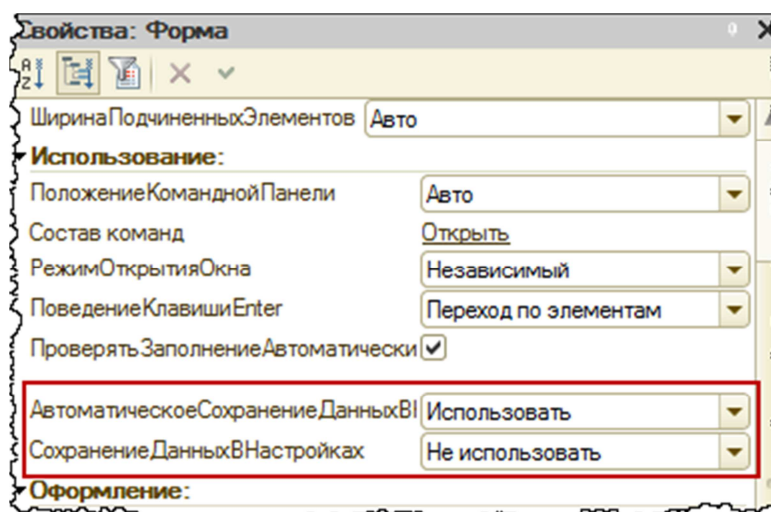


Рис. 10. Настройка механизма сохранения данных

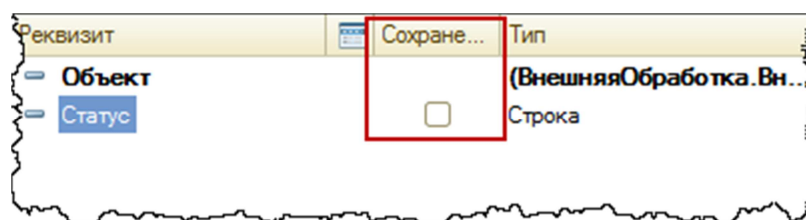


Рис. 11. Указание необходимости сохранения значений реквизитов в настройках

ОСОБЕННОСТЬ №7. НЕУНИКАЛЬНЫЕ КЛЮЧЕВЫЕ ПОЛЯ В ДИНАМИЧЕСКОМ СПИСКЕ

В платформе 8.2 появился замечательный объект – "Динамический список". С помощью него можно описывать списки информации, которые будут автоматически обновляться, например, при работе нескольких пользователей.

Одной из удобных возможностей динамического списка является возможность указания произвольного запроса для выборки данных (см. рис. 12). Также отметим свойство "Основная таблица", если эта таблица задана, то в списке могут выполняться действия с данными:

- Добавление;
- Изменение;
- Удаление, пометка на удаление;
- Ввод нового;
- И другие.

Если для получения данных динамического списка используется запрос и указана основная таблица, то необходимо обеспечить, чтобы ключевые поля таблицы встречались в выборке один раз. Пример запроса, где указанное правило может не соблюдаться приведен на рис. 13. Если какой-нибудь документ сделал более одной записи по регистру "Продажи", то этот документ в результате запроса будет встречаться неоднократно. При этом ключевым полем для ссылочных таблиц является "Ссылка", для регистров ключом может быть комбинация полей "Регистратор" и "Номер строки".

В случае если ключевое поле встречается неоднократно, динамический список начинает "колбасить". Например, при листании списка на глазах увеличивается количество элементов. Возникают и другие забавные эффекты.

Рис. 12. Свойства динамического списка

ВЫБРАТЬ

ДокТовары.Ссылка,
 ДокТовары.ПометкаУдаления,
 ДокТовары.Номер,
 ДокТовары.Дата,
 ДокТовары.Проведен,
 ДокТовары.Контрагент,
 ДокТовары.Договор,
 Продажи.Номенклатура

ИЗ

РегистрНакопления.Продажи КАК Продажи
 ЛЕВОЕ СОЕДИНЕНИЕ Документ.РеализацияТоваров КАК ДокТовары
 ПО Продажи.Регистратор = ДокТовары.Ссылка

Рис. 13. Запрос, возвращающий неуникальные ключевые поля

ОСОБЕННОСТЬ №8. ИСПОЛЬЗОВАНИЕ ОБЪЕДИНЕНИЙ В ДИНАМИЧЕСКОМ СПИСКЕ

В произвольном запросе динамического списка можно оказать объединение запросов к разным таблицам. Разумеется, в этом случае нельзя будет указать основную таблицу. Но есть и еще одна особенность.

Сортировка по полям динамического списка будет невозможна.

Например, на рис. 14 представлен запрос, который может служить источником данных для динамического списка. При попытке выполнить сортировку по полю "Ссылка" в пользовательском режиме возникает ошибка (см. рис. 15).

Решается эта проблема следующим образом: необходимо "обернуть" объединение запросов во вложенный запрос, пример приведен на рис. 16. Тогда сортировка будет выполняться корректно.


```

ВЫБРАТЬ
    Контрагенты.Ссылка
ИЗ
    Справочник.Контрагенты КАК Контрагенты

ОБЪЕДИНИТЬ ВСЕ

ВЫБРАТЬ
    Номенклатура.Ссылка
ИЗ
    Справочник.Номенклатура КАК Номенклатура

```

Рис. 14. Объединение запросов для динамического списка

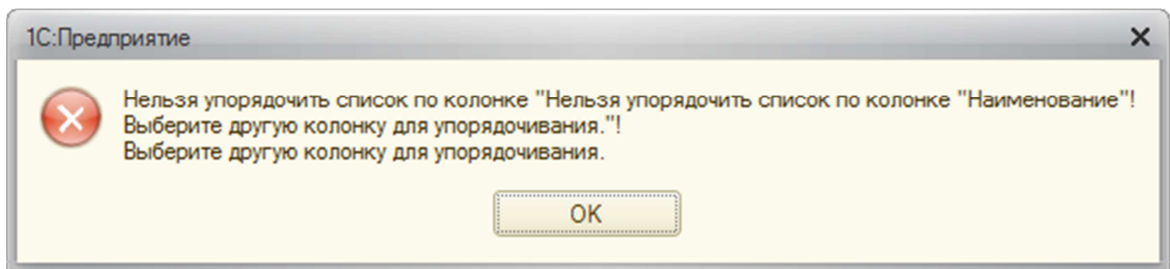


Рис. 15. Ошибка сортировки в динамическом списке

```

ВЫБРАТЬ
    Данные.Ссылка
ИЗ
    (ВЫБРАТЬ
        Контрагенты.Ссылка КАК Ссылка
    ИЗ
        Справочник.Контрагенты КАК Контрагенты

    ОБЪЕДИНИТЬ ВСЕ

    ВЫБРАТЬ
        Номенклатура.Ссылка
    ИЗ
        Справочник.Номенклатура КАК Номенклатура) КАК Данные

```

Рис. 16. Описание вложенного запроса для динамического списка

ОСОБЕННОСТЬ №9. ПЕРЕДАЧА ПАРАМЕТРОВ В ЗАПРОС ДИНАМИЧЕСКОГО СПИСКА

Как известно, в запрос можно в качестве параметров передавать не только конкретные значения, но и коллекции: массив, список значений, таблица значений. Данные коллекции можно использовать в условиях сравнения, например, "...ГДЕ ДокументПоступление.Контрагент В (&СписокКонтрагентов)...".

Однако если попытаться передать список значений в качестве параметра запроса динамического списка (см. рис. 17), то система возьмет только первый элемент этого списка! Это особенность платформы. Для решения проблемы необходимо использовать массив вместо списка значений.

```
&НаСервере
□ Процедура ПриСозданииНаСервере (Отказ, СтандартнаяОбработка)

    Список = Новый СписокЗначений;
    Список.Добавить (Справочники.Номенклатура.Ящик);
    Список.Добавить (Справочники.Номенклатура.Паллета);
    СписокТоваров.Параметры.УстановитьЗначениеПараметра ("СписокТоваров", Список);

КонецПроцедуры
```

Рис. 17. Установка параметров запроса динамического списка