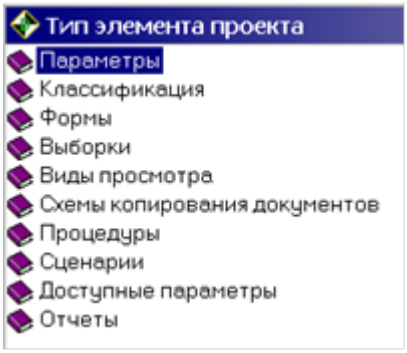
Документ

Документ - электронный эквивалент бумажного документа. Документы можно просматривать, изменять, анализировать, распечатывать, формировать на их основе различные отчеты. Для хранения документов применяется таблица '*Документ*'.

Внутри раздела '*Документ*' можно создать следующие разделы:



- **Параметры** – содержит описания параметров, переменных и выражений.
- **Классификация** – содержит описание классов и типов документов.
- **Формы** – содержит описания форм для ввода документа.
- **Выборки** – содержит описания простых выборок, применяемых для отбора документов.
- **Виды просмотра** – содержит описания видов просмотра, работающих от контекста документа.
- **Схемы копирования документов** - содержит описания схем создания одного документа на основе другого. Схемы применяются при описании методов и процедур.
- **Процедуры** – в данном разделе описываются процедуры, которые обрабатывают одну запись документа.
- **Сценарии** – содержит описания сценариев, которые предполагается использовать в качестве родителей при создании других сценариев.
- **Доступные параметры** – не используется.
- **Отчеты** – содержит описания отчетов, обрабатывающих один документ.

> Подробное описание перечисленных разделов приведено ранее, в главе '**Основные объекты**' и в главах по разделам ('**Параметры**', '**Процедуры**', '**Виды просмотра**', '**Выборки**', '**Формы**', '**Сценарии**', '**Отчеты**').

Рассмотрим описание документов на примере.

Пример описания документа

Предположим требуется реализовать ввод документов для прихода и реализации товаров. Для решения этой задачи в проекте необходимо найти или создать:

- Параметры документов
- Классы и типы для документов
- Содержимое документов – классы и типы строк
- Формы ввода документов
- Вид просмотра документов
- Сценарии для форм, видов просмотра и контекстных меню
- Отчеты по документам

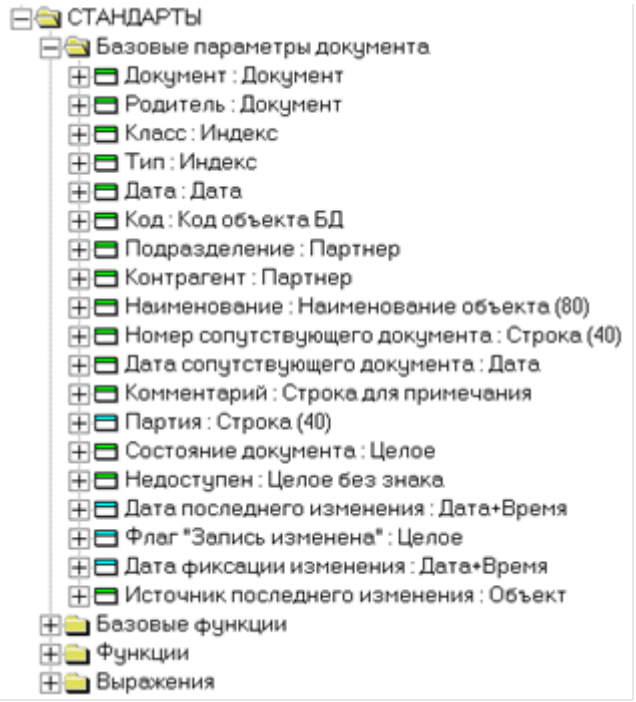
Рассмотрим возможный вариант действий автора проекта.

Прежде всего необходимо ознакомиться со списком параметров документа и найти те параметры, которые предполагается использовать. Результат этой работы пригодится при описании форм, видов просмотра и отчетов.

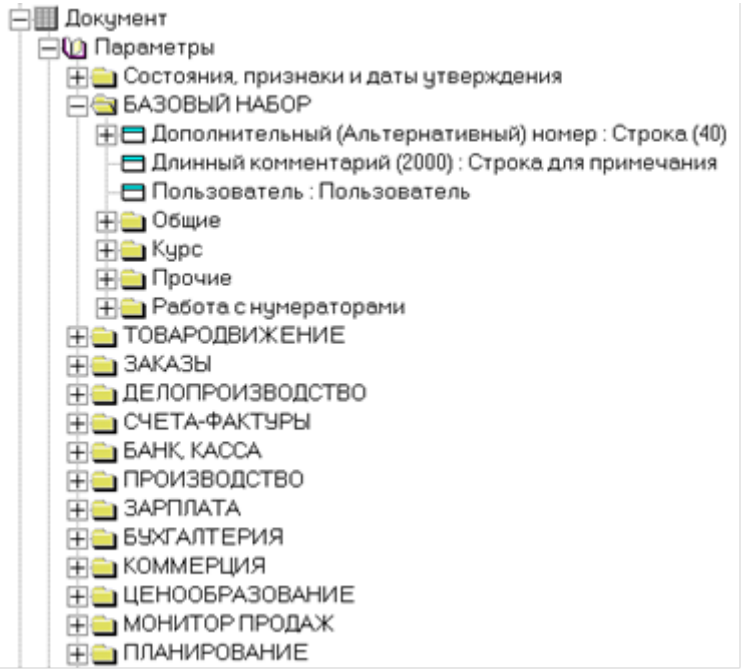
> Рассматриваемый пример будет продолжен в главах ‘**Строки**’ и ‘**Проводки**'.

Параметры документа

Описание базовых параметров документа находится в базовом разделе ‘*Системная область*’.



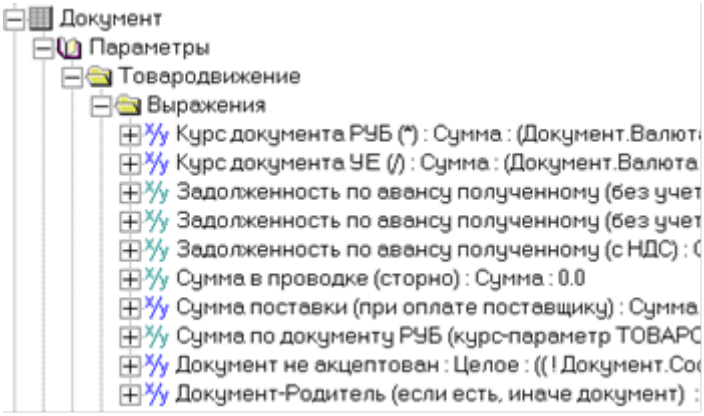
Остальные параметры расположены в библиотеке [*КЛАССЫ, ТИПЫ, ПАРАМЕТРЫ, ПЛАНЫ, ОПИСАНИЯ СПИСКОВ*].



При выборе параметра необходимо обращать внимание на тип данных.

На этом этапе можно провести еще одно исследование.

В проекте создано множество функций и выражений, работающих от текущей записи документа. Такие функции и выражения располагаются в разделе ‘*Параметры*’ раздела ‘*Документ*’.

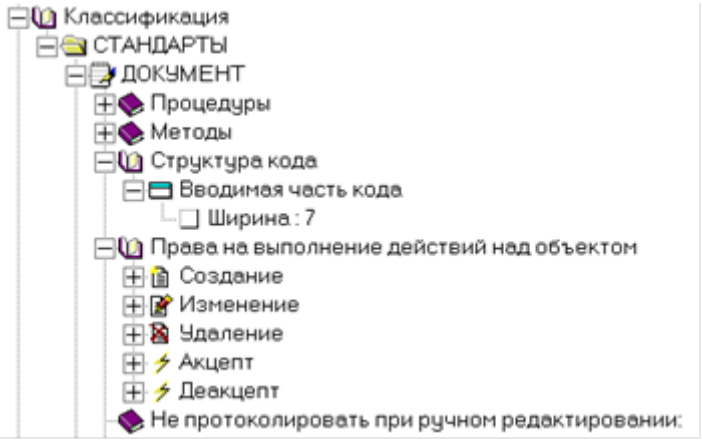


Некоторые из этих функций и выражений потребуются при описании формы, вида просмотра и отчетов.

Классификация документов

После ознакомления с возможными параметрами необходимо выбрать два типа документов. Первый тип понадобится для документов, которыми будет оформляться приход товаров. Второй тип будет использован для документов о реализации товаров.

Базовый класс документов описан в базовом разделе ‘*Системная область*’.

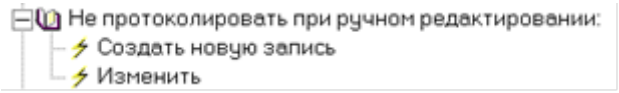


Для документов данного класса разрешены все действия с записью. Ширина параметра ‘Код’ записи равна 7 символов.

Подробнее о назначении разделов ‘*Процедуры*’ и ‘*Методы*’ написано в конце данной главы.

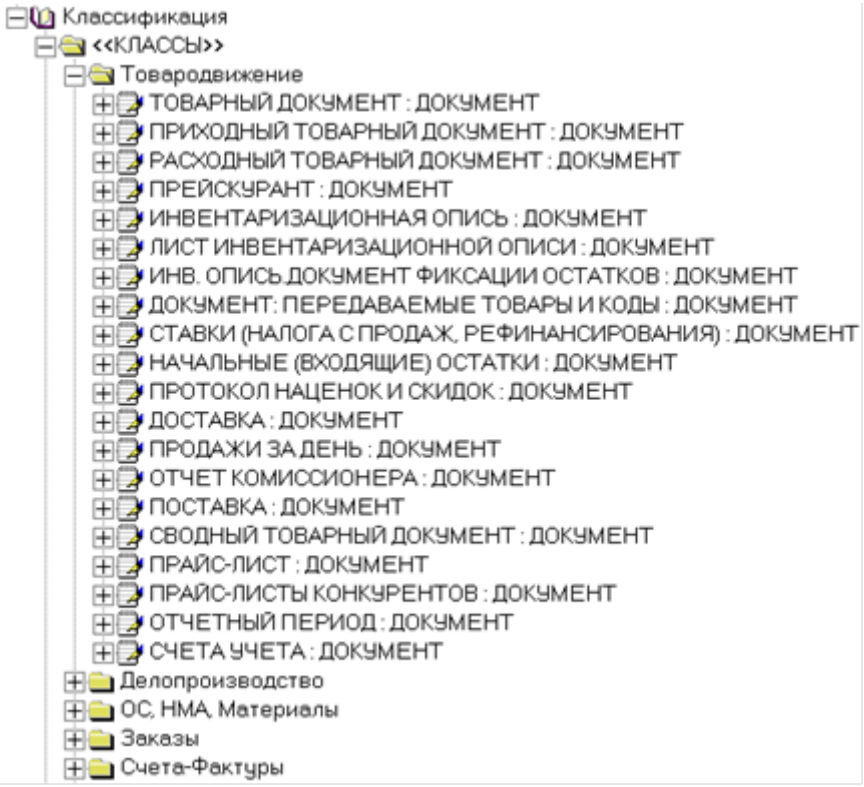
Значение атрибута ‘*Не протоколировать при ручном редактировании*’ не задано.

> Атрибут ‘*Не протоколировать при ручном редактировании*’ применяется для отмены автоматического формирования сообщений в протокол при использовании обычных форм ввода документа. Если данный атрибут не указан, то при создании или изменении документа (посредством формы ввода) в протоколе появляется соответствующее сообщение. В данном атрибуте можно указать одно или оба действия, при которых сообщение создавать не следует. Действия задаются с помощью команд ‘*Создать новую запись*’ и ‘*Изменить*’. Например.

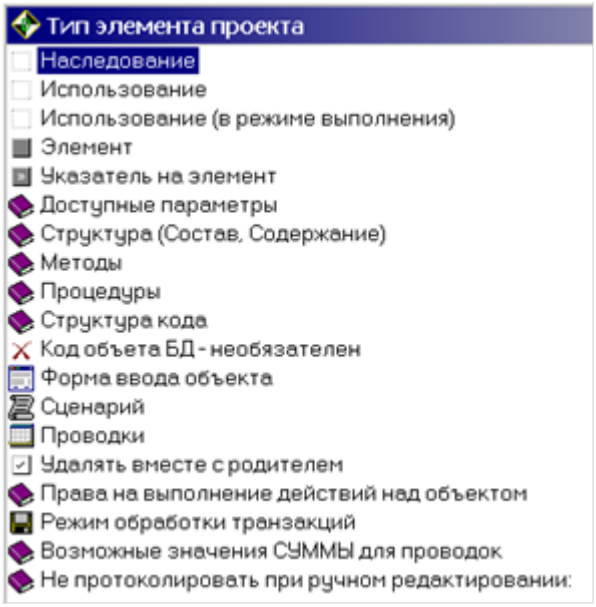


В библиотеке [*КЛАССЫ, ТИПЫ, ПАРАМЕТРЫ, ПЛАНЫ, ОПИСАНИЯ СПИСКОВ*] перечислены централизованно заведенные классы и типы документов.

Классы документа ссылаются на базовый класс ‘*ДОКУМЕНТ*’. Это позволяет не повторять для каждого класса описание атрибутов ‘*Методы*’, ‘*Права на выполнение действий над объектом*’, ‘*Структура кода*’ *и ‘Не протоколировать при ручном редактировании’*.

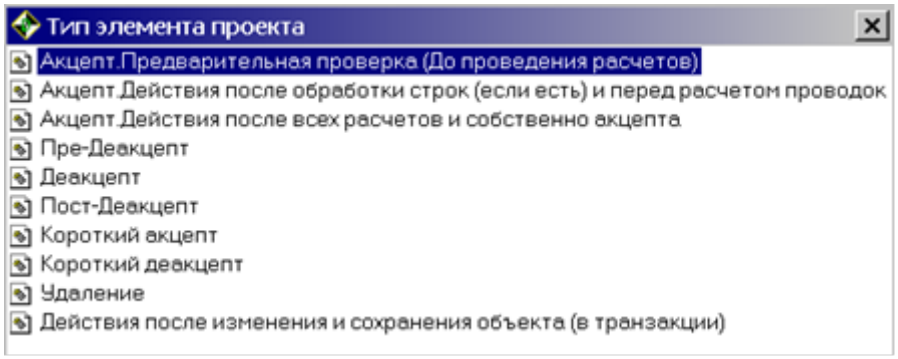


Для класса документа можно указать следующие параметры:



- **Наследование** – условие наследования от данного класса.
- **Использование** – условие доступа проектировщиков к классу.
- **Использование (в режиме выполнения)** – условие доступа пользователей к документам данного класса.
- **Доступные параметры** – не используется.
- **Структура (Состав, Содержание)** – содержит список классов и типов. Наследников документа данного класса можно создавать только таких типов, которые перечислены в указанном списке.
- **Методы** – содержит описания базовых алгоритмов обработки документа.

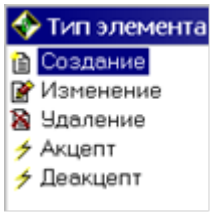
Можно указать следующие методы обработки:



- **Акцепт. Предварительная проверка (до проведения расчетов)** – действия, выполняемые на этапе проверки перед акцептом.
- **Акцепт. Действия после обработки строк (если есть) и перед расчетом проводок** – действия, выполняемые на этапе перед расчетом проводок.
- **Акцепт. Действия после всех расчетов и собственно акцепта** – действия, выполняемые на последнем этапе акцепта.
- **Пре-Деакцепт** – действия, выполняемые на этапе перед деакцептом.
- **Деакцепт** – действия, выполняемые на этапе деакцепта.
- **Пост-Деакцепт** – действия, выполняемые на этапе после деакцепта
- **Короткий акцепт** – действия, выполняемые в режиме короткого акцепта.
- **Короткий деакцепт** – действия, выполняемые в режиме короткого деакцепта.
- **Удаление** – действия, выполняемые при удалении документа.
- **Действия после изменения и сохранения объекта (в транзакции)** – действия, выполняемые после сохранения документа.

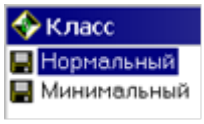
Методы создаются программистами и системными администраторами проекта. Авторы проектов имеют возможность указать в методах необходимые им процедуры.

- **Процедуры** – содержит список процедур для методов.
- **Структура кода** – задает правило формирования значения параметра ‘*Код*’ записи документа.
- **Код объекта БД – необязателен** – задает условие, при котором формирование кода документа не производится.
- **Форма ввода объекта** – задает форму ввода для документов данного класса.
- **Сценарий** – для описания контекстного меню документа, вызываемого при нажатии правой кнопки мыши.
- **Проводки** – для описания правил формирования проводок в процедуре акцепта документа. Проводки указываются отдельно для каждого плана счетов.
- **Удалять вместе с родителями** – при установленном признаке документ данного класса будет удален в момент удаления родителя объекта. Другими словами, объект данного класса должен присутствовать в БД только вместе со своим родителем.
- **Права на выполнение действий над объектом** – определяет условия выполнения основных действий с записью документа.



- **Создание** – список условий, при выполнении которых разрешается создать новую запись документа.
- **Изменение** – список условий, при выполнении которых разрешается изменить запись документа.
- **Удаление** – список условий, при выполнении которых разрешается удалить запись документа.
- **Акцепт** – список условий, при выполнении которых разрешается запустить процедуру акцепта документа.
- **Деакцепт** – список условий, при выполнении которых разрешается запустить процедуру деакцепта документа.

- **Режим обработки транзакций** – для указания момента открытия транзакции в процедуре акцепта документа.

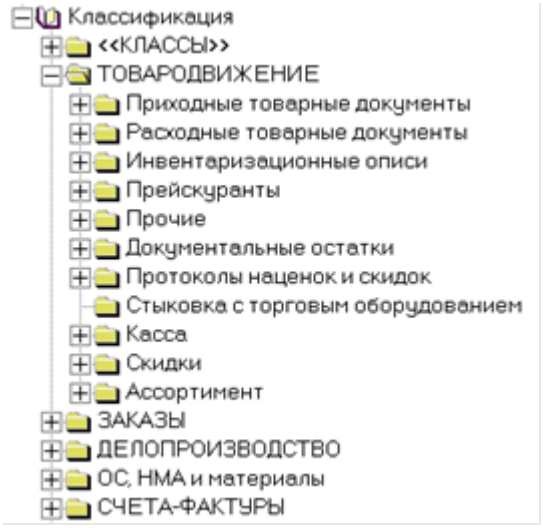


- **Нормальный** – единая транзакция для акцепта документа и акцепта строк документа.
- **Минимальный** – акцепт строк документа и акцепт записи документа выполняются в отдельных транзакциях.

> Подробнее об этапах акцепта документа написано в конце главы.

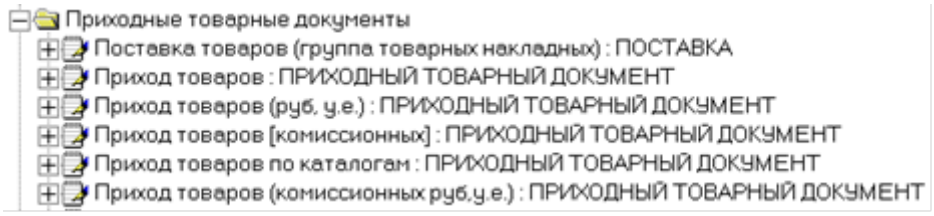
- **Возможные значения СУММЫ для проводок** – для ограничения списка сумм при описании правил формирования проводок.
- **Не протоколировать при ручном редактировании** – для перечисления стандартных действий, выполнение которых не должно сопровождаться записью соответствующего сообщения в протокол.

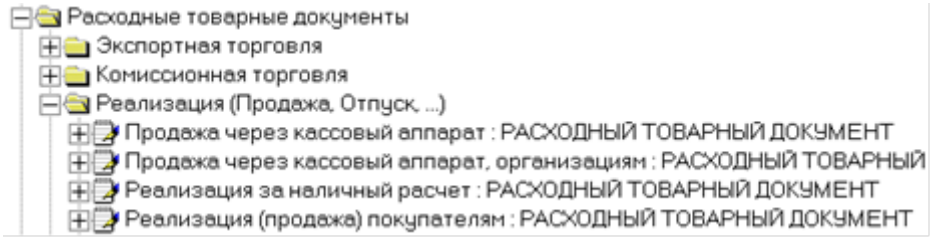
Типы документа созданы на основе классов.



В этой библиотеке никакие значимые атрибуты классов и типов не указываются. Атрибуты задаются в других библиотеках. Для этого применяют подстановки.

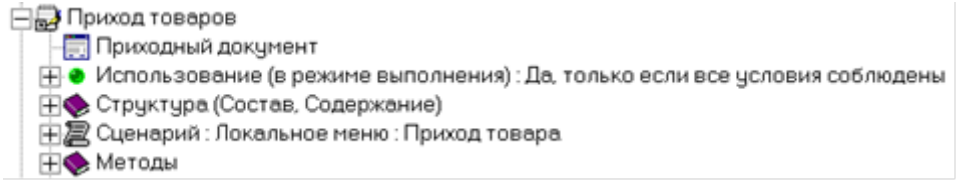
В поисках нужных нам типов документов раскроем две папки ‘*Приходные товарные документы*’ и ‘*Расходные товарные документы*’.



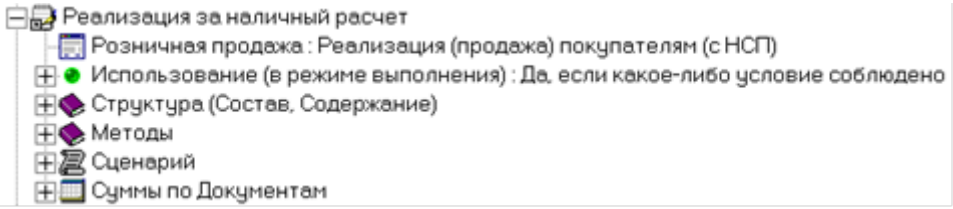


Для наших целей подходят два типа документов: ‘*Приход товаров*’ и ‘*Реализация за наличный расчет*’.

Посмотрим на описание атрибутов у этих типов.



Для данного типа документа заданы форма, условие использования, структура документа, сценарий для контекстного меню и методы.



В подстановке указана форма, условие использования, структура документа, сценарий для контекстного меню, методы и правила формирования проводок по плану счетов ‘*Суммы по Документам*’.

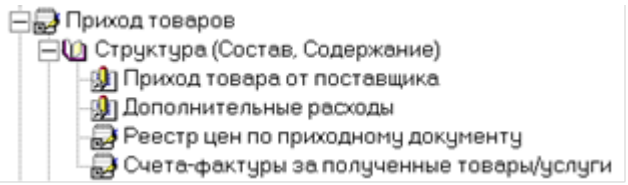
В проекте имеются и другие подстановки рассматриваемых типов. В этих подстановках заданы правила формирования проводок по другим планам счетам.

Подробнее о настройке проводок написано в главе ‘**Проводки**’.

> Необходимо всегда проверять наличие других подстановок, поскольку там могут быть изменены значения атрибутов.

Структура документа

В подстановках типов документов ‘*Приход товаров*’ и ‘*Реализация за наличный расчет*’ указан атрибут ‘*Структура (Состав, Содержание)*’. Этот атрибут задает возможные типы наследников документа (т.е. типы строк документа и типы документов, для которых данный документ будет являться родителем).

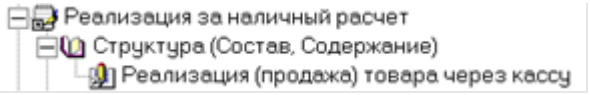


Для рассматриваемой задачи важно, что в списке имеется тип строки ‘*Приход товаров от поставщика*’. Это означает, что в документе можно будет создавать строки такого типа.

Ссылки на другие типы (строки и документов) для нашей задачи значения не имеют. Они означают, что в документе также можно заводить строки типа ‘*Дополнительные расходы*’ и формировать документы-наследники с типами ‘*Реестр цен по приходному документу*’ ‘*Счет-фактуры за полученные товары/услуги*’.

Отменить структуру можно двумя способами. Либо создать еще одну подстановку типа документа и не указывать перечисленные типы в атрибуте ‘*Структура (Состав, Содержание)*’. Либо можно запретить использование упомянутых в структуре типов с помощью подстановок этих типов.

Теперь посмотрим на второй тип документов.

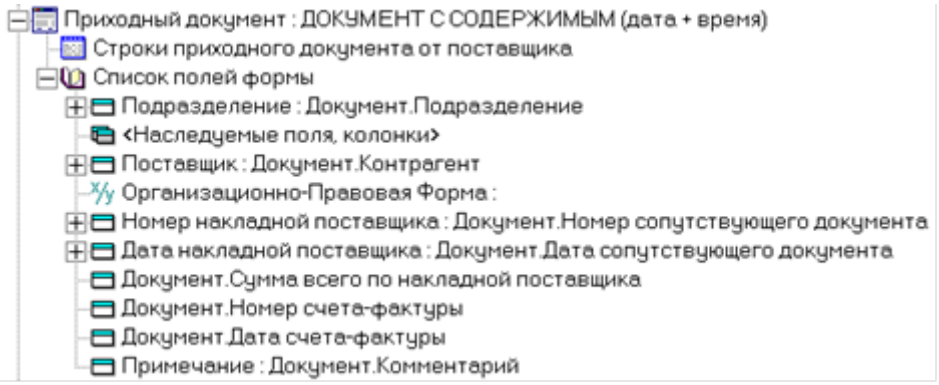


Описание структуры содержит ссылку на тип строки, что нам и требуется.

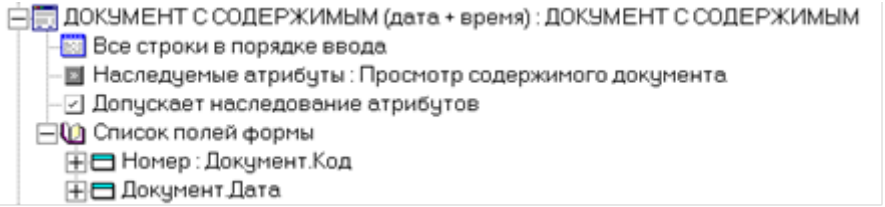
Форма документа

В описаниях рассматриваемых типов имеются ссылки на формы ввода. Рассмотрим формы подробнее.

Форма, заданная в подстановке типа '*Приход товаров*', находится в библиотеке '*# Товародвижение',* раздел '*Документ*', раздел '*Форма*', папка '*Приходные товарные документы*'.

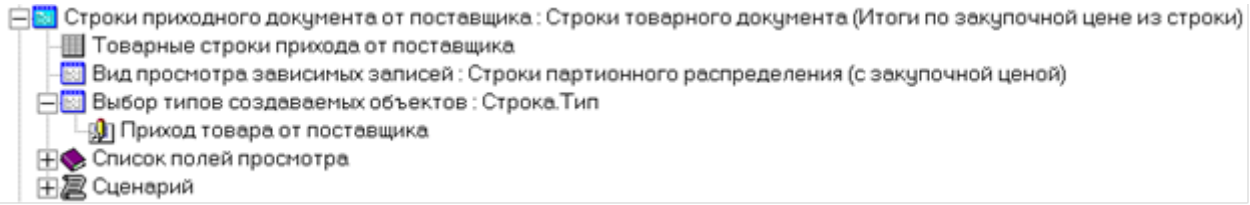


Описание атрибута '*Список полей формы*' включает содержимое соответствующего атрибута родителя формы. Родителем данной формы является форма '*ДОКУМЕНТ С СОДЕРЖИМЫМ (дата + время)*'. Рассмотрим эту форму.



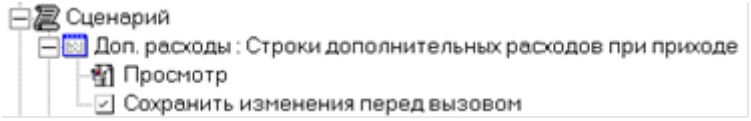
Список полей из этой формы по правилу наследования добавляются в список полей формы '*Приходный док*умент'.

Для формы '*Приходный док*умент' указан атрибут '*Просмотр содержимого документа*'. Значение атрибута ссылается на вид просмотра строк '*Строки приходного документа от поставщика*'.



Этот вид просмотра будет вызван по команде ‘*Просмотр содержимого’*. Оформляется вызов команды в сценарии. Например, вот так.

> *Ввод строк необязательно организовывать с помощью атрибута ‘Просмотр содержимого документа’. Можно из сценария формы вызвать непосредственно вид просмотра строк. Например.*



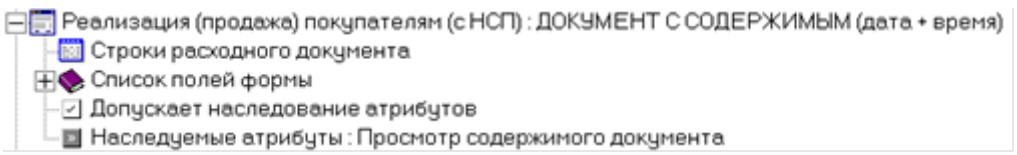
> *Для пункта сценария ‘Доп. расходы’ обязательно следует указать атрибут ‘Сохранить изменения перед вызовом’.*
>
> Наличие этого атрибута означает, что перед выполнением указанных действий вся введенная в форму информация будет записана в БД. В данном случае перед вызовом вида просмотра строк введенные в форму документа данные будут сохранены. Это позволит пользователю сразу перейти к вводу строк после заполнения полей формы. В противном случае, сначала необходимо сохранить введенную в форму информацию (по кнопке ‘Сохранить’), затем повторно вызвать форму и выбрать пункт сценария для ввода строк.

Применение атрибута ‘*Вид просмотра зависимых записей*’ удобно тем, что для описания нескольких форм документов используется один сценарий. Такое возможно, поскольку в сценарии не конкретизируется вид просмотра строк.

Если список полей формы не соответствует требованиям задачи, то его следует изменить. Это можно сделать двумя способами:

- Можно создать подстановку данной формы и изменить значение атрибута ‘*Список полей формы’*.
- Можно создать новую форму и новую подстановку типа документа. В подстановке потребуется указать ссылку на созданную форму.

Теперь перейдем к форме для ввода документов типа ‘*Реализация за наличный расчет*’. Форма находится в библиотеке *‘# Товародвижение’,* раздел ‘*Документ*’, раздел ‘*Форма*’, папка ‘Рас*ходные товарные документы*’.



Поскольку список полей формы содержит более не используемый параметр НСП, то

придется создать новую подстановку формы с подходящим списком полей.

Для отображения и ввода строк документа указан вид просмотра ‘*Строки расходного документа’*.

Подробнее про упомянутые виды просмотра строк ‘*Строки приходного документа от поставщика*’ и ‘*Строки расходного документа’* написано в следующей главе ‘**Строки**’.

> Необходимо проверить наличие подстановок для изученных форм и последующие действия выполнять с учетом изменения значений атрибутов в этих подстановках.

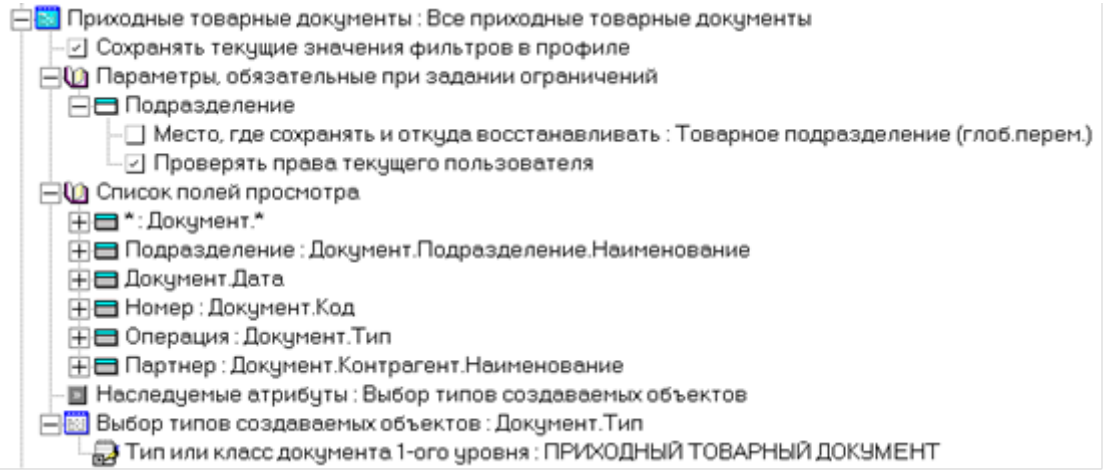
Вид просмотра документов

Для решаемой задачи требуется отобразить два списка: список документов прихода и список документов реализации.

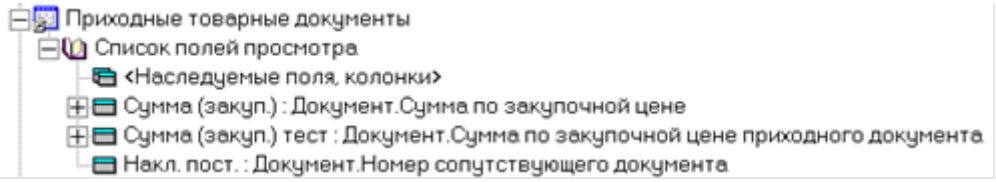
Такие виды просмотра могут быть расположены в разделе ‘*Виды просмотра*’ на верхнем уровне библиотеки. Попробуем найти подходящие.

Вид просмотра ‘Приходные товарные документы’

В библиотеке *‘! Базовый набор’*, раздел ‘*Виды просмотра*’, папка *‘! Базовый набор’*, папка *‘Документ*’, папка *‘Приходные документы’* находится вид просмотра ‘*Приходные товарные документы*’.



Перед изучением назначенных для этого вида просмотра атрибутов необходимо проверить наличие подстановок у вида просмотра. Единственная подстановка находится в библиотеке *‘# Товародвижение*‘.

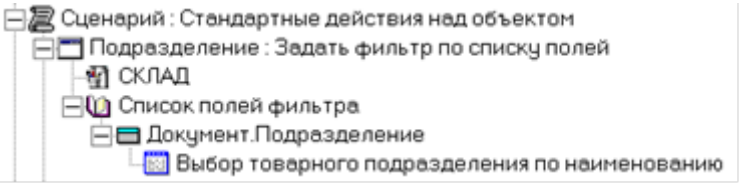


В подстановке указано, что еще три поля добавляются в список полей вида просмотра.

Поскольку больше подстановок не имеется, то определяем значения тех атрибутов, которые описаны в самом виде просмотра.

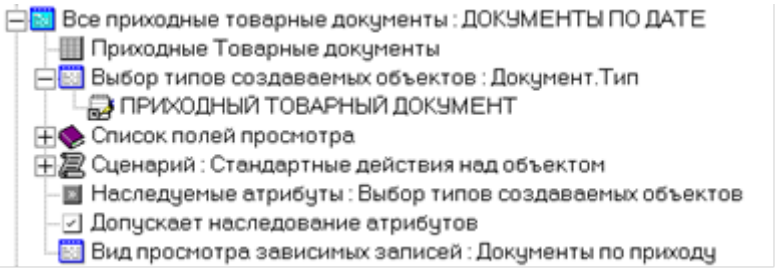
Для вида просмотра указан признак '*Сохранять текущие значения фильтров в профиле*'. Это позволит при повторном вызове вида просмотра работать с ранее установленными ограничениями на отображаемые данные.

Атрибут '*Параметры, обязательные при задании ограничений*' содержит ссылку на параметр документа '*Подразделение*'. Это означает, что в виде просмотра будут отображаться документы только по одному подразделению. Значение выбранного подразделения записывается в глобальной переменной '*Товарное подразделение (глоб. перем.)*'. Для выбора подразделения применяется специальный пункт на панели инструментов. Вот так выглядит описание этого пункта сценария.

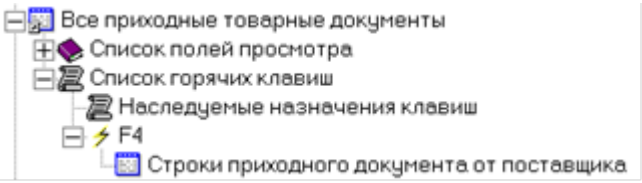


Атрибут '*Выбор типов создаваемых объектов*' определяет, что при работе с видом просмотра можно создавать новые документы класса '*ПРИХОДНЫЙ ТОВАРНЫЙ ДОКУМЕНТ*'.

Далее следует изучить описания атрибутов у родителей рассматриваемого вида просмотра. Описание родителя – вид просмотра '*Все приходные товарные документы*' – находится в дереве проекта чуть выше.



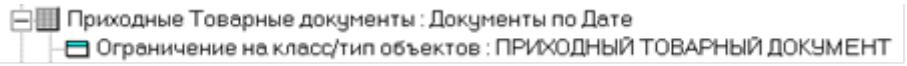
Сразу ищем подстановки. Единственная подстановка имеется в библиотеке '*# Товародвижение*'.



Из подстановки по правилу наследования добавляется описание атрибута ‘*Список горячих клавиш*’. По клавише F4 будет вызываться вид просмотра строк документа ‘*Строки приходного документа от поставщика*’. Этот же вид просмотра строк вызывается из формы документа, что было рассмотрено ранее.

Описание атрибута ‘*Список полей просмотра*’ нас не интересует, поскольку в описании вида просмотра ‘*Приходные товарные документы*’ имеется аналогичный атрибут, и там не применен атрибут ‘*Наследуемые поля, колонки*’. Т.е. список полей явно задан и не наследуется от родителя.

Атрибут ‘*Выборка*’ определяет класс отображаемых документов. Изучим выборку, на которую ссылается атрибут.



В виде просмотра будут отображены документы класса ‘*ПРИХОДНЫЙ ТОВАРНЫЙ ДОКУМЕНТ*’.

Атрибут ‘*Выбор типов создаваемых объектов*’ определяет, что при работе с видом просмотра можно создавать новые документы класса ‘*ПРИХОДНЫЙ ТОВАРНЫЙ ДОКУМЕНТ*’.

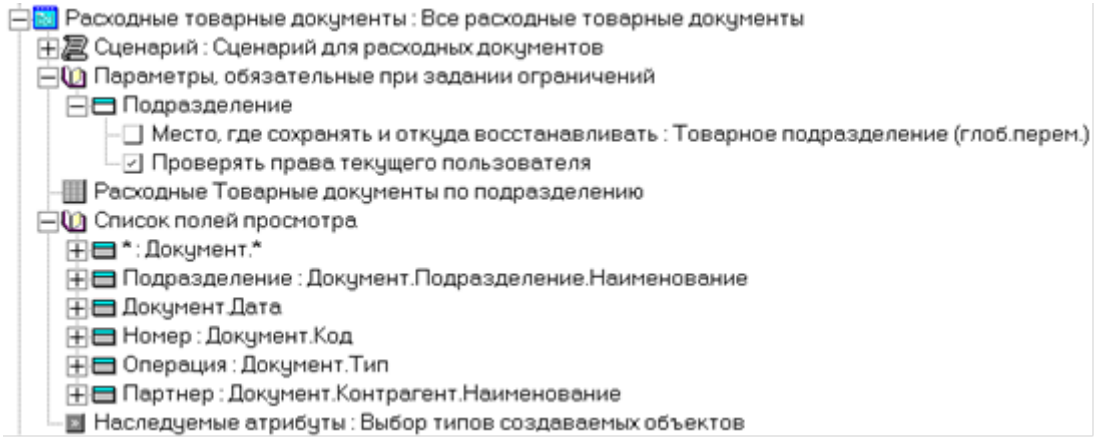
Поскольку для вида просмотра ‘*Приходные товарные документы*’ сценарий не был указан, то описание сценария наследуется от его родителя, т.е. из вида просмотра ‘*Все приходные товарные документы*’.

Атрибут ‘*Вид просмотра зависимых записей*’ задает вид просмотра для отображения наследников документа. При нажатии клавиши ‘*Серый плюс*’ будет высвечен вид просмотра ‘*Документы по приходу*’.

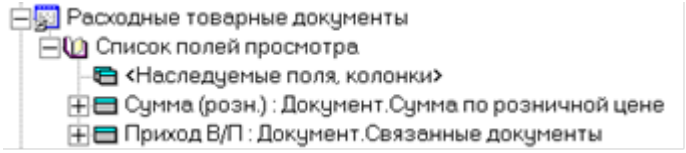
Изучив атрибуты непосредственного родителя вида просмотра можно исследовать и более дальних родителей: виды просмотра ‘*ДОКУМЕНТЫ ПО ДАТЕ*’, ‘*ПРОСМОТР ДОКУМЕНТОВ*’, ‘*ДОКУМЕНТЫ*’. Но, скорее всего, описание всех существенных характеристик рассматриваемого вида просмотра мы уже нашли.

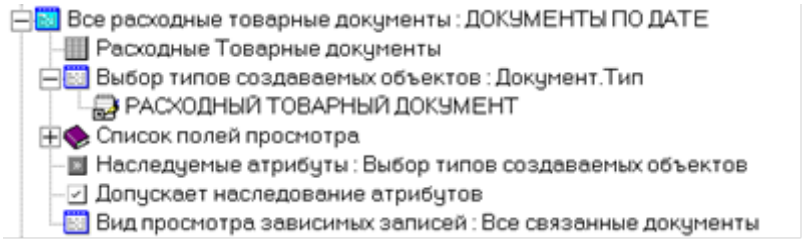
Вид просмотра ‘Расходные товарные документы’

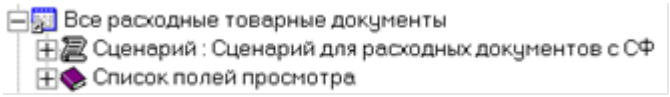
В библиотеке *‘! Базовый набор’*, раздел ‘*Виды просмотра*’, папка *‘! Базовый набор’*, папка *‘Документ*’, папка *‘Приходные документы*’ находится вид просмотра ‘*Расходные товарные документы*’.



Отыщем подстановку этого вида просмотра, а также описание и подстановку родителя.







В результате изучения найденных описаний можно сделать следующие выводы об атрибутах вида просмотра:

- Список полей задан в самом виде просмотра. В этот список добавляются два поля, указанные в подстановке.
- Сценарий рассмотрим позже.
- Атрибут ‘*Параметры, обязательные при задании ограничений*’ содержит ссылку на параметр документа ‘*Подразделение*’. Это означает, что в виде просмотра будут отображаться документы только по одному подразделению. Значение выбранного подразделения записывается в глобальной переменной ‘*Товарное подразделение (глоб. перем.)*’.
- Атрибут ‘*Выборка*’ определяет класс отображаемых документов. В виде просмотра будут отображены документы класса ‘*РАСХОДНЫЙ ТОВАРНЫЙ ДОКУМЕНТ*’.
- Атрибут ‘*Выбор типов создаваемых объектов*’ определяет, что при работе с видом просмотра можно создавать новые документы класса ‘РАС*ХОДНЫЙ ТОВАРНЫЙ ДОКУМЕНТ*’.

- Атрибут ‘*Вид просмотра зависимых записей*’ задает вид просмотра для отображения наследников документа. При нажатии клавиши ‘Серый плюс’ будет высвечен вид

просмотра ‘*Все связанные документы*’.

Следует отметить, что если какие-либо значения атрибутов не соответствуют нашим требованиям, то можно либо создать новый вид просмотра, либо применить подстановку вида просмотра для изменения значений атрибутов.

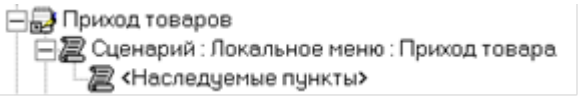
Сценарии для документов

В описании типов документов, форм и видов просмотра встречались атрибуты со ссылками на различные сценарии. Уточним значения этих атрибутов.

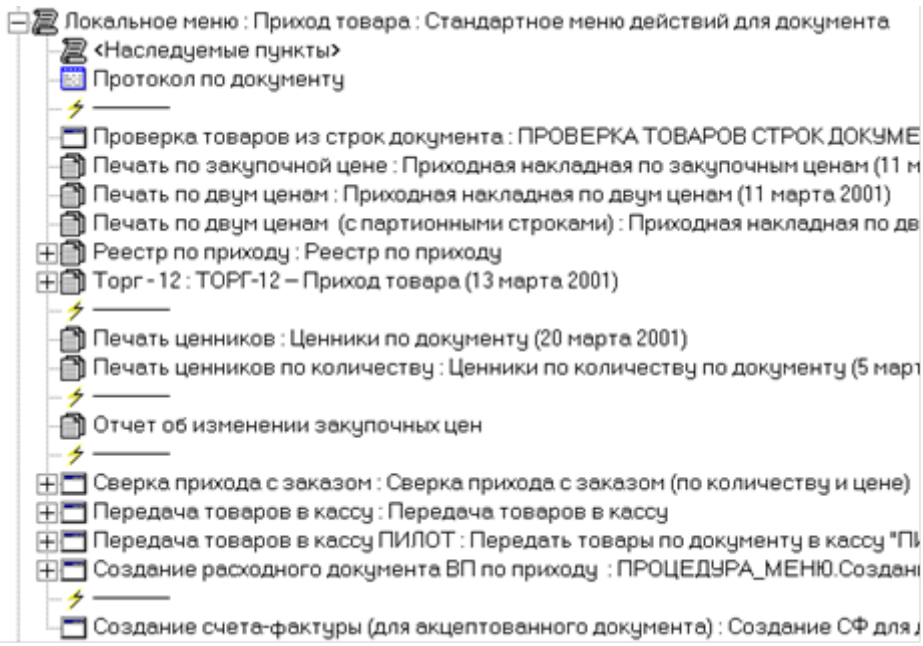
Контекстное меню

Сначала разберем сценарии у типов документов. Этот сценарий задает контекстное меню. Контекстное меню вызывается из вида просмотра при нажатии правой кнопки мыши. Выбранные в меню действия выполняются для текущей записи вида просмотра.

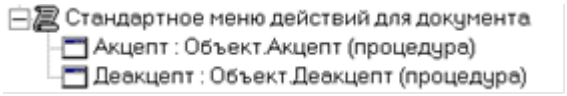
Еще раз рассмотрим подстановку типа документа ‘*Приход товаров*’.



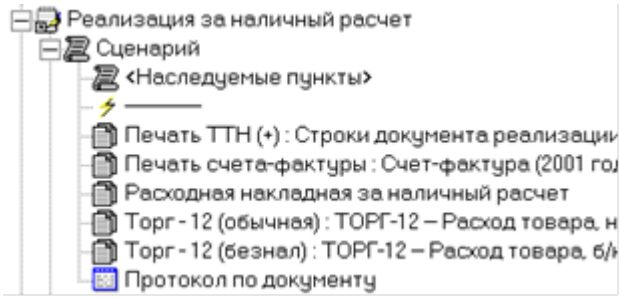
В подстановке указано, что пункты сценария наследуются из родителя сценария – из сценария ‘*Локальное меню: Приход товара*’.



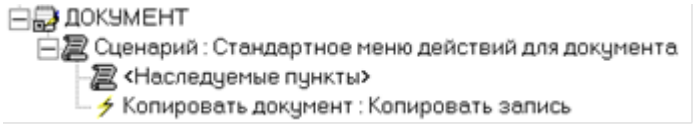
В список вместо атрибута ‘*Наследуемые пункты*’ следует добавить пункты из сценария ‘*Стандартное меню действий для документа*’.



Теперь изучим подстановку типа документа ‘*Реализация за наличный расчет*’.



Поскольку у сценария не указан родитель, то наследуемые пункты следует искать в описании типа документа. Непосредственно для типа документа сценарий не задан. Продолжаем поиски. Вскоре сценарий обнаруживается у одного из дальних родителей типа документа.



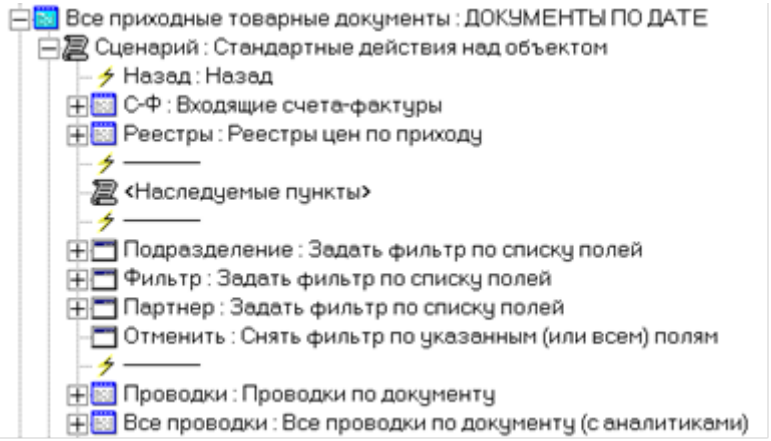
Добавляем пункты из уже рассмотренного сценария ‘*Стандартное меню действий для документа*’.

В итоге, в сценарии для типа документа ‘*Реализация за наличный расчет*’ атрибут ‘*Наследуемые пункты*’ заменяется на три пункта: ‘*Акцепт*’, ‘*Деакцепт*’ и ‘*Копировать документ*’.

Сценарии для видов просмотра

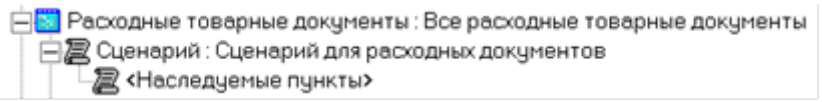
Теперь разберем указание сценария в видах просмотра документов. Сценарий в виде просмотра определяет панель инструментов.

Поскольку для вида просмотра ‘*Приходные товарные документы*’ сценарий не был указан, то описание сценария наследуется от его родителя, т.е. из вида просмотра ‘*Все приходные товарные документы*’.

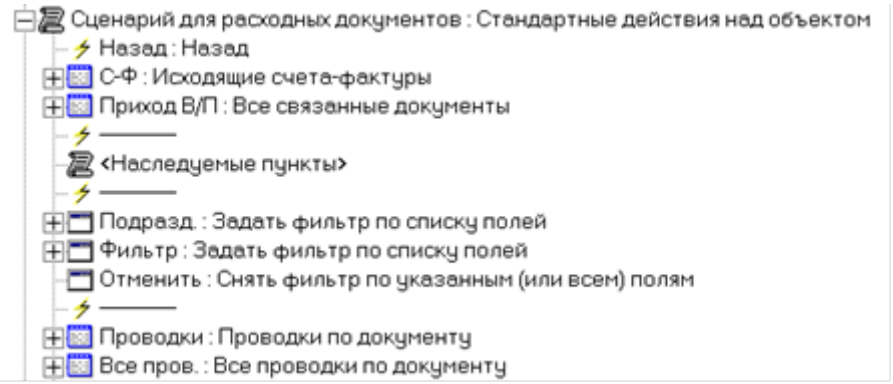


Атрибут ‘Наследуемые пункты’ заменяется на содержимое сценария ‘*Стандартные действия над объектом*’, поскольку именно на этот сценарий ссылается сценарий из вида просмотра.

Рассмотрим сценарий из вида просмотра ‘*Расходные товарные документы*’.



Наследуемые пункты следует искать у родителя данного сценария.

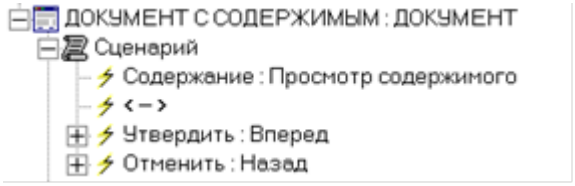


Как и в предыдущем случае, атрибут ‘Наследуемые пункты’ заменяется на содержимое сценария ‘*Стандартные действия над объектом*’.

Сценарии для форм

Перейдем к сценариям формы. Сценарий в форме задает панель инструментов формы.

Ни одна из рассматриваемых форм (‘*Приходный документ*’ и ‘*Реализация (продажа) покупателям (с НСП)*’) не содержит описание сценария. Последовательно просматриваем родителей форм. В описании одного из родителей (оказалось, что найденная форма является дальним родителем для обеих форм) сценарий имеется.



Этот сценарий будет применен в обеих формах.

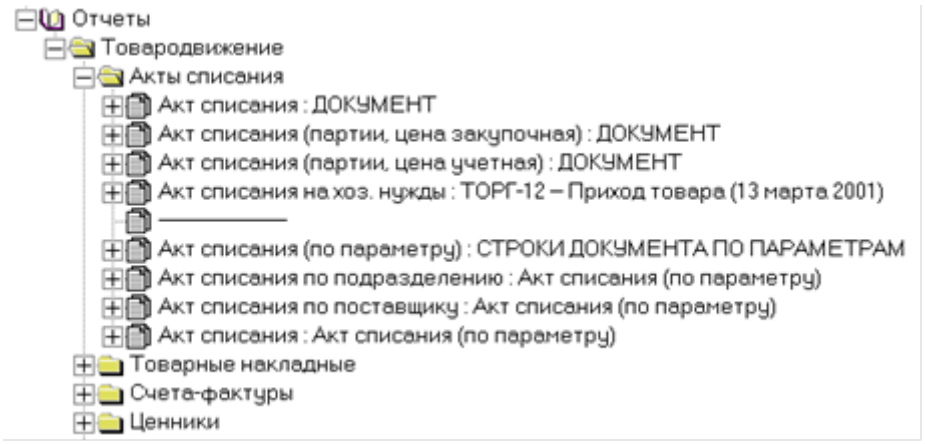
В заключение, рассмотрим ситуацию, когда список пунктов сценария не соответствует нашим требованиям. В этом случае возможны различные действия. Если сценарий ссылается на другой сценарий, то можно применить подстановку сценария. Либо завести новую подстановку типа документа, вида просмотра или формы, в которой указать новый сценарий.

> При изменении сценария необходимо быть особо внимательным, поскольку следует изучить подстановки не только самого сценария и родителей сценария, но и тех проектных элементов, где этот сценарий применяется.

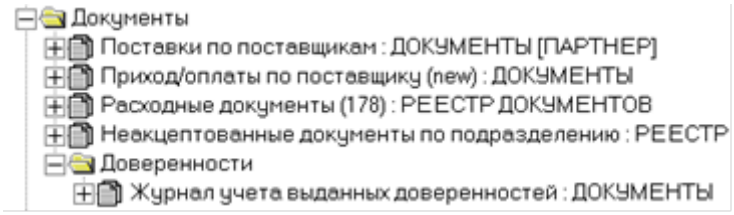
Отчеты по документам

Последний этап решения поставленной задачи состоит в описании отчетов по документам.

Отчеты по одному документу располагают в разделе ‘*Отчеты*’ внутри раздела ‘*Документ*’.



В раздел ‘*Отчеты*’ верхнего уровня библиотеки помещают отчеты по многим документам.



Этапы выполнения акцепта документа

Акцепт и **деакцепт** – эти два термина были введены для отслеживания состояния

документа.

Любой документ может быть либо акцептован, либо неакцептован. Неакцептованный документ является, по сути, заготовкой документа. И с точки зрения программы этот документ еще не готов. Такие документы доступны для редактирования и удаления.

Акцептованный документ утверждается пользователем путем выполнения специального действия – установкой признака акцепта. В момент установки акцепта запускается процедура формирования проводок по документу.

Для акцептованных документов возможно единственное действие – снять акцепт (или деакцепт). Смысл деакцепта – вернуть документ и базу данных в состояние, предшествующее акцепту. В процедуре деакцепта удаляются проводки по документу.

Акцепт бывает автоматический и отложенный. Автоматический акцепт выполняется сразу же после того, как пользователь завершает ввод документа. Документы с отложенным акцептом после выхода из редактирования остаются неакцептованными. Их нужно акцептовать вручную, вызывая команду акцепта из сценария.

Для того, чтобы внести изменения в акцептованный документ, нужно сначала снять с него акцепт, отредактировать документ, после чего заново его акцептовать. Однако следует помнить, что даже простое снятие и установка акцепта документа, без его редактирования, могут привести к тому, что изменяется информация в базе данных. Это происходит из-за того, что в процедуре акцепта могут быть выполнены некоторые необратимые изменения данных, которые невозможно восстановить при деакцепте.

Обычно, при описании полей вида просмотра заводится специальное поле, в котором отображается состояние документа. Если акцепт документа прошел успешно, то в поле появится знак акцепта - '*'. При этом сам документ и его строки становятся недоступными для редактирования и удаления. При попытке повторно акцептовать уже акцептованный документ никаких действий не выполняется и никаких сообщений не выдается.

Для удобства настройки процедуры акцепта и деакцепта разбиваются на этапы. Автор проекта имеет возможность указать действия, которые требуется выполнить на каждом этапе. Для того чтобы выделить такие действия в проекте применяется понятие метода.

Поскольку процедуры акцепта и деакцепта уникальны для каждого класса документов, то проектировщики указывают необходимые методы при описании классов документов.

С акцептом тесно связано понятие транзакции.

Транзакция - процедура внесения нескольких связанных изменений в БД. Применяется для поддержки целостности БД. Основной принцип обработки транзакций - транзакция должна быть выполнена либо целиком, либо не выполнена вовсе.

Когда говорят, что транзакция открывается, то имеют в виду, что дальнейшие действия будут выполняться только внутри транзакции. Закрыть транзакцию – попытаться

выполнить все перечисленные для транзакции действия. Точнее, сохранить в БД результат выполнения действий внутри транзакции. Если транзакция завершится успешно, то все перечисленные в ней действия будут выполнены. Если транзакция не завершится, то ни одно из действий внутри транзакции выполнено не будет.

Для транзакции акцепта документа указывается режим ее выполнения. При **нормальном** режиме акцепт записи документа и акцепт строк выполняются в единой транзакции. В **минимальном** режиме транзакция для акцепта записи документа открывается после обработки (акцепта) строк. Тем самым, акцепты каждой строки и акцепт записи документа выполняются в раздельных транзакциях. Минимальный режим применяется для избежания продолжительных блокировок во время акцепта документов, содержащих очень много строк.

Перейдем к рассмотрению этапов акцепта и деакцепта.

Этапы выполнения акцепта документа и связанные с ними методы

1. Проверка допустимости акцепта по условиям, заданным в атрибуте '*Права на выполнение действий над объектом*', команда 'А*кцепт*'.
2. Проверка допустимости акцепта по условиям, заданным в методе '**Акцепт. Предварительная проверка (до проведения расчетов)'.**
3. Проверка того, что указанный документ неакцептован. Если документ уже акцептован, то процедура акцепта прерывается.
4. Открытие транзакции для **нормального** режима обработки транзакций.
5. Обработка строк. Последовательно перебираются строки документа. Каждая строка акцептуется.
6. Расчет суммовых проводок. Последовательно перебираются строки документа и рассчитываются проводки, для которых установлен режим создания '*Суммовые (для строк документа)*'
7. Открытие транзакции для **минимального** режима обработки транзакций.
8. Создание проводок, рассчитанных на 6 этапе. Проводки сохраняются в базе данных.
9. Выполнение действий, указанных в методе '**Акцепт. Действия после обработки строк (если есть) и перед расчетом проводок**'.
10. Расчет и сохранение в базе данных остальных проводок документа.
11. Выполнение действий, указанных в методе '**Акцепт. Действия после расчета проводок, но перед акцептом**'.
12. Установка состояния '*акцептован*' для документа.
13. Запись в протокол сообщения о выполненном акцепте документа.
14. Закрытие транзакции как для **нормального** , так и для **минимального** режимов.
15. Выполнение действий, указанных в методе '**Акцепт.Действия после всех расчетов и собственно акцепта**'.

В **нормальном** режиме обработки транзакций акцепт строк документа, расчет

проводок документа и вызов методов акцепта самого документа происходит в единой транзакции. В случае **минимального** режима каждая из строк акцептуется в отдельной (самостоятельной) транзакции. И возможна следующая ситуация. Если акцепт какой-либо из строк завершится с ошибкой, то акцепт как оставшихся строк, так и самого документа выполнен не будет. В результате некоторые строки документа будут акцептованы (те строки, которые были обработаны до возникновения ошибки), а остальные строки останутся не акцептованы. Поэтому минимальный режим следует использовать аккуратно и иметь в виду потенциальную возможность возникновения вышеописанной ситуации.

Этапы выполнения короткого акцепта документа и связанные с ними методы

1. Проверка допустимости акцепта по условиям, заданным в атрибуте '*Права на выполнение действий над объектом*', команда 'А*кцепт*'.
2. Проверка допустимости акцепта по условиям, заданным в методе '**Акцепт. Предварительная проверка (до проведения расчетов)'.**
3. Проверка того, что указанный документ неакцептован. Если документ уже акцептован, то процедура акцепта прерывается.
4. Открытие транзакции для **нормального** режима обработки транзакций.
5. Расчет суммовых проводок. Последовательно перебираются строки документа и рассчитываются проводки, для которых установлен режим создания '*Суммовые (для строк документа)*'
6. Открытие транзакции для **минимального** режима обработки транзакций.
7. Создание проводок, рассчитанных на 5 этапе. Проводки сохраняются в базе данных.
8. Выполнение действий, указанных в методе '**Короткий** а**кцепт**'.
9. Расчет и сохранение в базе данных остальных проводок документа.
10. Выполнение действий, указанных в методе '**Акцепт. Действия после расчета проводок, но перед акцептом**'.
11. Установка состояния '*акцептован*' для документа.
12. Запись в протокол сообщения о выполненном акцепте документа.
13. Закрытие транзакции как для **нормального** , так и для **минимального** режимов.

Отличия этапов выполнения короткого акцепта от обычного акцепта следующие :

- Не вызывается обработка строк документа в предположении, что все строки уже акцептованы (либо в результате автоакцепта строк, либо после короткого деакцепта документа).
- На 7 этапе выполняются действия, указанные в методе '**Короткий акцепт'**, а не в методе '**Акцепт. Действия после обработки строк (если есть) и перед расчетом проводок**'.
- Не вызывается этап для выполнения действий, указанных в методе '**Акцепт. Действия после всех расчетов и собственно акцепта**'

> Очевидно, что короткий акцепт выполняется значительно быстрее обычного акцепта. Однако при использовании короткого акцепта необходима полная уверенность, что все строки документа акцептованы. Если документ будет содержать хотя бы одну не акцептованную строку, то в результате короткого акцепта такого документа будет нарушена логическая целостность данных. Добавление в короткий акцепт процедуры проверки строк бессмысленно, так как при этом в значительной мере будет потеряно преимущество короткого акцепта – малое время выполнения.

Этапы выполнения деакцепта документа и связанные с ними методы

1. Проверка допустимости акцепта по условиям, заданным в атрибуте ‘*Права на выполнение действий над объектом*’, команда ‘*Деакцепт*’.
2. Проверка того, что указанный документ акцептован. Если документ уже деакцептован, то процедура деакцепта прерывается.
3. Выполнение действий, указанных в методе ‘**Пре-деакцепт**’. Если действия завершаются с ошибкой, то процедура деакцепта прерывается.
4. Открытие транзакции.
5. Выполнение действий, указанных в методе ‘**Деакцепт**’.
6. Закрытие транзакции.
7. Выполнение действий, указанных в методе ‘**Пост-деакцепт**’.

Этапы выполнения короткого деакцепта документа и связанные с ними методы

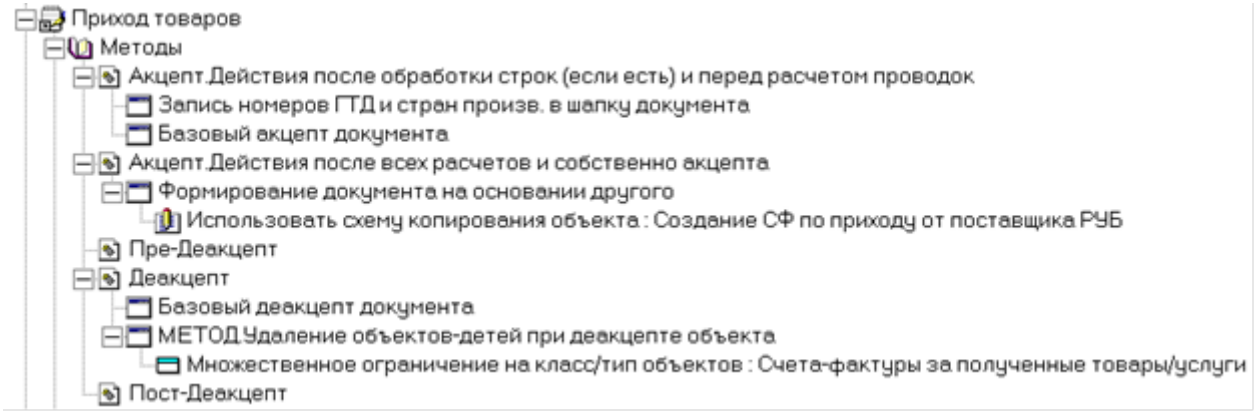
1. Проверка допустимости акцепта по условиям, заданным в атрибуте ‘*Права на выполнение действий над объектом*’, команда ‘*Деакцепт*’.
2. Проверка того, что указанный документ акцептован. Если документ уже деакцептован, то процедура деакцепта прерывается.
3. Открытие транзакции.
4. Выполнение действий, указанных в методе ‘**Короткий деакцепт**’.
5. Закрытие транзакции.

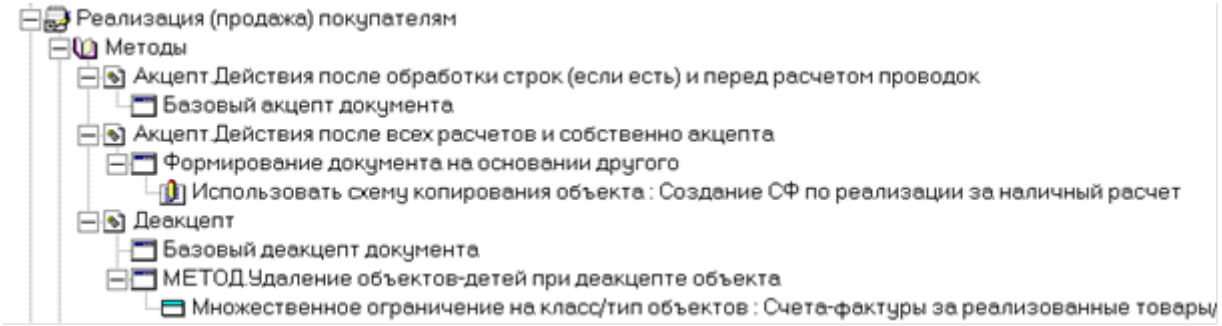
Отличия этапов выполнения короткого деакцепта от обычного деакцепта заключаются в том, что не выполняются действия, указанные в методах ‘**Пре-деакцепт**’ и ‘**Пост-деакцепт**’.

Процедуры для акцепта и деакцепта документа

При описании методов используются процедуры-методы из раздела ‘*Процедуры*’ верхнего уровня библиотеки или из раздела ‘*Процедуры*’ внутри раздела ‘*Документ*’.

Для обсуждаемого примера описание методов выглядит следующим образом.





Версия #2

Демонов Сергей создал Tue, Mar 29, 2022 5:29 PM

Демонов Сергей обновил Tue, Mar 29, 2022 5:56 PM