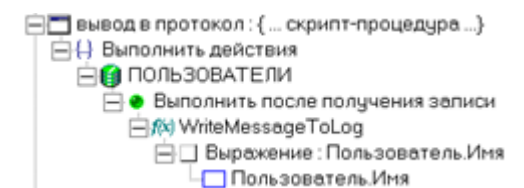


* Отчёты

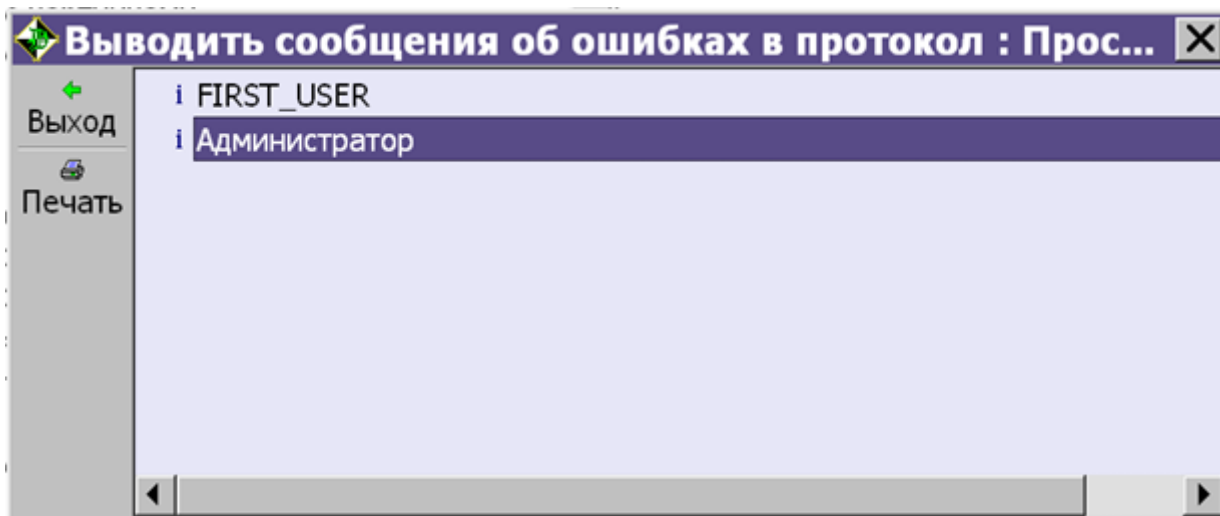
Процедуры, предназначенные для отбора, сортировки, группировки данных и выдачи их на экран, принтер, в файл, часто называют отчетами. Это не совсем точное название, поскольку отчет – это, скорее, результат выполнения такой процедуры. Но оставим это привычное слово для обозначения процедуры, а для второго случая будет применяться слово результат.

В общем случае, отчетом может быть любая скрипт-процедура.

Например, запись сообщений в протокол.

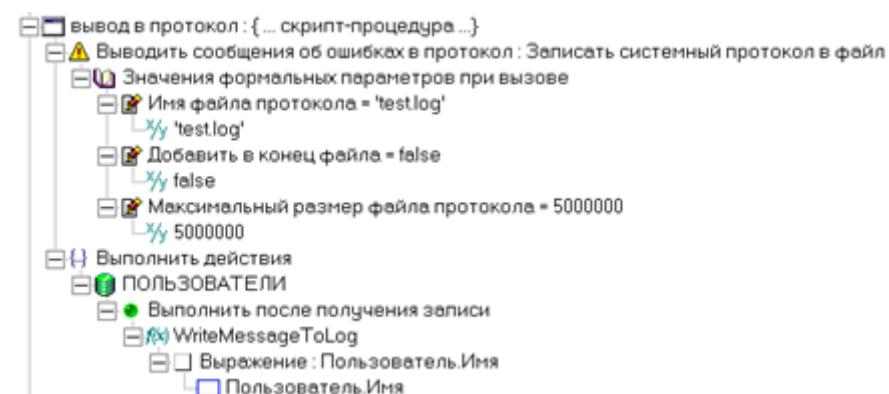


Перебираются все пользователи, и их имена выводятся в протокол.



Просмотр результата выполнения процедуры вывода в протокол.

Можно немного изменить процедуру и не отображать результат на экране, а записывать протокол в файл.



В рассматриваемом примере для этого достаточно указать соответствующий атрибут.

Процедуры в вышеприведенных примерах являются отчетами, но отчетами очень простыми. В этих процедурах никаким образом не структурируется и не форматируется результат работы. Если требуется представить результат выполнения отчета-скрипта в красивой и понятной форме, то применяются шаблоны отчетов. С помощью шаблона не только описывается содержание результата (что нужно отобразить), но и указывается форма результата (как нужно отобразить). Сначала рассмотрим различные формы представления результата отчета.

Результат отчета можно отобразить с помощью редактора обычных текстовых файлов, HTML браузера, редактора текстов Microsoft WinWord, редактора электронных таблиц Microsoft Excel. Все эти программы работают по сходному алгоритму. Программа получает файл и отображает его в форме, присущей именно этой программе. Форма отображения задается специальными инструкциями, расположенными в файле. Причем, каждая программа строго требует соблюдения определенного формата для обрабатываемых файлов.

Теперь разберем содержание результата отчета. В общем случае, результат содержит информацию двух видов: полученные данные и обрамляющий их текст. Или, другими словами. Одна составная часть результата постоянна и не изменяется при разных вызовах отчета-скрипта. Это статичное изображение. Другая часть зависит от того, какие данные обрабатываются при очередном вызове отчета. Отображение этой части выглядит как наложение фрагментов текста на постоянную часть.

Каждый шаблон имеет внутреннюю структуру. Эта структура отражается с помощью блоков. Блок – это составная часть шаблона. Список блоков и последовательность их вызова однозначно определяют структуру шаблона.

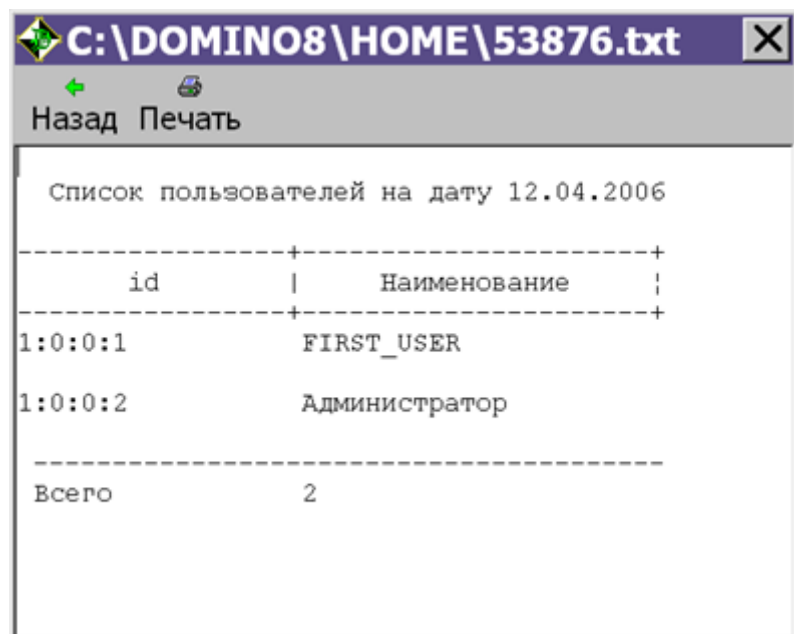
Блок содержит как постоянный текст, так и подставляемые данные. Блоки помещаются в результат отчета только целиком. Блоки можно выводить многократно, каждый раз с новыми полученными данными.

Блоки применяются при отборе данных для отчета. Удобнее и быстрее отбирать данные порциями. Сначала подбираются все данные, которые будут напечатаны в первом блоке. Печатается первый блок. Затем, подбираются данные для второго блока. Печатается

второй блок. И так далее. Если требуется напечатать блок несколько раз, то отбор данных и печать блока обычно проводится в цикле.

Итак, в шаблоне описывается содержание результата отчета и форма представления этого результата. Поскольку форма представления накладывает ограничения на правила создания шаблона, то шаблоны стали классифицировать по названиям этих форм.

Рассмотрим пример отображения в текстовом формате (TEXT).



The screenshot shows a text file window titled "C:\DOMINO8\HOME\53876.txt". The window contains a report titled "Список пользователей на дату 12.04.2006". The report is formatted as a table with two columns: "id" and "Наименование". The data rows are:

id	Наименование
1:0:0:1	FIRST_USER
1:0:0:2	Администратор
Всего	2

Результат выполнения скрипта-отчета.

“ {{ЗАГОЛОВОК}}

Список пользователей на дату {ДАТА}

-----+-----+

id | Наименование |

-----+-----+

{{ПОЛЬЗОВАТЕЛЬ}}

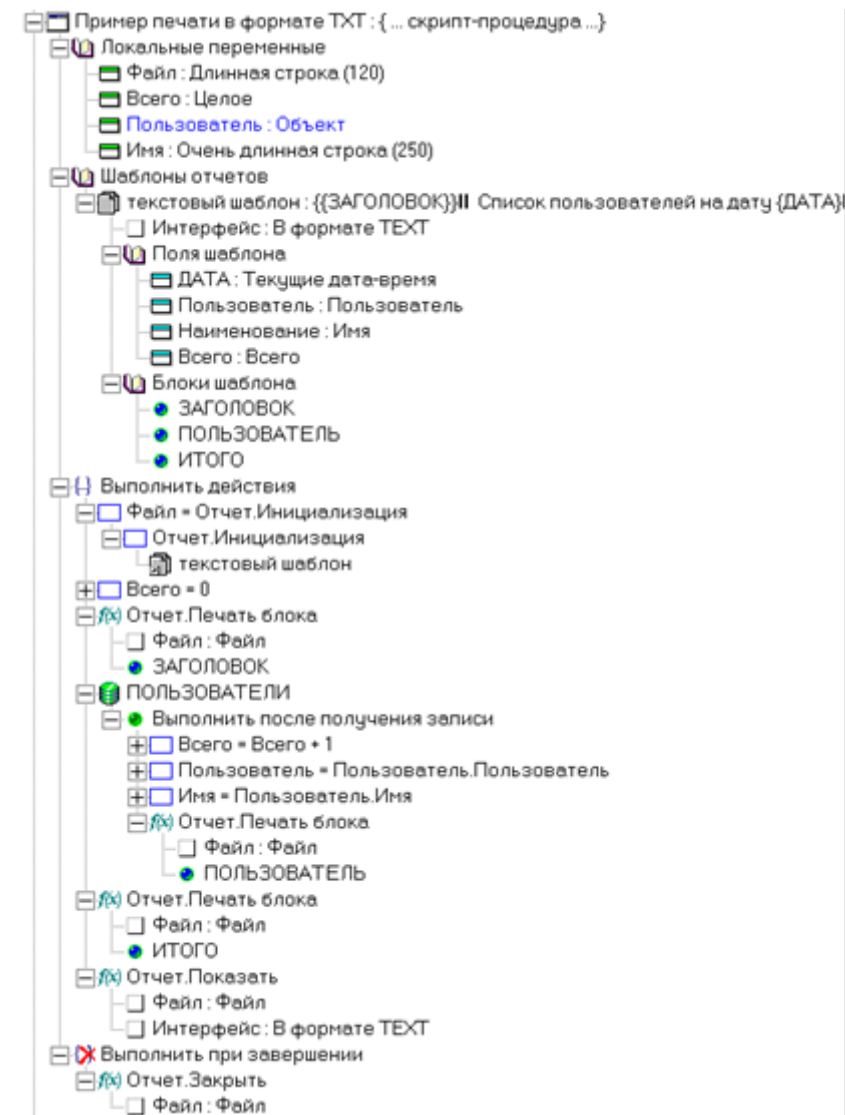
[<Пользователь][<Наименование]

{{ИТОГО}}

Всего

{Всего}

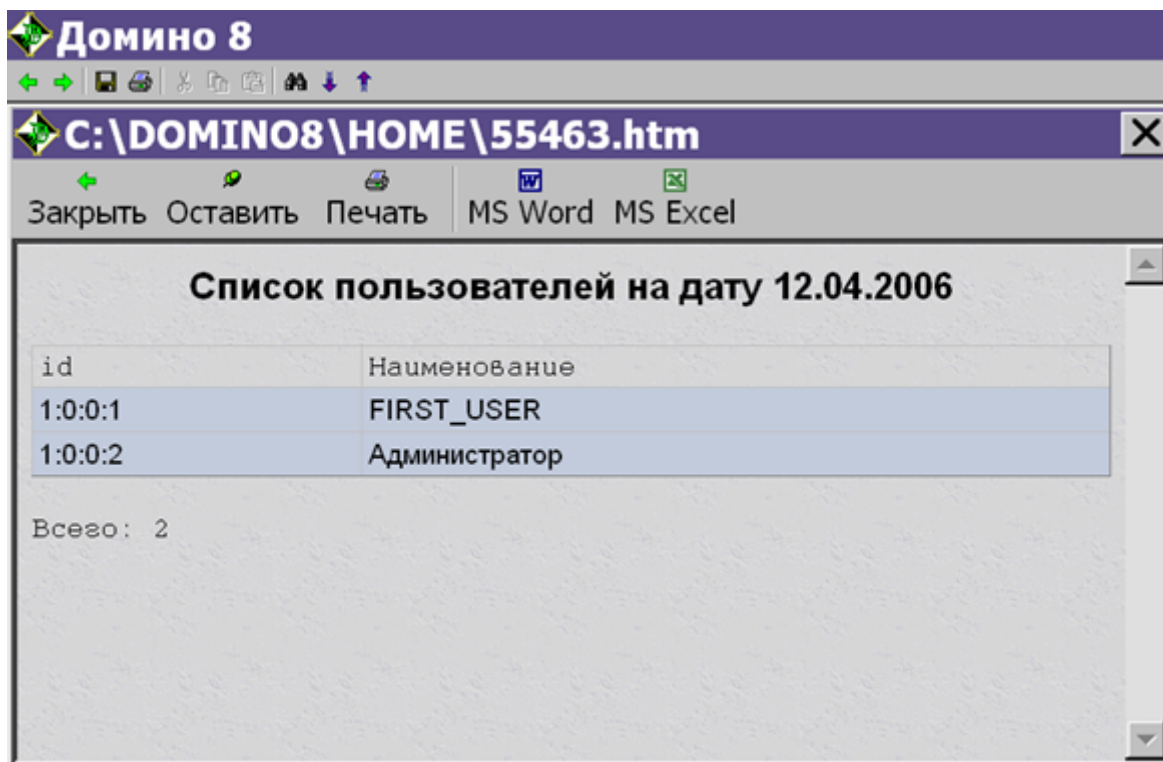
Шаблон отчета. При печати полей указано выравнивание влево. Шаблон содержит три блока: ЗАГОЛОВОК, ПОЛЬЗОВАТЕЛЬ, ИТОГО. Блок ПОЛЬЗОВАТЕЛЬ печатается столько раз, сколько имеется записей пользователей. В данном шаблоне этот блок применен для заполнения таблицы.



Текст скрипта-отчета.

Изобразительных возможностей для текстового формата очень мало, поэтому такой формат применяется редко. Основное использование – это печать кассовых чеков на матричный принтер.

Посмотрим, как можно отобразить тот же текст в формате HTML.



Пример печати списка пользователей в формате HTML

Текст HTML шаблона по сравнению в текстовым шаблоном выглядит сложнее.

“ {{ЗАГОЛОВОК}}

```
<html>
```

```
<head>
```

```
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
```

```
  <link rel=stylesheet href="file:{STYLE}" type="text/css">
```

```
</head>
```

```
<body>
```

```
<H1> Список пользователей на дату {ДАТА} </H1>
```

```
<table class="MainTable" border="1" width="100%" cellpadding="0" cellspacing="4" align=center>
```

```
<tr class="TableHeader">
```

```
<td> id </td>

<td> Наименование </td>

</tr>

{{ПОЛЬЗОВАТЕЛЬ}}

<tr class="TableBody">

    <td align=left>{Пользователь}</td>

    <td align=left>{Наименование}</td>

</tr>

{{ИТОГО}}

</table>

<p>Всего: {Всего}

</body></html>
```

Структура шаблона (список блоков) осталась прежней, поскольку отбор данных не меняется.

В тексте отчета-скрипта достаточно изменить ссылку на другой шаблон.

Шаблоны в формате HTML имеют много возможностей по управлению изображением и достаточно просто описываются. Поэтому это самые распространенные шаблоны в проектах на ДОМИНО. Подробнее о том, какие бывают шаблоны и как они описываются, написано ниже.

Теперь перейдем к разбору текста отчета-скрипта из приведенного примера.

В начале находится вызов функции *‘Отчет.Инициализация’*. Далее следует вызов процедуры *‘Отчет.Печать блока’* для блоков *‘Заголовок’*, *‘Пользователь’*, *‘Итого’*. Затем вызывается процедура *‘Отчет.Показать’*. И заканчивается процедура блоком *‘Выполнить при завершении’*, в котором запускается процедура *‘Отчет.Заккрыть’*.

Наличие вызовов этих процедур и функции является необходимым для использования шаблона отчета.

Схема выполнения отчета-скрипта для печати с шаблоном должна содержать следующую последовательность вызовов этих процедур:

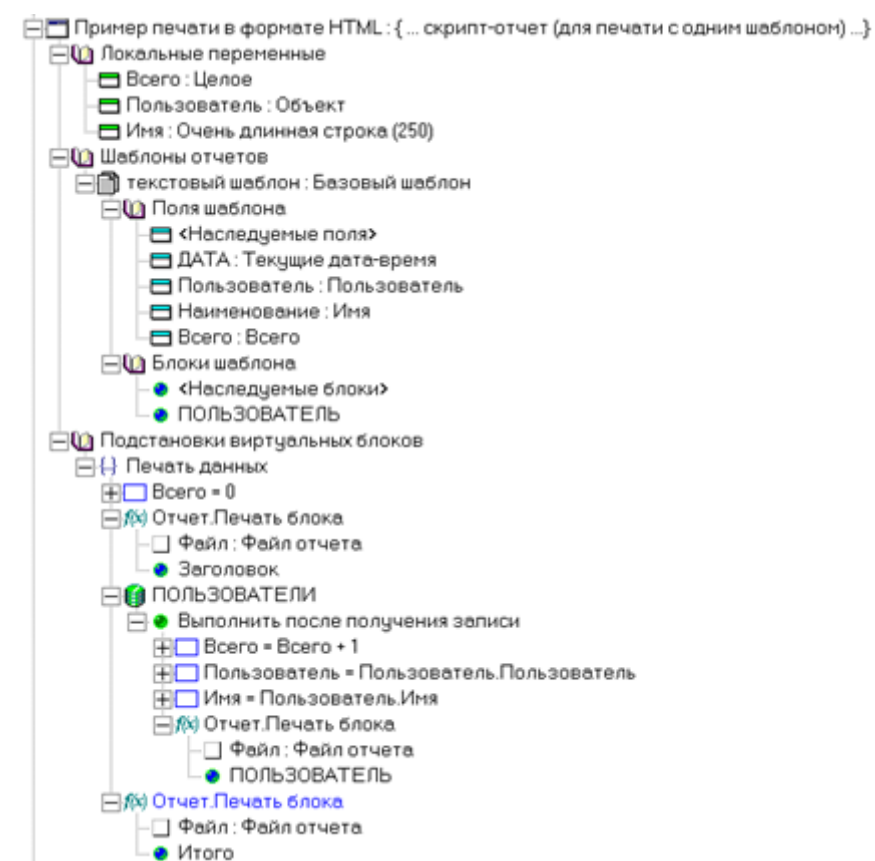
- Отчет.Инициализация
- Отчет.Печать блока – необходимое число раз
- Отчет.Показать
- Отчет.Заккрыть

Подробнее об этих процедурах написано ниже.

Для упрощения создания отчетов разработаны специальные базовые отчеты-скрипты, в которые вынесены инициализация, отображение и закрытие отчета. Там же определены виртуальные блоки, содержимое которых можно заполнить в отчетах-наследниках.

При использовании базового отчета-скрипта проектировщику необходимо лишь описать шаблон (со ссылкой на базовый шаблон) и из подстановки виртуального блока 'Печать данных' вызвать функцию печати нужного блока.

Посмотрим, как можно переписать пример печати справочника пользователей.



Вместо двух блоков 'Выполнить действия' и 'Выполнить при завершении' появляется постанровка виртуального блока 'Печать данных'. Внутри этого блока находятся вызовы процедуры 'Отчет.Печать блока'. Изменилось описание шаблона. Теперь шаблон ссылается на базовый шаблон. В описании блоков шаблона используется наследование блоков из базового шаблона, но это делать не обязательно. Можно определить все блоки заново.

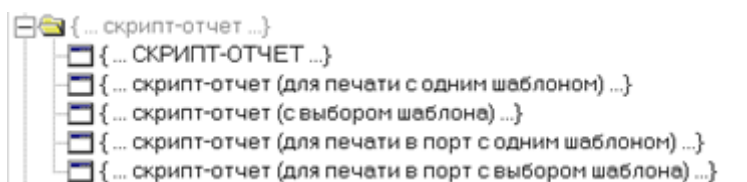
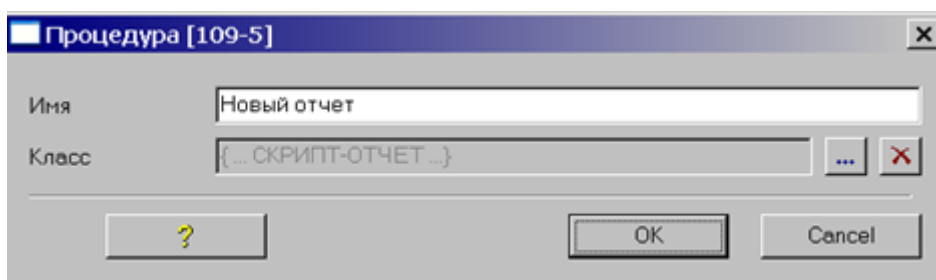
Текст процедуры стал короче и понятнее, поскольку из него удалены обязательные вызовы одной функции и двух процедур.

Рекомендуется при написании отчетов-скриптов ссылаться на базовые отчеты.

В последующих разделах рассматриваются отчеты-скрипты, являющиеся наследниками базовых отчетов.

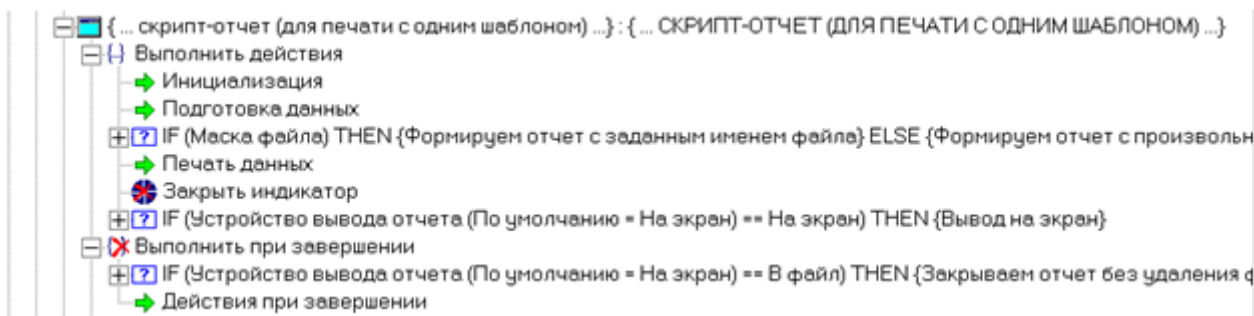
Атрибуты отчетов-скриптов

Для создания отчета на языке скриптов необходимо при заполнении поля класс указать любой из элементов папки {...скрипт-отчет...}.



Последние четыре пункта являются наследниками первого отчета и отличаются от него тем, что для них указано описание атрибутов 'Выполнить действия' и 'Выполнить при завершении'.

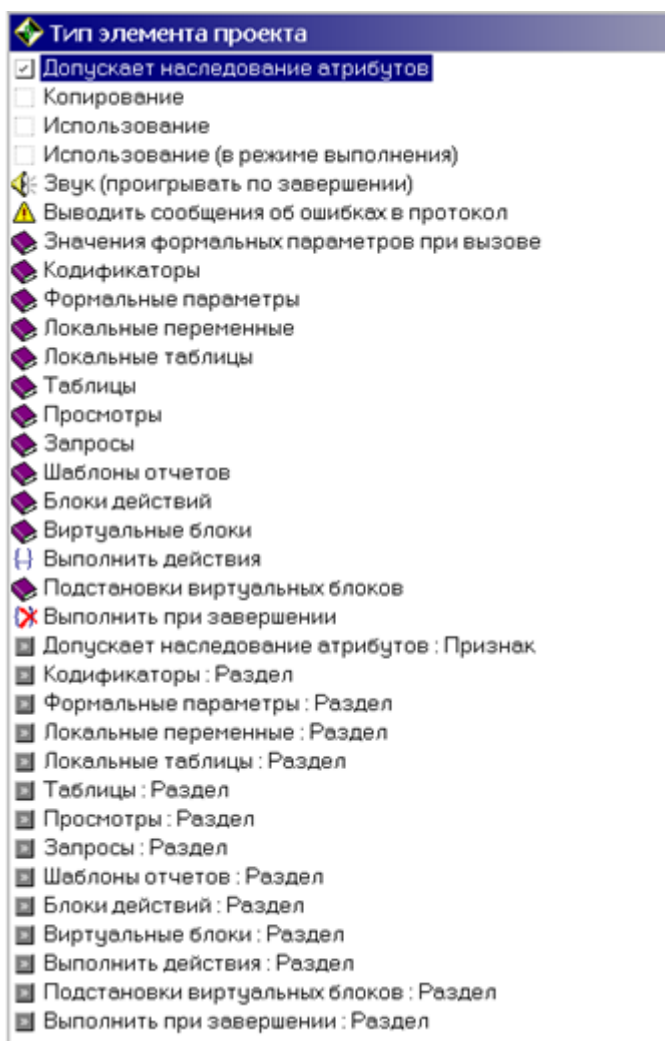
Например.



В блоках 'Выполнить действия' и 'Выполнить при завершении' вызываются виртуальные блоки 'Инициализация', 'Подготовка данных', 'Печать данных', 'Действия при завершении'.

Схемы выполнения большинства отчетов аналогичны вышеприведенной, поэтому при разработке новых отчетов удобнее ссылаться именно на последние пункты. При этом проектировщику необходимо заполнить содержимое виртуальных блоков.

При описании отчетов доступны следующие атрибуты:



Назначение большинства атрибутов скрипта-отчета совпадает с назначением соответствующих атрибутов обычной процедуры. Другая часть атрибутов присуща именно отчетам. Такие атрибуты рассмотрим подробнее.

- **Допускает наследование атрибутов** – при установленном признаке в описании наследника можно будет переопределить атрибуты.
- **Копирование** – условие допустимости копирования проектировщиками описания отчета.
- **Использование** – условие доступа проектировщиков к отчету.
- **Использование (в режиме выполнения)** – условие доступа пользователей к отчету.
- **Звук (проигрывать при завершении)** – по завершении выполнения отчета будет проиграна указанная мелодия.
- **Выводить сообщения об ошибках в протокол** – при заполненном атрибуте сообщения об ошибках будут записаны в указанный файл, а не высвечены на экране.
- **Значения формальных параметров при вызове** – список значений формальных параметров. Раздел заполняется при вызове отчета.
- **Кодификаторы** – Здесь описываются используемые внутри процедуры кодификаторы. Кодификатор – набор именованных проектных элементов, используемых в качестве перечислимого типа данных. Обычно кодификатор применяется для выбора пользователем одной альтернативы из списка возможных вариантов.
- **Формальные параметры** – список формальных параметров отчета.
- **Локальные переменные** – список локальных переменных отчета.
- **Локальные таблицы** – Здесь описываются используемые внутри отчета локальные таблицы. Под локальной таблицей можно понимать двумерную матрицу из строк и столбцов с описанием правил сортировки и группировки строк.
- **Таблицы** – Здесь описываются используемые внутри отчета таблицы сервера базы данных. Описанные здесь таблицы могут быть использованы в SQL конструкциях (операторах) отчета.
- **Просмотры** – Здесь также описываются таблицы сервера БД. Эти таблицы вынесены из раздела 'Таблицы' для того, чтобы программа при запуске не изменяла описание этих таблиц на сервере в соответствии с описанием в проекте.
- **Запросы** – Здесь описываются используемые внутри процедуры SQL запросы к серверу базы данных. Запросы могут быть описаны либо целиком, либо сюда могут быть вынесены их общие части. Запросы выносятся из тела отчета в данный раздел только в случае их многократного вызова. В противном случае целесообразнее описывать запросы непосредственно в теле отчета.
- **Шаблоны отчетов** – Здесь описываются шаблоны отчетов.
- **Блоки действий** – Здесь описываются блоки команд (операторов), вынесенные из основного блока процедуры. Блоки, описанные в этом разделе, в отличие от блоков основного тела процедуры, не вызываются автоматически, а требуют явного вызова.
- **Виртуальные блоки** – Используются при наследовании процедур с перенесением части логики в программу-наследника. Расположенные в этом разделе блоки не имеют содержимого. В теле процедуры имеются ссылки на виртуальные блоки, но свое содержимое виртуальные блоки получают в разделе '*Подстановки виртуальных блоков*' в процедуре-наследнике.

- **Подстановки виртуальных блоков** – В этом разделе виртуальные блоки получают свое содержимое в виде последовательности команд (операторов). При наследовании от базового отчета доступны следующие виртуальные блоки: 'Инициализация', 'Подготовка данных', 'Печать данных', 'Действия при завершении'. Подстановка блока 'Печать данных' является обязательной.
- **Выполнить действия** - Основной блок команд (операторов), автоматически выполняемый при запуске отчета. При наследовании от базового отчета данный раздел недоступен.
- **Выполнить при завершении** – Последовательность операторов, которые будут выполнены при завершении отчета. При наследовании от базового отчета данный раздел недоступен.
- **Наследуемые атрибуты** – задают наличие описанных выше разделов и признаков в отчете-наследнике.

<input checked="" type="checkbox"/>	Допускает наследование атрибутов : Признак
<input checked="" type="checkbox"/>	Кодификаторы : Раздел
<input checked="" type="checkbox"/>	Формальные параметры : Раздел
<input checked="" type="checkbox"/>	Локальные переменные : Раздел
<input checked="" type="checkbox"/>	Локальные таблицы : Раздел
<input checked="" type="checkbox"/>	Таблицы : Раздел
<input checked="" type="checkbox"/>	Просмотры : Раздел
<input checked="" type="checkbox"/>	Запросы : Раздел
<input checked="" type="checkbox"/>	Шаблоны отчетов : Раздел
<input checked="" type="checkbox"/>	Блоки действий : Раздел
<input checked="" type="checkbox"/>	Виртуальные блоки : Раздел
<input checked="" type="checkbox"/>	Выполнить действия : Раздел
<input checked="" type="checkbox"/>	Подстановки виртуальных блоков : Раздел
<input checked="" type="checkbox"/>	Выполнить при завершении : Раздел

Шаблоны отчетов

В шаблоне отчета описываются содержание и форма представления результата отчета.

Шаблон отчета состоит из текстовой части и атрибутов шаблона.

Текстовая часть шаблона содержит изображение результата отчета, в которое добавлены специальные элементы двух видов (для оформления и для выделения полей и блоков) .

Элементы первого вида требуются программе, применяемой для отображения результата. С помощью этих элементов оформляется высвечиваемый текст. Результат отчета может быть представлен с помощью следующих программ:

- редактор текстовых файлов
- HTML браузер
- редактор текстов Microsoft WinWord
- редактор электронных таблиц Microsoft Excel

Для каждой из этих программ применяются собственные (отличные от других) элементы оформления. Кроме того, программа сможет корректно отобразить текст только в том случае, если текст сформирован по правилам (в формате), накладываемым этой программой. Редактор текстовых файлов отображает текстовые файлы. Для HTML броузера требуются тексты в формате HTML. WinWord и Excel работают только со 'своими' файлами.

По этой причине текстовую часть шаблона приходится создавать в формате той программы, которая будет использоваться для его отображения.

- Формат TEXT для текстового редактора
- Формат HTML для HTML броузера
- Форматы RTF и MSWord Formatter (DOT) для Microsoft WinWord (Excel)

Внешне файлы разного формата различаются расширениями.

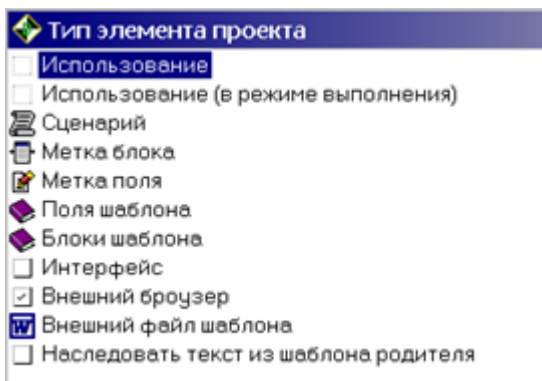
Шаблоны форматов TEXT и HTML могут быть созданы непосредственно в проекте. Файлы с шаблонами для WinWord подключаются с помощью атрибута *'Внешний файл шаблона'*.

В текстовую часть шаблона также включаются специальные элементы для ДОМИНО. Эти элементы выделяют месторасположение данных внутри обрамляющего текста. Выделяются поля и блоки. На место полей шаблона будут подставлены данные. Блоки необходимы потому, что являются минимальной единицей печати. Блок может быть напечатан только целиком.

Процесс обработки шаблона состоит из двух этапов. На первом этапе ДОМИНО на основе шаблона создает файл, в котором на место полей вставляет данные. Данные вставляются по-блочно, т.е. заполняются все поля одного блока. Порядком и частотой печати блоков управляет скрипт-отчет. На втором этапе запускается программа отображения. Программа высвечивает полученный файл в соответствии с имеющимися в файле элементами оформления.

В том случае, когда отчеты связаны цепочкой наследования, то шаблон отчета-ребенка может ссылаться на шаблон отчета-родителя. При этом, если отчет-ребенок не содержит текстовую часть, то текстовая часть берется из ближайшего родителя. Блоки и поля ребенка наследуются от родителя, если в них указаны элементы *'<Наследуемые блоки>'* и *'<Наследуемые поля>'*. В противном случае блоки и поля определяются заново.

При описании шаблона отчета доступны следующие атрибуты:



- **Использование** – условие доступа проектировщиков к шаблону. Используется только при наследовании шаблонов.
- **Использование (в режиме выполнения)** – условие доступа пользователей к шаблону. Имеет смысл только для отчетов с выбором шаблона. Не используется для отчетов с одним шаблоном.
- **Сценарий** – содержит список пунктов на панели инструментов при выводе файла отчета с помощью внутренней интерфейсной программы. Не используется для шаблонов в формате RTF.
- **Метка блока** – изменение меток начала и конца блока. Метка блока – это последовательность символов, указывающая на начало или конец блока.

По умолчанию используются следующие метки:

- {{ - начало блока,
- }} - конец блока.

Между метками располагается имя блока.

Данный атрибут применяется для шаблонов в формате HTML и TEXT.

Переопределять метки блока не рекомендуется. Изменение меток допускается в том случае, если в тексте шаблона необходимо использование зарезервированных символов для других целей.

- **Метка поля** – изменение меток начала и конца поля. Метка поля – это последовательность символов, указывающая на начало или конец поля.

По умолчанию используются следующие метки:

- { - начало поля,
- } - конец поля,
- [- начало поля фиксированной ширины для формата TEXT,
-] - конец поля фиксированной ширины для формата TEXT.

Между метками располагается имя поля.

Данный атрибут применяется для шаблонов в формате HTML и TEXT.

Переопределять метки поля не рекомендуется. Изменение меток рекомендуется в том случае, если в тексте шаблона необходимо использование зарезервированных символов для других целей.

- **Поля шаблона** – данный раздел содержит список полей шаблона. Имена полей используются при описании текстовой части шаблона. Поле может ссылаться на переменную, выражение, функцию или константу. В этом случае в процессе генерации отчета вместо поля шаблона будет подставлено значение переменной, константы или результат вычисления выражения, функции.
- **Блоки шаблона** – данный раздел содержит список блоков шаблона. Имя блока применяется при описании текстовой части шаблона и при вызове функции 'Отчет.Печать блока'.
- **Интерфейс** – определяет форму и способ отображения результата отчета. Кроме того, выбранный интерфейс определяет вид и правила написания текстовой части шаблона (формат шаблона).

Атрибут может иметь одно из следующих значений:



- **В формате HTML** – текст в формате HTML. Для таких шаблонов действуют все правила, существующие для текстов в формате HTML.

Если не указан атрибут '*Внешний файл отчета*', то вывод сформированного файла осуществляется внутренней интерфейсной программой.

Если же атрибут '*Внешний файл отчета*' указан, то в зависимости от расширения указанного файла внутренняя интерфейсная программа вызовет ту компоненту, которая в операционной системе назначена для работы с данным типом файлов.

- **В формате TEXT** – формат обычного текстового файла. Никаких дополнительных правил для описания шаблона не имеется.

Использование пары символов '{' и '}' означает переменную ширину поля. Т.е. ширина определяется либо форматом в описании поля, либо шириной отображаемого значения. Использование же пары символов '[' и ']' означает, что ширина поля ограничена пространством между этими двумя символами.

Символы '<' и '>' в начале поля указывают на выравнивание значения влево или вправо.

Вывод сформированного файла осуществляется внутренней интерфейсной программой.

- **В формате RTF** – в этом случае в качестве шаблона используется внешний файл в формате RTF, сформированный с помощью программы Microsoft WinWord. Правила описания шаблона в формате RTF описаны ниже.

Для отображения результата применяется Microsoft WinWord.

- **MSWord Formatter** – в этом случае в качестве шаблона используется внешний файл в формате DOT, сформированный с помощью программы Microsoft WinWord. Правила описания шаблона в формате DOT описаны ниже.

Для отображения результата применяется Microsoft WinWord.

Примеры описания текстовой части для шаблонов разных форматов приведены ниже.

- **Внешний браузер** – наличие данного признака указывает, что сформированный отчет выводится внешней, по отношению к Domino, программой. Выбор программы производится по расширению файла отчета.

Расширение	Программа
xls	Excel
doc	Word
rtf	Word
htm	Internet browser
html	Internet browser

- **Внешний файл шаблона** – имя файла, в котором находится текстовая часть шаблона.
- **Внешний файл отчета** – имя файла для сохранения результата отчета.
- **Наследовать текст из шаблона родителя** – изменение специального тега, применяемого для наследования содержимого блоков из шаблона-родителя.

По умолчанию используются следующая последовательность символов:
<НАСЛЕДОВАТЬ_ТЕКСТ_ИЗ_ШАБЛОНА_РОДИТЕЛЯ>.

Данный тег применяется для шаблонов в формате HTML и TEXT.

Переопределять тег не рекомендуется. Изменение допускается в том случае, если в тексте шаблона необходимо использование данной последовательности символов для других целей.

Правила описания текстовой части шаблона в формате RTF

Для создания шаблона используется программа WinWord. Шаблон сохраняется в файле формата RTF. При создании изображения шаблона в качестве меток полей и блоков используются стандартные закладки (bookmark) программы WinWord. Имена таких закладок-меток для блоков должны начинаться с символов BLOCK (например, BLOCKЗаголовок). Метка блока ставится непосредственно перед тем текстом, который должен быть включен в блок. Если в блок должна быть включена строка таблицы, то метка блока должна предшествовать ей, т.е. ставиться непосредственно после завершающей границы предыдущей строки (если такая есть).

В следующем примере метки блоков и полей, которые не видны при обычном просмотре, для наглядности помечены символами <bkmk>:

“ <bkmk>BLOCKЗаголовок

Документы со строками за период с <bkmk>Дата_начальная по
<bkmk>Дата_начальная

Колонка 1	Колонка 2	Сумма розн.	<bkmk>BLOCKДокумент
Документ №<bkmk>Код от <bkmk>Дата			<bkmk>BLOCKСтрока
<bkmk>Поле1	<bkmk>Поле2	<bkmk>Сумма	<bkmk>BLOCKИтог_по_докумен
Итого по Документу <bkmk>Сумма			<bkmk>BLOCKИтог

Всего <bkmk>Сумма

К блоку «Заголовок» относятся весь текст перед таблицей и верхняя строка таблицы. К блоку «Документ» - вторая строка таблицы. К блоку «Строка» - третья строка таблицы. К блоку «Итог по документу» - четвертая строка таблицы. К блоку «Итог» - весь текст под таблицей.

Так как закладки в документе WinWord не могут содержать пробелов, то вместо них используется символ подчеркивания: '_'. В структуре отчета не требуется менять названия полей и блоков, поскольку разбор и замена символов производится автоматически.

Правила описания текстовой части шаблона в формате DOT

Для создания шаблона используется программа WinWord. Шаблон сохраняется в файле формата DOT.

В отличие от формата RTF, для описания блоков и полей не используются стандартные закладки (bookmark) программы WinWord. В качестве меток применяются следующие символы:

- {{ - начало блока,
- }} - конец блока,
- { - начало поля,
- } - конец поля.

Метки в таких шаблонах не переопределяются.

Правила описания текстовой части шаблона в формате HTML

Шаблон в формате HTML представляет собой обычный текстовый файл, размеченный тегами HTML. Тег – это управляющий код, заключенный в угловые скобки <...>.

HTML файл отображается специальной программой – браузером HTML файлов.

Браузер распознает теги и формирует изображение текста в соответствии с указанной инструкцией. Для корректного отображения результата шаблон отчета должен быть подготовлен в соответствии с правилами языка HTML. Основные правила формирования текста в формате HTML приведены в Приложении 'HTML в Домино'.

Кроме тегов в шаблоне обязательно должны быть выделены блоки и поля. Метки блоков и полей задаются в соответствующих атрибутах ('Метка блока', 'Метка поля').

Базовые процедуры для работы с отчетами

В проекте имеются одна специальная функция и четыре процедуры для обработки шаблонов в скриптах-отчетах.

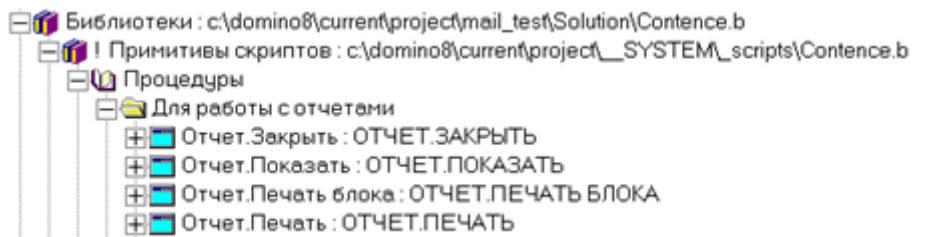
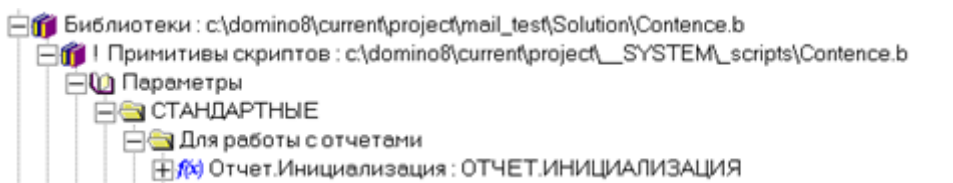


Схема выполнения отчета-скрипта для печати с шаблоном должна содержать следующую последовательность вызовов этих процедур:

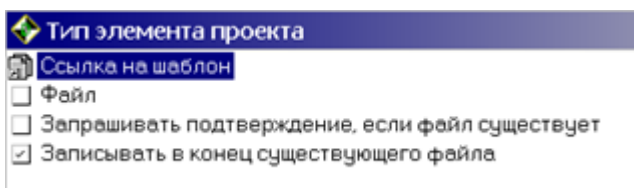
- Отчет.Инициализация
- Отчет.Печать блока – необходимое число раз
- Отчет.Показать
- Отчет.Закреть

При наследовании от базового отчета используется только функция '*Отчет.Печать блока*'.

Отчет.Инициализация

Выполняет подготовительные действия и создает файл, в который будет помещен результат выполнения скрипта-отчета.

Функция возвращает полное имя созданного файла и имеет следующие формальные параметры для вызова:



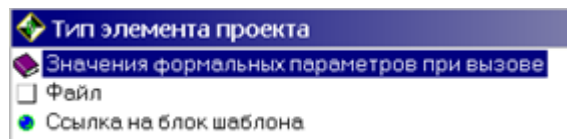
- **Ссылка на шаблон** – ссылка на проектный элемент, описывающий шаблон отчета. Можно указать выражение, возвращающее UID проектного элемента.
- **Файл** – имя файла или каталог на диске. Если каталог не указан, то файл создается в рабочем каталоге Domino (указан в ярлыке для запуска программы). Если параметр не указан, то применяется схема с автоматической генерацией уникальных имен файлов, но можно указать конкретное имя файла (атрибут '*Фиксированное имя*'). Обычно этот атрибут не задают.
- **Запрашивать подтверждение, если файл существует** – можно указать выражение. Если значение выражения равно 1, то на экране появится соответствующее сообщение. Параметр не обязательный.
- **Записывать в конец существующего файла** – если признак не будет установлен,

то старое содержимое файла будет удалено.

Отчет.Печать блока

Процедура печатает указанный блок. Данную процедуру необходимо вызвать хотя бы один раз, поскольку блок – это минимальная единица печати. Без печати блока никакие данные в результат отчета не попадут.

Процедура имеет следующие формальные параметры для вызова:

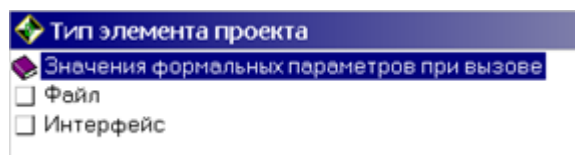


- **Значения формальных параметров при вызове** – не используется.
- **Файл** – полное имя файла, полученное в результате вызова функции Отчет.Инициализация.
- **Ссылка на блок шаблона** – ссылка на проектный элемент, описывающее блок. Можно указать выражение, возвращающее UID проектного элемента.

Отчет.Показать

Процедура выводит сформированный результат отчета на экран.

Процедура имеет следующие формальные параметры для вызова:



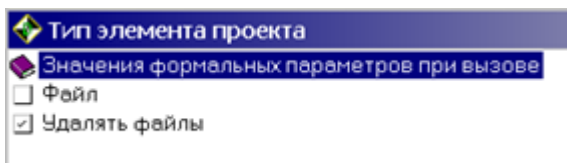
- **Значения формальных параметров при вызове** – не используется.
- **Файл** – полное имя файла, полученное в результате вызова функции Отчет.Инициализация.
- **Интерфейс** – вид шаблона отчета.

Процедура может отображать как закрытый файл (если требуется высветить уже имеющийся файл отчета), так и открытый (вызовом функции Отчет.Инициализация) файл.

Отчет.Заккрыть

Процедура закрывает файл отчета.

Процедура имеет следующие формальные параметры для вызова:



- **Значения формальных параметров при вызове** – не используется.
- **Файл** – полное имя файла, полученное в результате вызова функции `Отчет.Инициализация`.
- **Удалять файлы** – при установленном признаке созданный файл отчета удаляется.

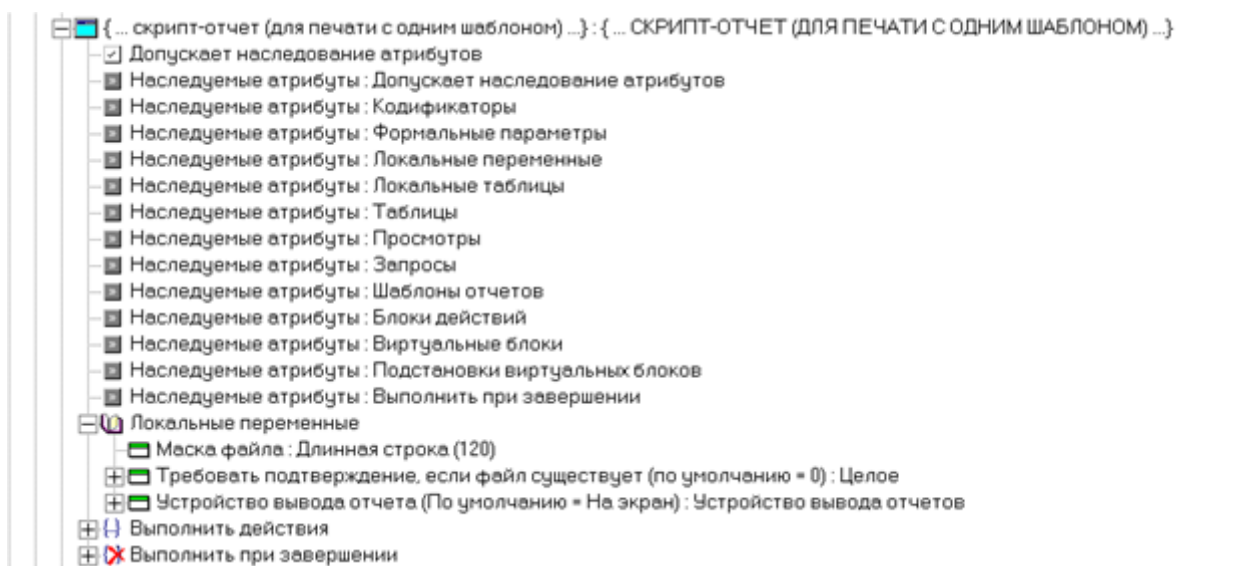
Рекомендуется удалять файлы при закрытии отчета.

Базовый 'Скрипт-отчет для печати с одним шаблоном'

Базовый скрипт-отчет создан для упрощения создания отчетов с одним шаблоном. В базовый отчет вынесены обязательные вызовы процедур для работы с шаблоном. В скрипте-наследнике необходимо лишь описать шаблон (со ссылкой на базовый шаблон) и из подстановки виртуального блока '*Печать данных*' вызвать функцию печати нужного блока.

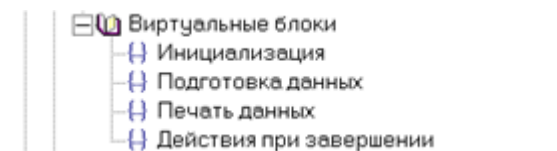
Поставленная цель достигается за счет того, что в скрипте-наследнике скрыта работа с функциями '`Отчет.Инициализация`', '`Отчет.Показать`' и '`Отчет.Закрыть`'. Разделы '*Выполнить действия*' и '*Выполнить при завершении*' для отчета-наследника просто недоступны. Зато доступны подстановки виртуальных блоков '*Инициализация*', '*Подготовка данных*', '*Печать данных*' и '*Действия при завершении*'. Из указанных блоков, в наследнике, обязательна только подстановка блока '*Печать данных*'.

Рассмотрим описание базового скрипта-отчета.

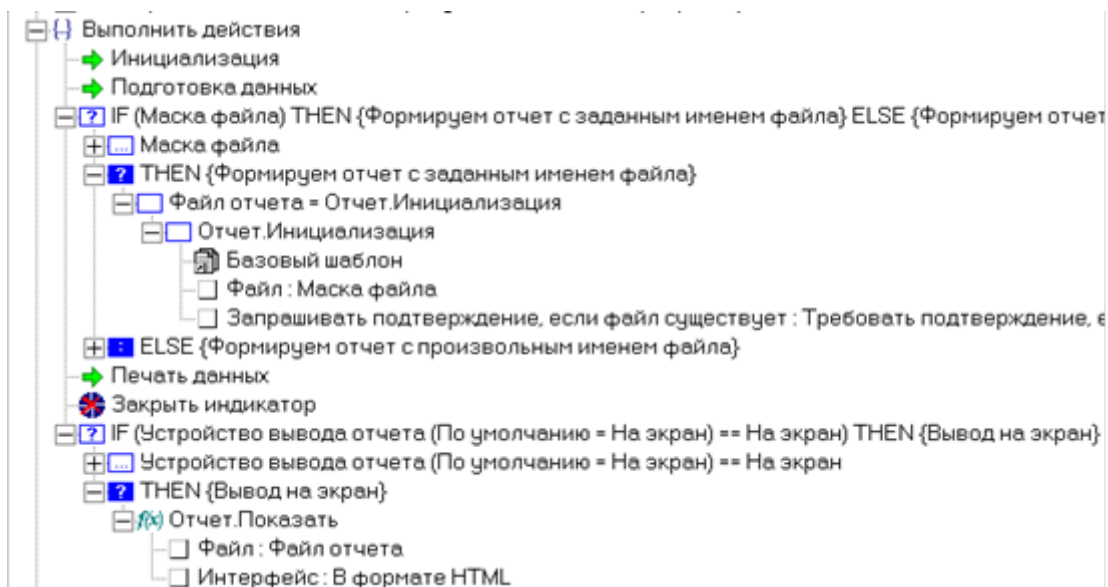


Описание скрипта содержит три раздела '*Локальные переменные*', '*Выполнить действия*', '*Выполнить при завершении*'.

От родителя достались определения следующих виртуальных блоков.



Разберем подробнее раздел *'Выполнить действия'*.



Алгоритм процедуры начинается с вызова двух виртуальных блоков *'Инициализация'* и *'Подготовка данных'*.

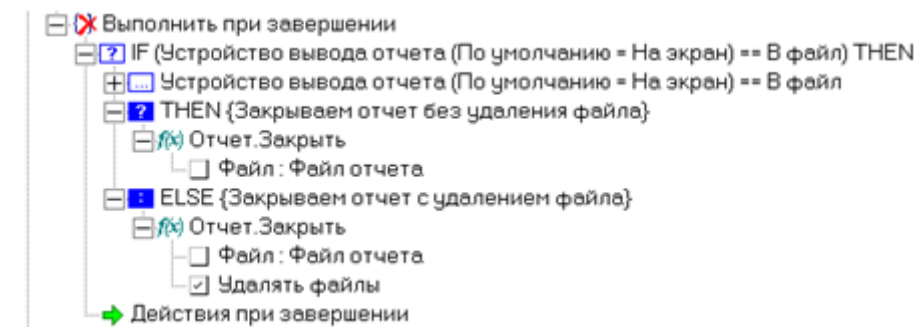
Далее, запускается функция *'Отчет.Инициализация'*, которая создает файл отчета. Если переменная *'Маска файла'* имеет значение, то файл создается с указанным именем. В противном случае имя файла автоматически генерируется программой.

В виртуальном блоке *'Печать данных'* должны находиться вызовы процедуры *'Отчет.Печать блока'*.

Вызов процедуры *'Отчет.Печать блока'* может находиться только в блоке *'Печать данных'*, поскольку блоки *'Инициализация'* и *'Подготовка данных'* вызываются до инициализации отчета, а блок *'Действия при завершении'* вызывается уже после закрытия отчета.

Если значение переменной *'Устройство вывода отчета'* равно *'На экран'*, то вызывается процедура *'Отчет.Показать'*.

Теперь посмотрим на блок *'Выполнить при завершении'*.

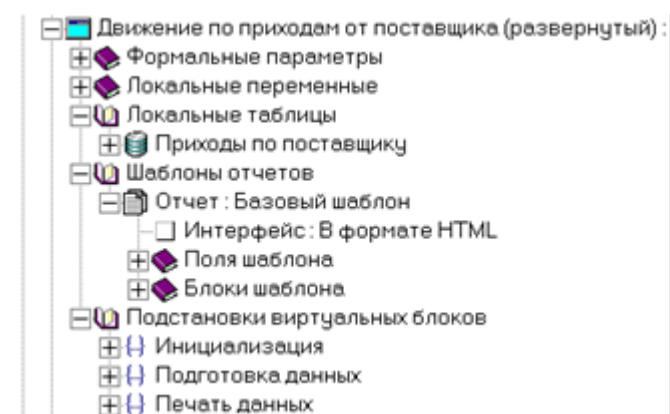


Вначале запускается процедура 'Отчет.Закрыть'. Если значение переменной 'Устройство вывода отчета' не равно 'В файл', то созданный файл отчета удаляется.

В самом конце вызывается виртуальный блок 'Действия при завершении'.

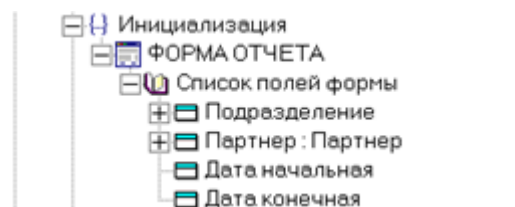
Как видим, в базовом скрипте-отчете не только отражена последовательность вызовов специальных процедур для работы с шаблоном, но также имеются точки (вызов виртуальных блоков), в которые проектировщики могут добавить дополнительные обработки данных.

Далее рассмотрим скрипт-отчет 'Движение по приходам от поставщика', который является наследником интересующего нас базового скрипт-отчета.



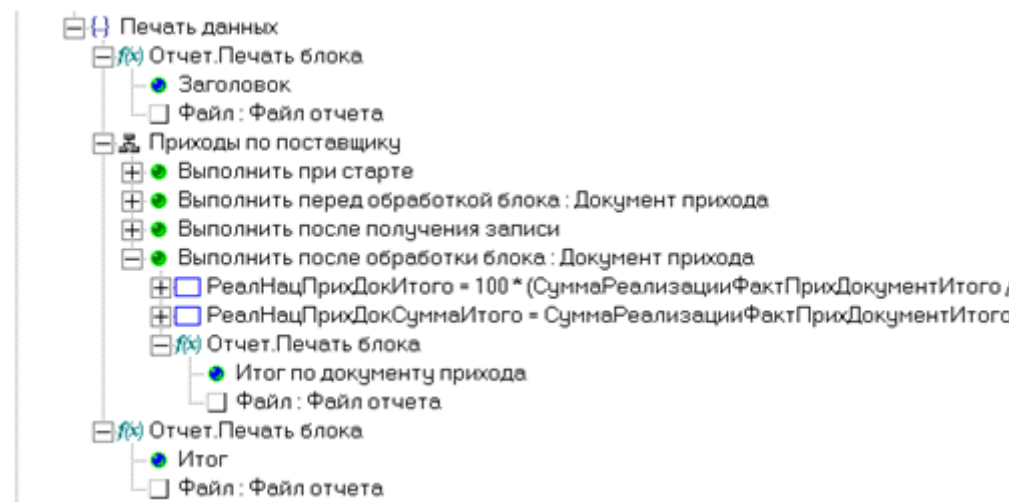
В данной процедуре определены три виртуальных блока.

В блоке 'Инициализация' заполняется форма с параметрами отчета.



В блоке 'Подготовка данных' заполняется локальная таблица 'Приходы по поставщику'.

Блок 'Печать данных' содержит вызовы процедуры 'Отчет.Печать блока' для различных блоков шаблона.



Автор данного отчета использовал не все возможности, предоставляемые базовым скрипт-отчетом. Не понадобился виртуальный блок '*Действия при завершении*'.

Применение базового скрипт-отчета значительно упростило написание процедуры. Текст процедуры стал короче и понятнее.

Версия #2

[Демонов Сергей](#) создал Fri, Apr 1, 2022 8:58 AM

[Демонов Сергей](#) обновил Fri, Apr 1, 2022 10:19 AM