Руководство пользователя FastReport .NET (WinForms, Mono, WPF, Avalonia, ASP.NET)

Версия 2025.2

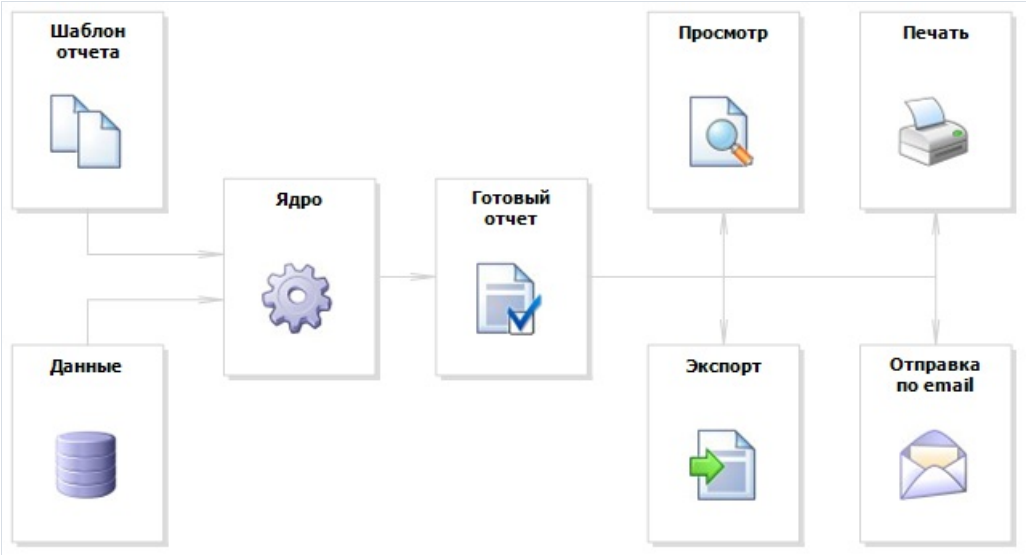
Основы

В этой главе будут рассмотрены принципы работы с отчетом в FastReport, а также подробно описаны основные элементы любого отчета:

- страницы отчета;
- бэнды;
- объекты отчета.

Отчет

Процесс работы с отчетом можно представить в следующем виде:



Шаблон отчета (далее – отчет) – это то, что мы видим в дизайнере. Отчеты хранятся в файлах с расширением .FRX. Отчет можно создать с помощью дизайнера или программным способом.

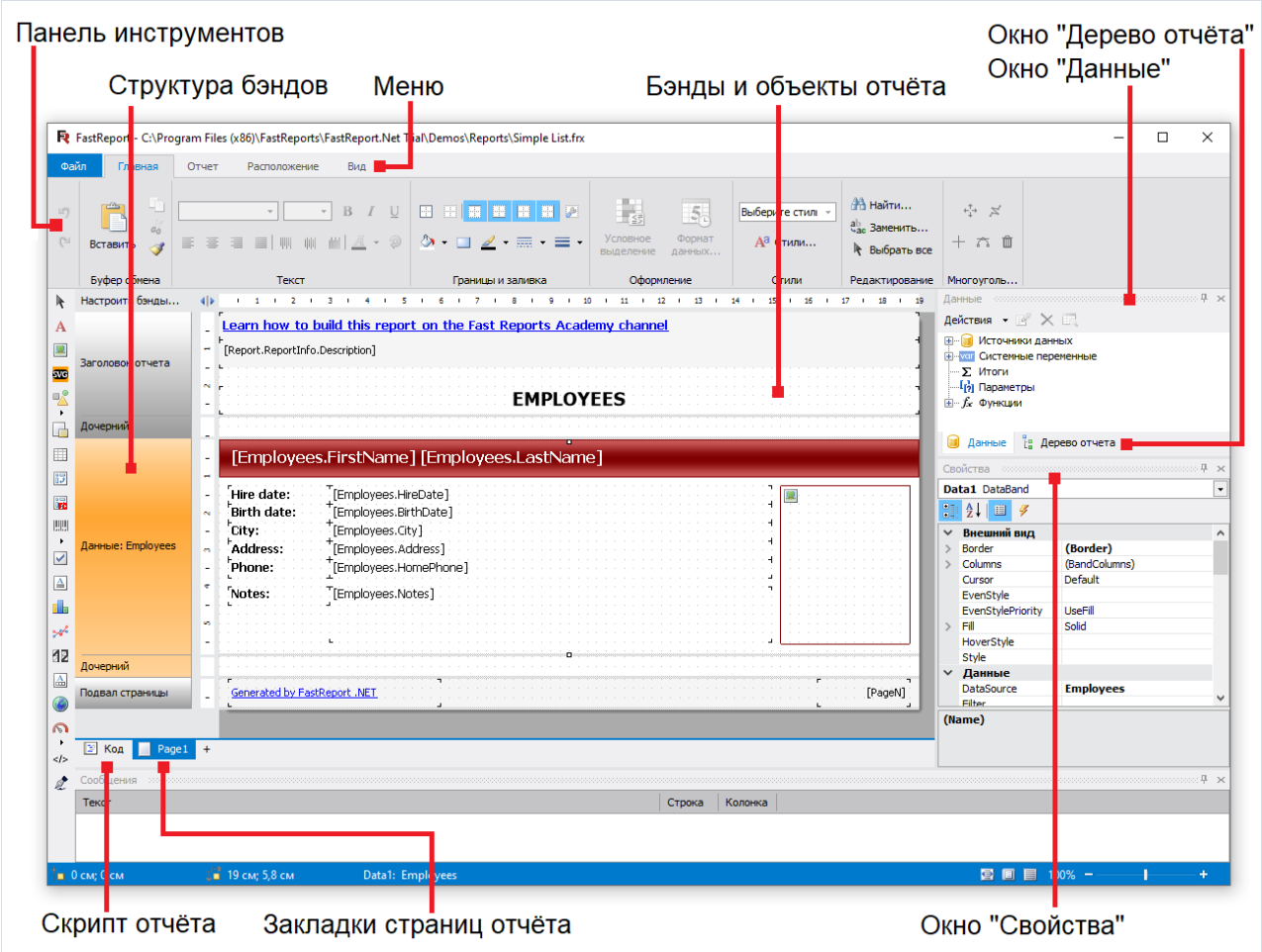
Данные могут быть любыми: это данные, определенные в программе, либо данные из СУБД, например, MS SQL. FastReport также может работать с объектами бизнес-логики (далее – бизнес-объекты).

Готовый отчет – это то, что мы видим в окне предварительного просмотра. Готовый отчет можно просмотреть, распечатать, сохранить в одном из поддерживаемых форматов (.doc, .xls, .pdf и прочих) или отправить по электронной почте.

Дизайнер отчетов

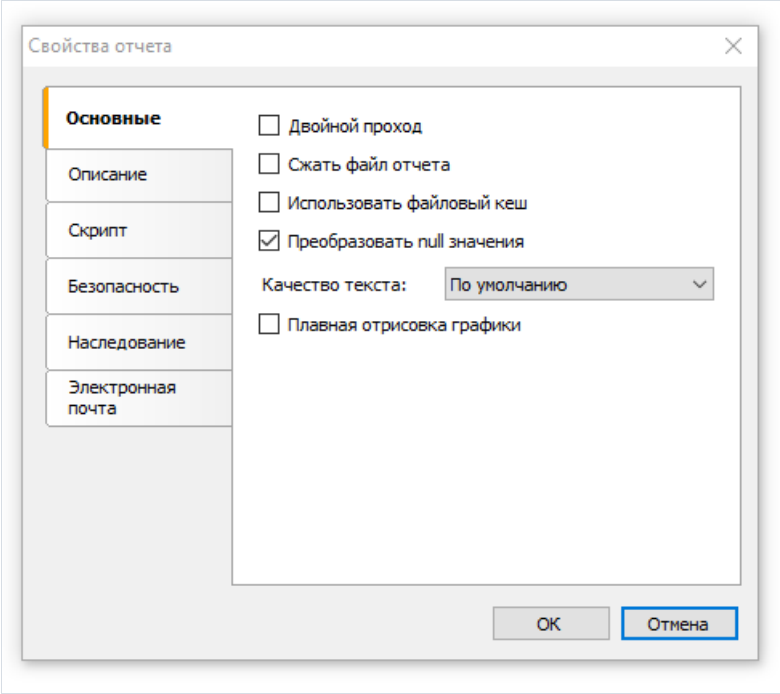
Для создания шаблона отчета используется дизайнер отчетов. Дизайнер предоставляет пользователю удобные средства для разработки внешнего вида отчета и позволяет сразу выполнить предварительный просмотр.

Дизайнер отчетов является составной частью FastReport и не зависит от среды разработки (например, MS Visual Studio). Если вы являетесь разработчиком приложений, вы можете использовать дизайнер в своих приложениях. Это даст возможность вашим пользователям самостоятельно изменять существующие или создавать новые отчеты.



Настройки отчета

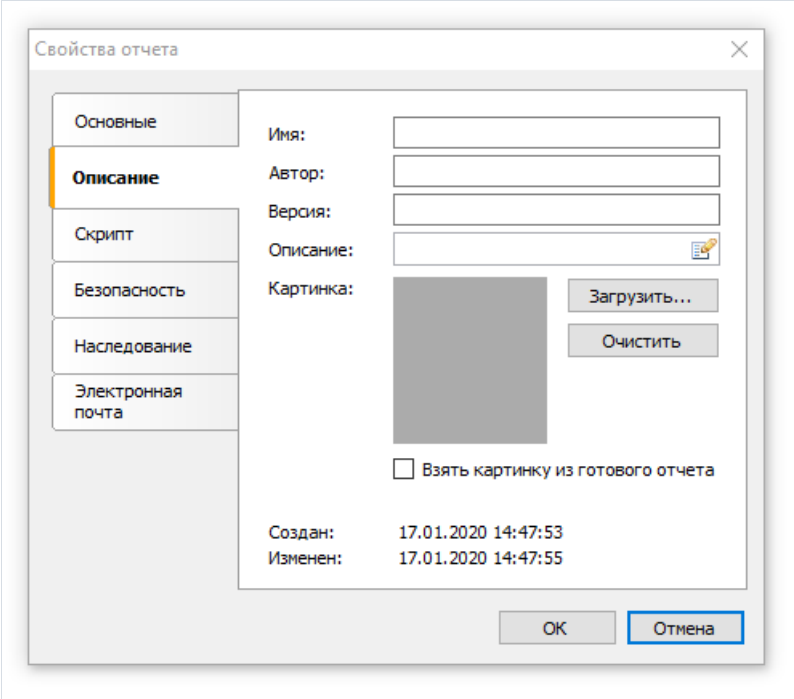
Окно с настройками отчета можно вызвать в меню "Отчет|Настройки...". Вы увидите диалоговое окно с несколькими закладками:



На закладке "Основные" вы можете управлять следующими параметрами отчета:

- параметр "Двойной проход" позволяет включить два прохода у отчета. Это может быть необходимо в случае, если вы печатаете в отчете информацию об общем количестве страниц;
- параметр "Сжать файл отчета" позволяет сохранить отчет в сжатом виде. Для сжатия используется алгоритм zip, поэтому вы легко сможете извлечь оригинальное содержимое с помощью любого архиватора;
- параметр "Использовать файловый кэш" позволяет сэкономить память при построении отчета. Используйте этот параметр, если ваш отчет содержит большое количество страниц;
- параметр "Преобразовывать null значения" управляет преобразованием null значений полей данных в значения по умолчанию (0, пустая строка, `false` в зависимости от типа данных поля);
- параметр "Качество текста" позволяет выбрать режим отображения текста в отчете. Этот режим не влияет на печать отчета;
- параметр "Плавная отрисовка графики" включает режим сглаживания при отрисовке графических объектов (линий, рамок, рисунков).

На закладке "Описание" вы можете задать описание отчета. Все эти параметры являются необязательными и служат для информационных целей:



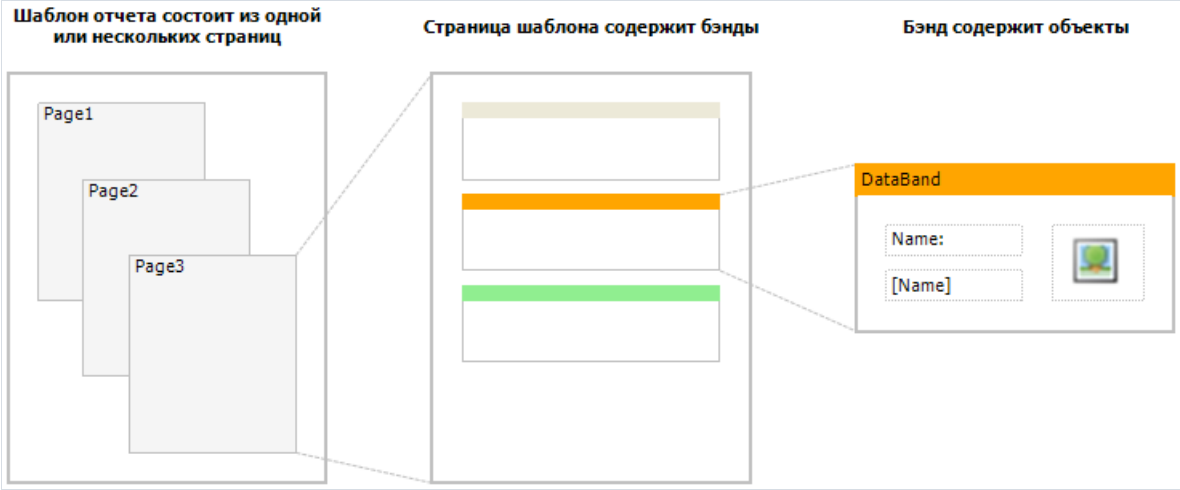
На закладке "Скрипт" можно выбрать язык скрипта для текущего отчета. Подробнее работа со скриптом рассматривается в главе "Скрипт".

На закладке "Безопасность" вы можете задать пароль, который будет запрошен при открытии отчета. Отчет с паролем хранится в зашифрованном виде, поэтому не забывайте введенный пароль! Восстановить отчет в этом случае будет практически невозможно.

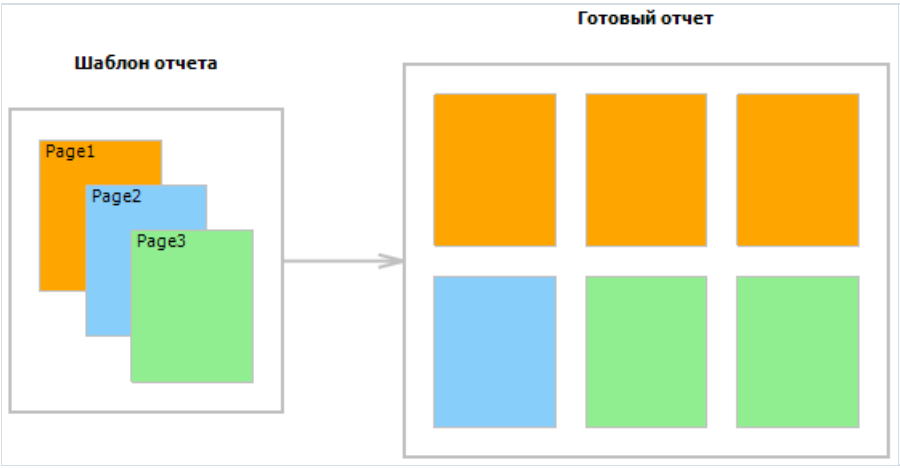
На закладке "Наследование" можно управлять наследованием отчета. Эта функциональность будет рассмотрена позже.

Страницы отчета

Отчет состоит из одной (чаще всего) или нескольких страниц. Страница отчета, в свою очередь, содержит бэнды (англ. band – полоска). На бэндах располагаются объекты отчета, такие как "Текст", "Рисунок" и прочие.



Особенность FastReport – шаблон отчета может состоять из нескольких страниц. Например, вы можете создать шаблон, содержащий титульную страницу и страницу с данными. При построении такого отчета сначала будет напечатана первая страница, затем вторая и т.д. Каждая страница шаблона может генерировать одну или несколько страниц готового отчета – это зависит от содержащихся на ней данных:

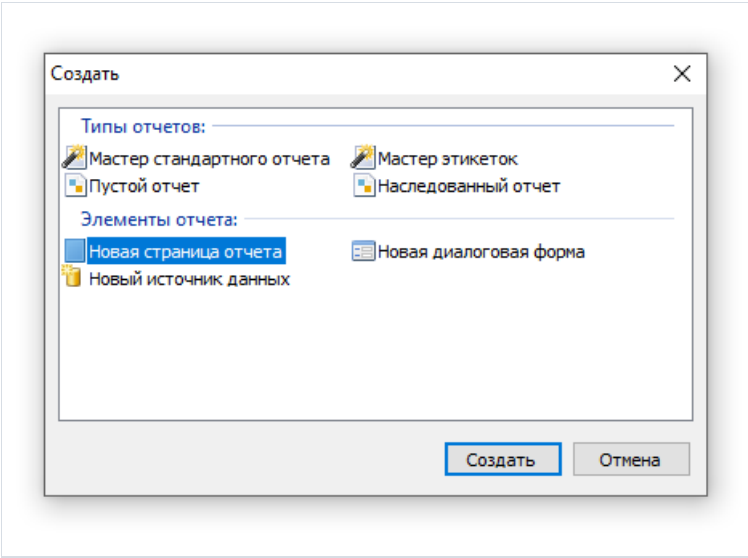


Страницы отчета также используются при работе с вложенными отчетами (subreport). В отличие от других генераторов отчетов, вложенные отчеты в FastReport хранятся на отдельной странице шаблона, а не в отдельном файле.

Кроме страниц отчета, шаблон может содержать одну или несколько диалоговых форм. Диалоговые формы могут быть использованы для запроса параметров перед построением отчета. Подробнее работа с диалогами будет рассмотрена в главе "Диалоговые формы".

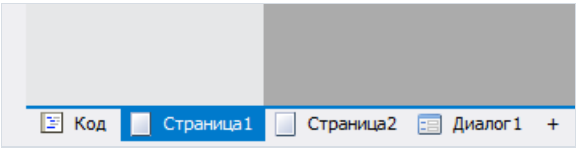
Управление страницами

Когда вы создали новый отчет, он уже содержит одну страницу с несколькими бэндами. Для добавления новой страницы нажмите кнопку  на панели инструментов. Страницу можно также добавить, нажав кнопку "Новый..." и выбрав в окне пункт "Новая страница отчета":



Аналогичным образом в отчет добавляется диалоговая форма. Для этого используйте кнопку  .

Страницы шаблона отображаются в дизайнере в виде закладок:



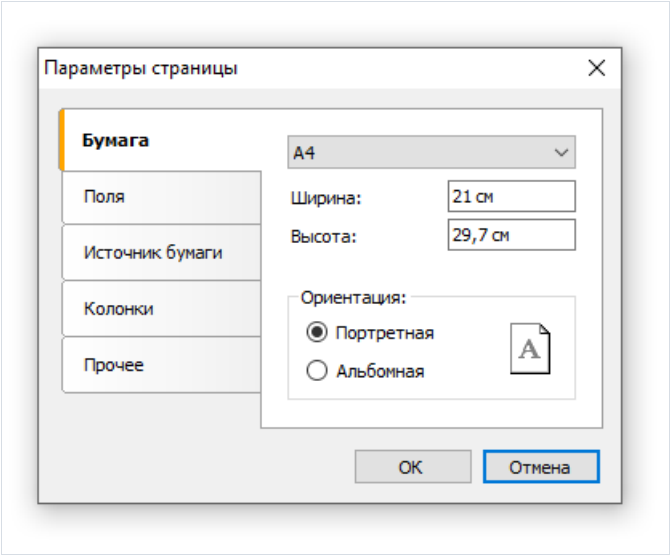
> Самая первая закладка – это код отчета. Она не может быть перемещена или удалена.

Чтобы переключиться на нужную страницу, просто щелкните мышкой на ее закладке. Поменять порядок страниц можно с помощью мыши. Для этого нажмите левой кнопкой мыши на закладке и, не отпуская кнопки, переместите закладку на желаемое место.

Для удаления страницы нажмите кнопку  . Эта кнопка неактивна, если отчет состоит только из одной страницы.

Свойства страницы

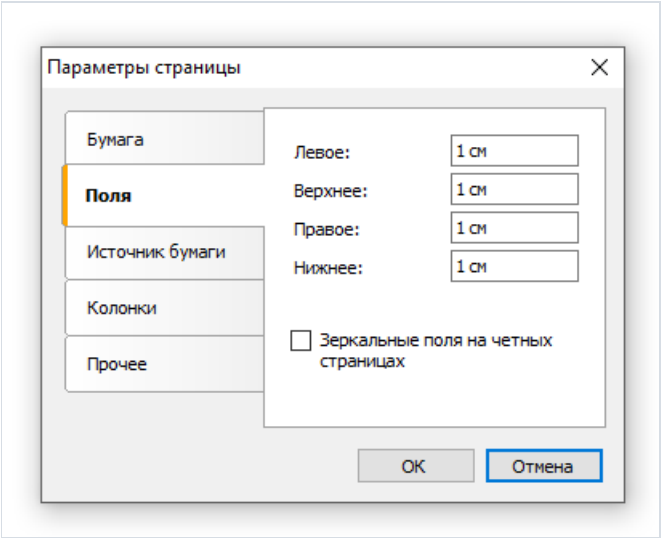
Каждая страница отчета может иметь свои настройки, такие как размер бумаги, ориентация (альбомная или портретная), отступы, колонки, источник бумаги и прочие. Шаблон отчета может содержать несколько страниц с разной ориентацией и размерами бумаги. Окно с настройками страницы можно вызвать кнопкой  или выбрав в меню пункт "Файл|Параметры страницы...". Окно содержит несколько групп настроек, которые можно выбирать в списке слева:



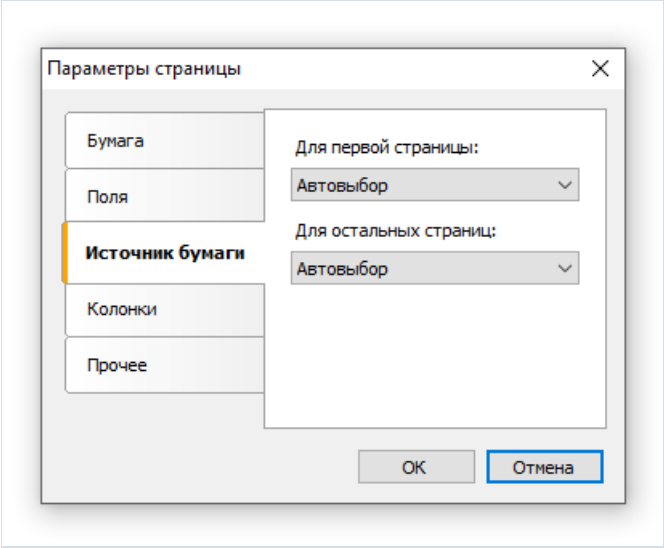
Группа "Бумага" позволяет задать размер и ориентацию листа бумаги. Можно выбрать одно из значений размера, используя выпадающий список. Он содержит все размеры бумаги, которые поддерживаются текущим принтером.

> Текущий принтер можно настроить, вызвав меню "Файл|Параметры принтера...".

Группа "Поля" позволяет задать поля страницы. Здесь же можно указать, что левое и правое поле должны меняться местами для четных страниц готового отчета:

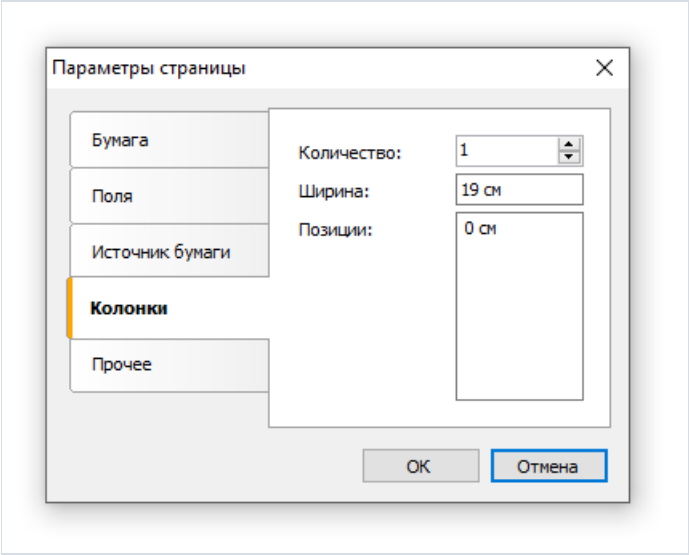


Группа "Источник бумаги" позволяет выбрать источник бумаги. Заметьте, что источник можно задать отдельно для первой страницы готового отчета и для остальных страниц:

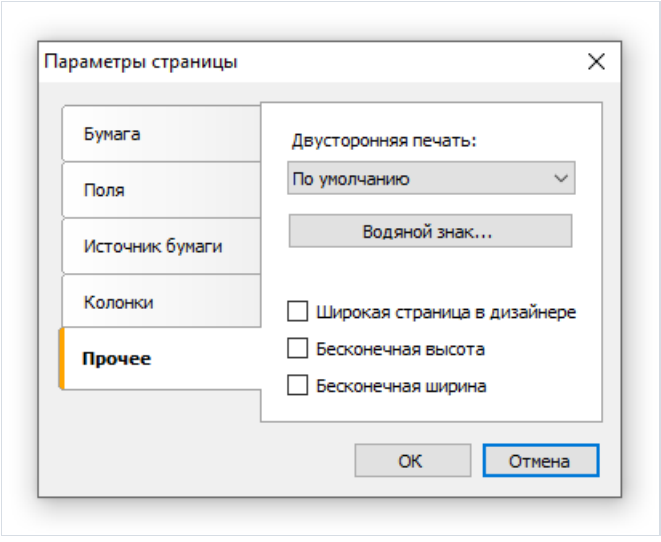


> Источник бумаги можно выбрать непосредственно перед печатью, в окне "Печать".

Группа "Колонки" позволяет задать параметры колонок для многоколоночного отчета. Для этого надо указать число колонок и (необязательно) поправить ширину колонки и позицию каждой из колонок:



Группа "Прочее" позволяет задать некоторые вспомогательные свойства страницы. Можно указать режим дуплекса для двусторонней печати, если ваш принтер поддерживает такой режим. Также здесь можно настроить водяной знак, который будет печататься на страницах готового отчета:



Флажок "Широкая страница в дизайнере" позволяет увеличить ширину страницы в режиме дизайнера. Это может пригодиться, если вы работаете с объектами, которые могут расти в ширину и разбиваться на страницы. Это объекты "Таблица" и "Матрица".

> Режим двусторонней печати можно выбрать непосредственно перед печатью, в окне "Печать".

Бэнды

Бэнд (англ. band – полоска) – это объект, который размещается непосредственно на странице отчета и является контейнером для остальных объектов, таких, как "Текст", "Рисунок" и прочих.

Всего в FastReport есть 13 типов бэндов. В зависимости от своего типа, бэнд печатается в определенном месте отчета:

Бэнд	Как печатается
Заголовок отчета	Печатается один раз в самом начале отчета. Вы можете выбрать порядок печати – перед бэндом "Заголовок страницы" или после него – с помощью свойства страницы `TitleBeforeHeader`. Изменить это свойство можно с помощью служебного окна "Свойства". По умолчанию свойство равно `true`, т.е. заголовок отчета печатается перед заголовком страницы.
Подвал отчета	Печатается один раз в конце отчета, после последней строки данных, но перед бэндом "Подвал страницы".
Заголовок страницы	Печатается вверху на каждой странице отчета.
Подвал страницы	Печатается внизу на каждой странице отчета.
Заголовок колонки	Этот бэнд используется при печати многоколоночного отчета (когда в настройках страницы указано количество колонок > 1). Он печатается вверху каждой колонки, после заголовка страницы.
Подвал колонки	Печатается внизу каждой колонки, перед подвалом страницы.
Данные	Этот бэнд подключается к источнику данных и печатается столько раз, сколько строк в источнике.
Заголовок данных	Этот бэнд подключается к бэнду "Данные" и печатается перед первой строкой данных.
Подвал данных	Этот бэнд подключается к бэнду "Данные" и печатается после последней строки данных.
Заголовок группы	Печатается в начале каждой группы, когда значение условия группировки меняется.
Подвал группы	Печатается в конце каждой группы.
Дочерний	Этот бэнд может быть подключен к любому бэнду, в том числе другому дочернему бэнду. Он печатается сразу после своего родителя.
Фоновый	Печатается в виде фона на каждой странице отчета.

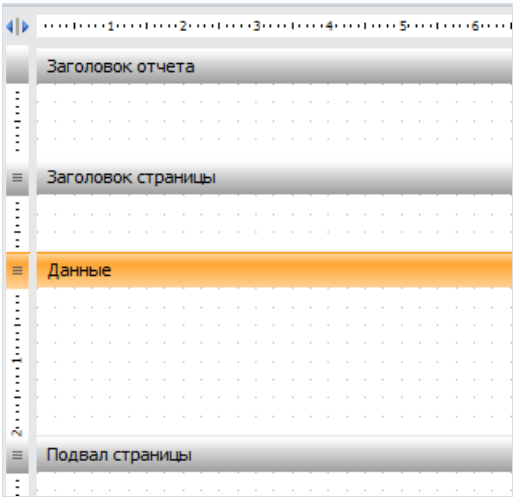
Отображение бэндов в дизайнере

Бэнд в дизайнере отображается в виде прямоугольной области:

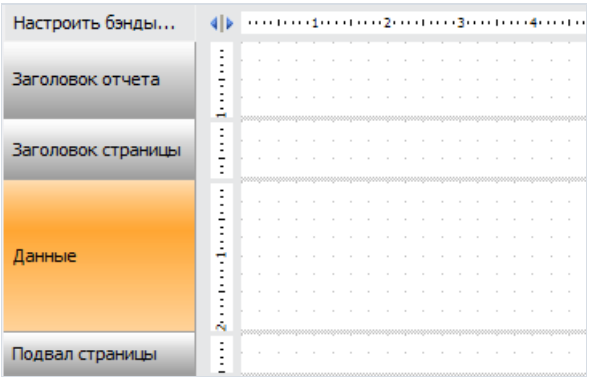
Бэнд, как и многие другие объекты отчета, может иметь рамку и заливку (по умолчанию они отключены). Кроме этого, бэнд отображает сетку. Для настройки вида сетки зайдите в меню "Вид|Настройки..." и выберите "Страница отчета". В группе "Сетка" можно задать видимость и тип сетки, а также расстояние между ее узлами. Сетку можно также включить или выключить в меню "Вид".

Если подвести курсор мыши к нижней границе бэнда, то курсор меняет вид – теперь можно менять высоту бэнда. Бэнд предварительно выбирать не обязательно. Также менять высоту бэнда можно перемещая с помощью мыши ограничитель бэнда на левой линейке.

Дизайнер имеет два режима отображения бэндов, между которыми можно переключиться в любой момент. Это можно сделать, нажав кнопку  на пересечении линеек или в меню "Вид|Настройки..." выбрать "Страница отчета" и в группе "Вид бэндов" выбрать нужный вид. В первом режиме каждый бэнд имеет заголовок, содержащий название бэнда и полезную информацию о нем (например, имя источника данных, к которому он подключен).



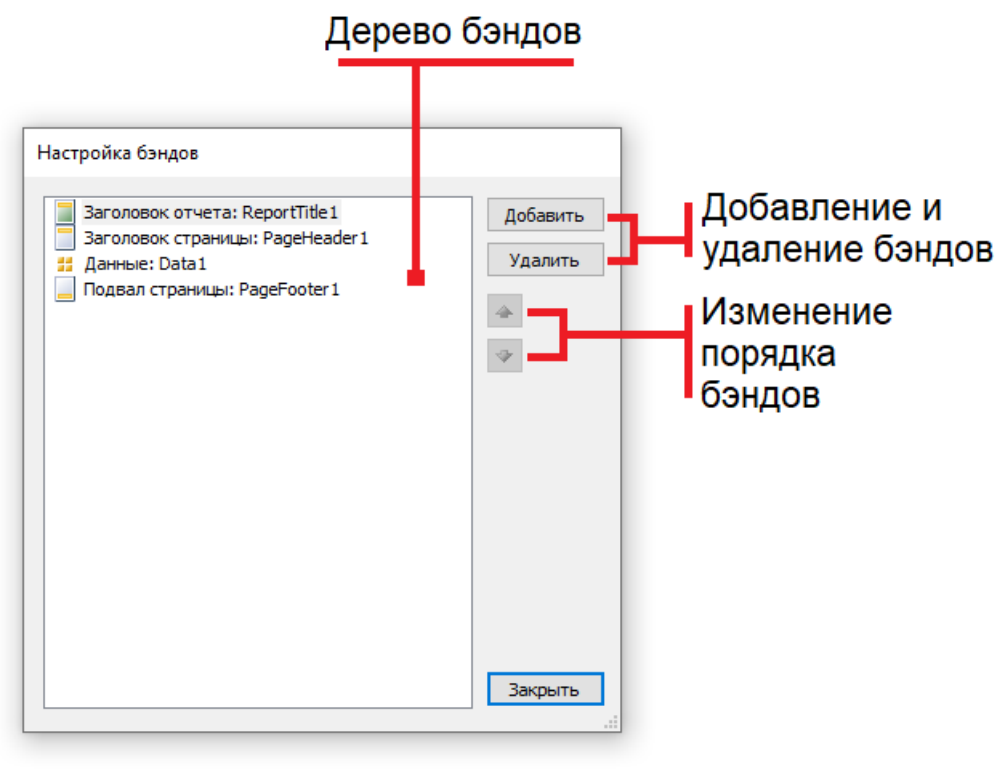
Во втором режиме бэнды заголовков не имеют. Вместо этого в левой части окна отображается структура бэндов. Этот режим позволяет легко разобраться в структуре отчета, особенно если его создавали не вы. Бэнды отображаются максимально компактно.



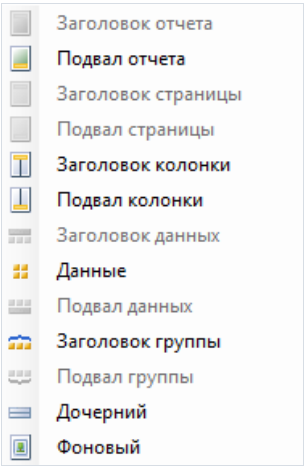
У этого режима есть один недостаток – ширины экрана может не хватить для отображения бэнда целиком. В этом случае можно свернуть служебные окна, либо переключиться на первый режим.

Настройка бэндов

Добавление и удаление бэндов выполняется в окне "Настройка бэндов". Его можно вызвать из меню "Отчет| Настроить бэнды...", либо с помощью кнопки "Настроить бэнды...", расположенной над деревом бэндов:

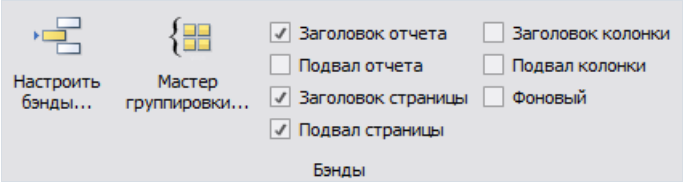


В этом окне можно добавить в отчет бэнды, удалить их или изменить их порядок. Для добавления бэнда нажмите кнопку "Добавить" или щелкните правой кнопкой мыши на дереве бэндов. Появится контекстное меню, содержащее список бэндов. Бэнды, которые не могут быть добавлены, отображаются серым цветом:



Операция добавления зависит от того, какой бэнд выбран в дереве бэндов. Например, добавить бэнды "Заголовок данных" и "Подвал данных" можно лишь в том случае, если выбран бэнд "Данные".

Существует и другой способ настроить бэнды. Это можно сделать из меню "Отчет":



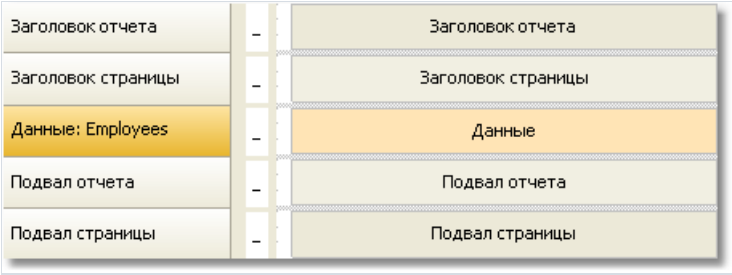
Как видно, можно включить или выключить бэнды – заголовки и подвалы отчета, страницы, колонки.

Наконец, удалить бэнд можно, если выделить его в окне дизайнера и нажать клавишу `Delete`.

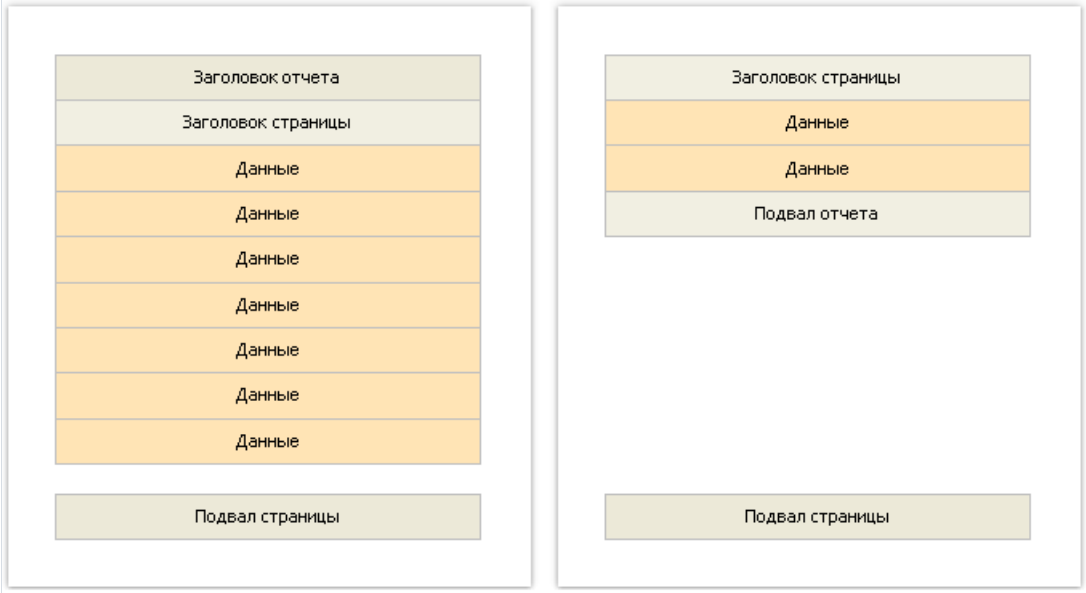
> При настройке бэндов FastReport не позволит выполнить действия, которые приведут к созданию неправильного шаблона отчета. Например, вы не можете удалить бэнд "Данные", который присоединен к группе (бэнд "Заголовок группы") – для этого сначала надо удалить группу. Другой пример – при удалении бэнда "Данные" автоматически удаляются его заголовок и подвал. Также нельзя удалить бэнд, если он единственный на странице.

Порядок печати бэндов

Итак, на странице расположено несколько бэндов. Как FastReport будет формировать готовый отчет? Рассмотрим следующий пример.



Сначала будет напечатан заголовок отчета. Сразу за ним – заголовок страницы. Далее будет печататься бэнд "Данные". Он будет напечатан столько раз, сколько строк в источнике данных, к которому подключен бэнд. После того, как все строки бэнда "Данные" напечатаны, печатается бэнд "Подвал отчета" и внизу страницы – бэнд "Подвал страницы". На этом печать отчета заканчивается. Готовый отчет будет выглядеть примерно так:



В процессе печати FastReport проверяет, достаточно ли места на текущей странице готового отчета, чтобы напечатать бэнд. Если места нет, происходит следующее:

- печатается подвал страницы;
- добавляется новая страница;
- печатается заголовок страницы;
- продолжается печать бэнда, который не поместился на предыдущей странице.

Свойства бэндов

Каждый бэнд имеет несколько полезных свойств, влияющих на процесс печати. Настроить их можно, используя контекстное меню бэнда. Для этого щелкните правой кнопкой мыши на пустом месте бэнда, не занятом другими объектами. Также можно щелкнуть на заголовке бэнда (если включен классический режим отображения) или на структуре бэндов (в противном случае). Другой способ – выбрать бэнд и поменять соответствующее свойство в окне "Свойства".

Свойство	Описание
"Может расти", "Может сжиматься" (CanGrow, CanShrink)	Свойства определяют, может ли бэнд расти или сжиматься в зависимости от размера объектов, которые на нем расположены. Подробнее об использовании этого свойства можно почитать в главе "Построение отчетов".
"Может разрываться" (CanBreak)	Если это свойство включено, и на странице недостаточно места для печати бэнда, делается попытка напечатать часть содержимого бэнда на имеющемся месте, т.е. "разорвать" его. Подробнее об использовании этого свойства можно почитать в главе "Построение отчетов".
"Формировать новую страницу" (StartNewPage)	Печать бэнда с таким свойством начинается с новой страницы. Обычно это свойство используется при печати групп; при этом каждая группа печатается на новой странице.
"Печатать внизу страницы" (PrintOnBottom)	Бэнд с таким свойством печатается в самом низу страницы, перед бэндом "Подвал страницы". Это может оказаться полезным при печати некоторых документов, где необходимо итоговую сумму печатать внизу страницы.
"Повторять на каждой странице" (RepeatOnEveryPage)	Это свойство имеется у бэндов - заголовков и подвалов данных и групп. Такой бэнд будет напечатан вверху каждой новой страницы, пока идет печать данных. Подробнее об использовании этого свойства можно почитать в главе "Построение отчетов".

Объекты отчета

В отчете можно использовать широкий набор объектов:

Иконка	Название	Описание
	"Текст" (TextObject)	Показывает одну или несколько строк текста.
	"Рисунок" (PictureObject)	Показывает рисунок.
	"SVG" (SVGObject)	Используется для отображения масштабируемых векторных изображений в отчетах без потери качества.
	"Линия" (LineObject)	Показывает линию. Линия может быть вертикальной, горизонтальной или диагональной.
	"Фигура" (ShapeObject)	Показывает одну из геометрических фигур – прямоугольник, эллипс, треугольник и др.
	"Ломаная линия" (PolyLineObject)	Показывает ломаную линию с различным количеством вершин и поддержкой кривизны отрезков.
	"Многоугольник" (PolygonObject)	Показывает многоугольник с различным количеством вершин и поддержкой кривизны отрезков.
	"Форматированный текст" (RichObject)	Показывает форматированный текст (в формате RTF).
	"Штрих-код" (BarcodeObject)	Показывает штрих-код.
	"Флажок" (CheckBoxObject)	Показывает флажок, который может иметь два состояния – "включен" или "выключен".
	"Таблица" (TableObject)	Показывает таблицу, состоящую из строк, колонок и ячеек.
	"Матрица" (MatrixObject)	Показывает матрицу (также известную под названиями "сводная таблица", "кросс-таб").
	"Диаграмма" (MSChartObject)	Показывает диаграмму.
	"Почтовый индекс" (ZipCodeObject)	Показывает почтовый индекс.
	"Текст в ячейках" (CellularTextObject)	Показывает текст, каждый символ которого печатается в отдельной ячейке.
	"Цифровая подпись" (DigitalSignatureObject)	Показывает видимое поле, которое может быть использовано для электронной подписи.

Иконка	Название	Описание
	"HTML" (HtmlObject)	Позволяет использовать язык разметки HTML с множеством тэгов.
	"Вложенный отчет" (SubreportObject)	Предоставляет возможность отобразить другой отчет в текущем.
	"Контейнер" (ContainerObject)	Используется для размещения групп других объектов.
	"Карта" (MapObject)	Отображает компонент карты.
	"Линейный датчик" (LinearGauge)	Это графический компонент, который используется для отображения прогресса, показателей или состояния в виде линейного датчика.
	"Простой датчик" (SimpleGauge)	Это графический компонент, который используется для отображения прогресса, показателей или состояния в виде простого датчика.
	"Круговой датчик" (RadialGauge)	Это графический компонент, который используется для отображения прогресса, показателей или состояния в виде кругового датчика.
	"Простой датчик прогресса" (SimpleProgressGauge)	Это графический компонент, который используется для отображения прогресса, показателей или состояния в виде простого индикатора прогресса.

Объекты можно использовать как для отображения информации (объект "Текст"), так и для оформления отчета (объекты "Рисунок", "Линия", "Фигура"). Сложные объекты типа "Таблица" и "Матрица" могут содержать в себе другие простые объекты.

Общие свойства объектов

Все объекты отчета наследуются от одного базового класса (`ReportComponentBase`) и имеют некоторый общий набор свойств. Прежде чем изучать каждый объект, рассмотрим эти свойства.

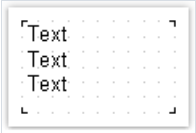
Вы можете менять значение свойств с помощью окна "Свойства". Там, где это возможно (например, при работе с рамкой и заливкой), лучше использовать панели инструментов или контекстное меню объекта.

Свойство	Описание
Координаты и размеры (Left, Top, Width, Height)	Объект отчета в FastReport представляет собой прямоугольник. У него есть координаты (свойства `Left` , `Top`) и размеры (свойства `Width` , `Height`).
"Якорь" (Anchor)	Это свойство определяет, как будет изменяться позиция объекта и/или его размеры при изменении размеров контейнера, на котором он лежит. Используя якорь, можно сделать так, чтобы объект расширялся или сдвигался синхронно с контейнером. Подробнее об использовании этого свойства можно прочитать в разделе "Автоматический подбор высоты объектов".
"Стыковка" (Dock)	Это свойство определяет, к какой стороне контейнера будет пристыкован объект. Подробнее об использовании этого свойства можно прочитать в разделе "Автоматический подбор высоты объектов".
Рамка и заливка (Border, Fill)	Эти свойства содержат настройки рамки и заливки соответственно. Их можно менять, используя панели инструментов.
"Может расти", "Может сжиматься" (CanGrow, CanShrink)	Эти свойства позволяют подбирать высоту объекта таким образом, чтобы он вмещал весь текст. Подробнее об использовании этих свойств можно прочитать в разделе "Автоматический подбор высоты объектов".
"Сдвиг" (ShiftMode)	Объект, у которого свойство включено, будет сдвинут вниз или вверх, если над ним имеется объект, который расширяется или сжимается. Подробнее об использовании этого свойства можно прочитать в разделе "Автоматический подбор высоты объектов".
"Расти вниз" (GrowToBottom)	Объект с таким свойством при печати растягивается до нижней границы бэнда. Подробнее об использовании этого свойства можно прочитать в разделе "Автоматический подбор высоты объектов".
"Может разрываться" (CanBreak)	Это свойство есть у текстовых объектов – "Текст" и "Форматированный текст". Оно определяет, может ли содержимое объекта разрываться на части. Это может произойти, если у бэнда, на котором лежит объект, включено аналогичное свойство.
"Печатать на..." (PrintOn)	Это свойство определяет, на каких страницах может быть напечатан объект. Подробнее об использовании этого свойства можно прочитать в разделе "Многостраничный отчет типа "Буклет".
"Курсор" (Cursor)	Это свойство определяет вид указателя мыши, когда он находится над объектом. Свойство работает только в окне предварительного просмотра.

Свойство	Описание
"Видимый" (Visible)	Свойство определяет, будет ли объект отображаться в отчете. Невидимый объект не отображается в окне предварительного просмотра и не печатается на принтере.
"Печатаемый" (Printable)	Свойство определяет, будет ли объект печататься на принтере. Если это свойство отключено, объект будет виден в окне предварительного просмотра, но не попадет на распечатку.
"Гиперссылка" (Hyperlink)	Это свойство позволяет сделать объект отчета интерактивным. Подробно работа с гиперссылками будет рассмотрена в разделе "Интерактивные отчеты".
"Закладка" (Bookmark)	Это свойство используется совместно со свойством "Гиперссылка". Оно может содержать любое выражение. Выражение будет вычислено при работе отчета, и его значение будет использовано в качестве имени закладки.
"Ограничения" (Restrictions)	Это свойство позволяет задавать ограничения на некоторые операции над объектом. По умолчанию свойство пустое, т.е. над объектом можно совершать все операции.
"Стиль" (Style)	Данному свойству можно присвоить имя стиля. При этом объект станет выглядеть так, как указано в стиле. Если параметры стиля поменяются, внешний вид объекта также изменится.

Объект "Текст"

Объект "Текст" является основным объектом, с которым вам придется работать чаще всего. Он выглядит следующим образом:

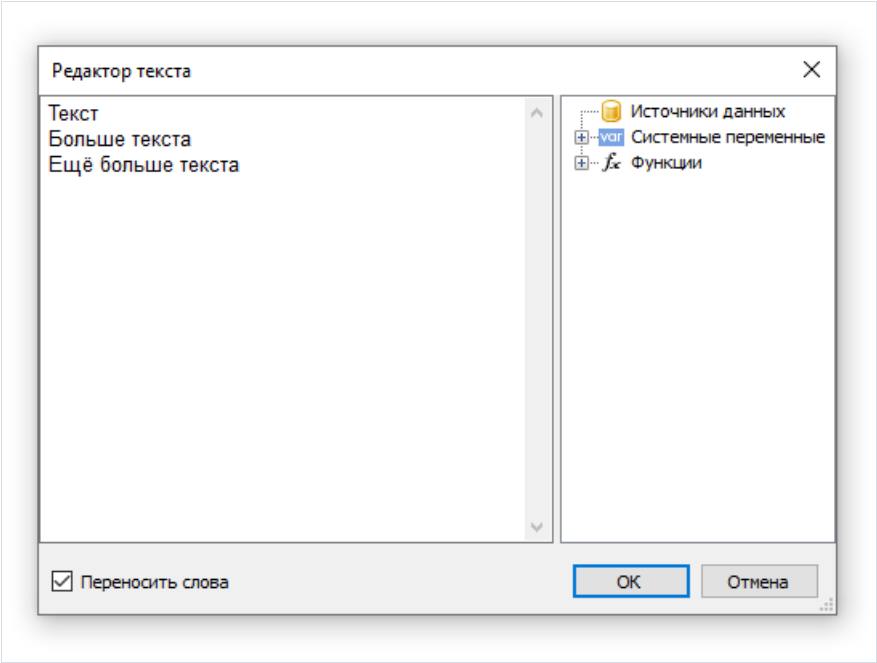


Объект может отображать любые текстовые данные, а именно:

- одну или несколько строк текста;
- поля источников данных;
- системные переменные типа "Номер страницы";
- параметры отчета;
- итоговые значения;
- выражения;
- любую комбинацию из вышеперечисленных элементов.

Редактирование текста

Для редактирования текста объекта сделайте на нем двойной щелчок мыши. Вы увидите редактор текста:



В правой части редактора находится окно данных, которые можно добавлять в текст. Это можно сделать, перетащив (drag&drop) элемент мышкой в нужное место текста. Другой способ вставки элемента в текст – сделайте двойной щелчок на элементе, и он будет добавлен в текущую позицию курсора.

Чтобы сохранить изменения и закрыть окно редактора, нажмите кнопку OK, либо комбинацию клавиш `Ctrl+Enter`.

Другой способ редактирования текста – редактирование на месте (прямо на странице отчета). Для этого выделите объект "Текст" и нажмите клавишу `Enter`. Для окончания редактирования щелкните мышкой за пределами объекта либо нажмите комбинацию клавиш `Ctrl+Enter`. Клавиша `Esc` отменяет изменения.

> Во время редактирования объекта на месте можно изменять его размеры мышкой.

Отображение выражений

Объект "Текст" может содержать как обычный текст, так и выражения. Причем выражения могут содержаться в объекте вперемешку с текстом. Например:

```
Сегодня [Date]
```

При печати такого объекта все выражения, содержащиеся в тексте, будут вычислены. В текст будет подставлено значение выражения, например:

```
Сегодня 12 сентября 2010г.
```

Как видно, выражения обозначаются с помощью квадратных скобок. Это настраивается в свойстве `Brackets`, которое по умолчанию содержит строку `[,]`. При необходимости вы можете указать другую пару символов, например `<,>` или `<!,!>`. В последнем случае выражения в тексте будут выглядеть так:

```
Сегодня <!Date!>
```

Кроме того, можно запретить обработку выражений – за это отвечает свойство `AllowExpressions`. В этом случае текст будет отображаться "как есть".

В квадратных скобках может содержаться любое выражение, корректное с точки зрения компилятора. Подробнее о выражениях читайте в главе "Выражения".

Например, объект со следующим текстом:

```
2 * 2 = [2 * 2]
```

будет напечатан так:

```
2 * 2 = 4
```

Частая ошибка – попытка написать выражение за пределами квадратных скобок. Напоминаем, что считается выражением и вычисляется только то, что находится в квадратных скобках. Остальной текст печатается "как есть". Например:

```
2 * 2 = [2] * [2]
```

Такой текст будет напечатан следующим образом:

```
2 * 2 = 2 * 2
```

В выражениях могут встречаться элементы, заключенные в собственные квадратные скобки (см. раздел "Обращение к данным отчета"). Например, это обращения к системным переменным. Рассмотрим следующий пример содержимого объекта "Текст":

```
Следующая страница: [[Page] + 1]
```

Текст содержит выражение `[Page] + 1`: это содержимое внешней пары скобок. `Page` – это системная переменная, которая возвращает номер текущей страницы отчета. Она заключена в собственные скобки. Это обязательно должны быть квадратные скобки, независимо от настройки объекта "Текст" (свойство `Brackets`, которое мы рассматривали ранее).

Строго говоря, мы должны были использовать две пары квадратных скобок при попытке распечатать системную переменную `Date` в ранее рассмотренных примерах:

```
Сегодня [[Date]]
```

Здесь внешняя пара скобок обозначает выражение, внутренняя необходима, так как мы обращаемся к системной переменной. Однако FastReport позволяет опускать лишнюю пару скобок, если в выражении есть только один член.

Отображение полей данных

Для обращения к полям источников данных используется следующая форма записи:

```
[Имя источника.Имя поля]
```

Как видно, здесь так же используются квадратные скобки. Имя источника отделяется от имени поля точкой, например:

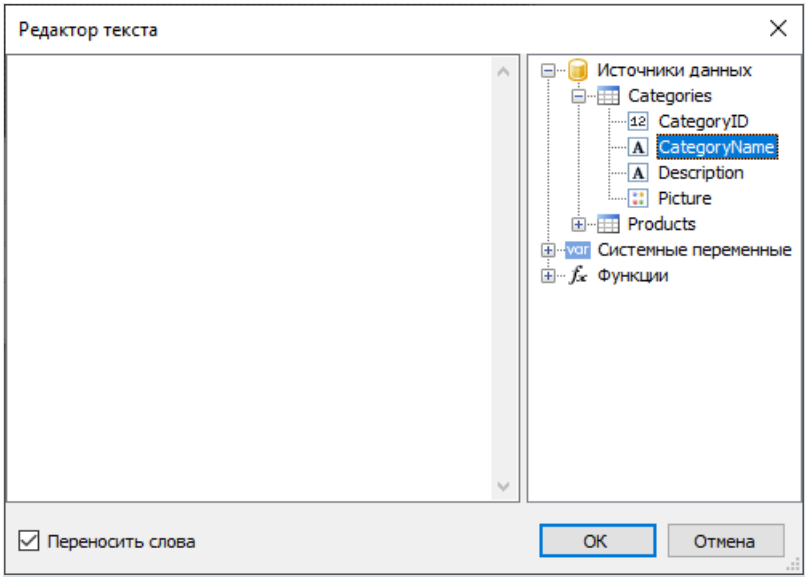
```
[Employees.FirstName]
```

Подробнее об использовании полей в выражениях см. раздел "Поля источников данных".

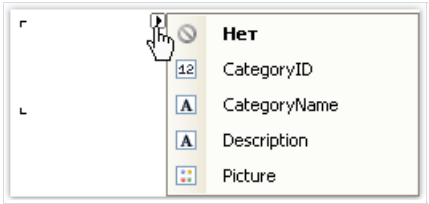
Для вставки поля в объект "Текст" можно использовать несколько способов.

Способ 1: в редакторе объекта "Текст" пишем название поля вручную. Этот способ самый неудобный, так как можно легко допустить ошибку.

Способ 2: в редакторе объекта выбираем нужное поле и перетаскиваем его в текст (также можно сделать двойной щелчок на поле):



Способ 3: щелкаем мышкой на маленькой стрелочке в верхнем правом углу объекта и в открывшемся списке выбираем нужное поле:



Способ 4: перетаскиваем нужное поле из окна "Данные" на страницу отчета с помощью мыши (drag&drop). В данном случае создается объект "Текст", который содержит ссылку на выбранное поле.

Поддержка HTML тегов в тексте

Ранее в FastReport в объекте "Текст" была доступна возможность использования некоторых простейших тегов HTML с помощью свойства `HtmlTags`. Однако, свойство `HtmlTags` было заменено новым свойством `TextRenderType`. Новое свойство предоставляет более широкий спектр функциональности, чем просто обработка HTML тегов.

Свойство `TextRenderType` имеет три возможных значения:

- **Default** - просто текст, без преобразования тегов;
- **HtmlTags** - применение HTML тегов. Список их довольно ограничен: `<b>`, `<i>`, `<u>`, `<strike>`, `<br>`, `<sub>`, `<sup>`, `<img>`;
- **HtmlParagraph** - позволяет регулировать межстрочный интервал, красную строку и все те же теги, что и **HtmlTags**;
- **Inline** - отображает текст в простом формате. Только для внутреннего использования - не рекомендуется для внешнего использования.

Одним из тегов, доступных в предыдущем свойстве `HtmlTags`, был тег `<font>`. Этот тег сегодня считается устаревшим, и его поддержка не гарантирована во всех браузерах. В связи с этим был внедрен новый механизм визуализации, позволяющий применять определенные стили CSS с использованием атрибута `style` для тега `<span>`.

Более детально рассмотрим режимы обработки тегов HTML:

HtmlTags

Как ранее уже было сказано, объект "Текст" поддерживает следующие HTML теги:

- `<b>` - выделение текста жирным начертанием

Пример использования:

```
<b>FastReport</b>
```

Результат:

- `<i>` - курсивное начертание текста

Пример использования:

```
<i>FastReport</i>
```

Результат:

- `<u>` - подчеркивание текста снизу

Пример использования:

```
<u>FastReport</u>
```

Результат:

- `<strike>` - перечеркнутый текст

Пример использования:

```
<strike>FastReport</strike>
```

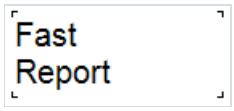
Результат:

- `<br>` - перенос строки

Пример использования:

```
Fast<br>Report
```

Результат:

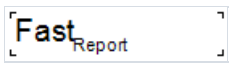


- `<sub>` - отображение текста в нижнем индексе

Пример использования:

```
Fast<sub>Report</sub>
```

Результат:

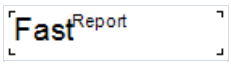


- `<sup>` - отображение текста в верхнем индексе

Пример использования:

```
Fast<sup>Report</sup>
```

Результат:



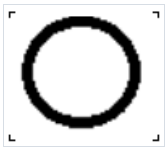
- `<img>` - вставка изображения в текст

  Тег `<img>` поддерживает только атрибут `src`, который может содержать ссылку на изображение (http, https, base64), а также атрибуты `width` и `height`, определяющие размеры изображения в пикселях. Таким образом, изображения могут быть вставлены непосредственно в текст. Ссылки на изображения с протоколами http и https применимы для веб-отчетов.

Пример использования:

```
<img
src="
v8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQAAAGkSURBVGhD7dDRjoMgEAXQ/v9Pd417poEVBBRttul569yRuenjOc+jxd5pEx7SaIQvjzr
+vfvneGvQkc8cnMe73cY+cOQabnQYWe1gtcTGLqst3Xu7LHXzWYmNXX1LFeKjvLIhruvYKJHN4MWcrKIVl8jm8W5OVrKblchm83pOtlEP
SmTXcCMnyw2UFlzJpYQgV5luCK7nXkKQ6CptehdXg2miNNoQ3MXVhCC0S5vey+1gGja/NwT3cjshWDVKm76DBsF09S09lQYJwZ/SwmD6P
noE02/p+fQIpt/S8+kRTL+l59MjmH5C6YU8mL6DBgnBPyptuvrE0gvBvdxOCFaFTraC6b3cDqahXXohuIurCUEoF7IbTO/iakIQukovBN
dzLyFIVNv4IiG4kksJQW6g9EJ2DTdystxeD9/lZLN5PSfbaJTwdU42j3dzspJ2A2/kZDN4MSer6DrvpQ3xUV4psVHRe9hjJTZG+LLExq6
Bk17dZbXExi6rLWP/k7ev4UaHsdK/HJnHu92OlP7l4DneGnS89Iv7g3x8yITSL+rU2Tvp+fwB+FeG5yXqCcoAAAAASUVORK5CYII="
alt=""/>
```

Результат:



HtmlParagraph

Этот режим включает новый обработчик HTML тегов. Он обрабатывает тот же набор тегов, что и `HtmlTags`, а также включает тег `<span>`. Несмотря на то, что набор тегов у этого режима почти тот же, отрисовываются они иначе. Особенно это заметно на тегах `<sub>` и `<sup>`.

Тег `<span>` предоставляет возможность задавать стили для текста с использованием атрибута `style`. Это позволяет применять различные CSS стили, такие как цвет текста, размер шрифта, выравнивание и другие параметры форматирования, непосредственно к определенному фрагменту текста. Таким образом, вместо использования устаревшего тега `<font>`, который ограничивает функциональность и уступает в гибкости, теперь можно достигать того же эффекта, используя тег `<span>` с простейшими CSS стилями в атрибуте `style`.

Примеры использования:

```
<span style="font-size:20pt;">FastReport</span>
<span style="color:red;">Fast</span>Report
<span style="font-family:Consolas;">FastReport</span>
<span style="background-color:yellow;">FastReport</span>
```

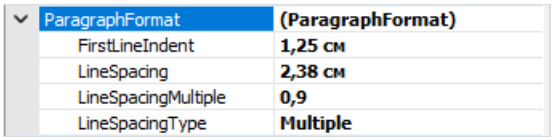
Результат:

ParagraphFormat

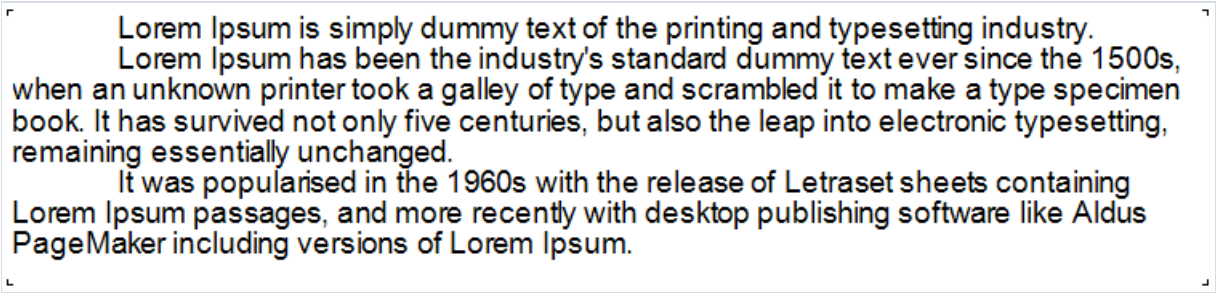
Отдельно стоит рассмотреть свойство `ParagraphFormat`. Оно работает в связке со свойством `HtmlParagraph` и представляет собой настройки для отображения абзацев (межстрочный интервал, отступ красной строки). А именно:

- **FirstLineIndent** - отступ первой строки;
- **LineSpacing** - расстояние между строками в сантиметрах;
- **LineSpacingMultiple** - коэффициент умножения на значение предыдущего параметра. Работает с типом Multiple;
- **LineSpacingType** - тип межстрочного расстояния:
  - Одиночный (Single);
  - Как минимум (At least);
  - В точности (Exact);
  - Множественный (Multiple).

Пример настроек:



Результат:



Свойства объекта "Текст"

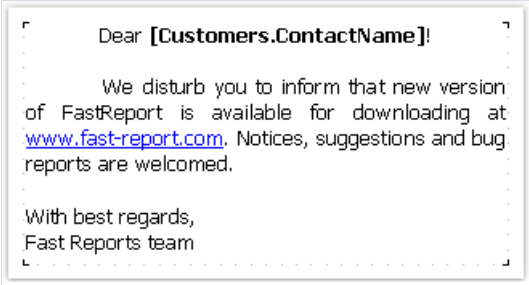
Свойство	Описание
"Выражения в тексте" (AllowExpressions)	Отключение обработки выражений в тексте объекта. По умолчанию включено.
"Поворот" (Angle)	Угол поворота текста в градусах.
"Автосжатие шрифта" (AutoShrink)	Автоматическое уменьшение размера шрифта (`Font.Size`) или его ширины (`FontWidthRatio`) так, чтобы вместить весь текст.
"Минимальный размер шрифта" (AutoShrinkMinSize)	Минимальный размера шрифта или минимальное значение свойства `FontWidthRatio` , в зависимости от значения свойства `AutoShrink` .
"Автоширина" (AutoWidth)	Автоматический подбор ширины объекта так, чтобы поместить самую длинную строку текста целиком, без переноса слов.
"Скобки" (Brackets)	Пара символов, которые используются для обозначения выражений в тексте объекта.
"Разрываться в..." (BreakTo)	С помощью этого свойства можно организовать "перетекание" текста между двумя объектами "Текст". Допустим, у нас есть объекты A и B. Объект A содержит большой объем текста, часть которого не помещается в объекте. Если в свойстве `BreakTo` этого объекта указать объект B, в нем будет напечатан текст, не поместившийся в объекте A.
"Обрезать" (Clip)	Обрезание текста, который не поместился в объекте. По умолчанию включено.
"Повторяющиеся значения" (Duplicates)	Это свойство определяет, как будут печататься повторяющиеся значения. Подробнее об использовании этого свойства можно почитать в разделе "Форматирование".
"Первая табуляция" (FirstTabOffset)	Это свойство определяет, на сколько пикселей сдвинуть первый символ табуляции.
"Шрифт" (Font)	Параметры шрифта.
"Ширина шрифта" (FontWidthRatio)	Коэффициент масштабирования шрифта по ширине. По умолчанию свойство равно 1. Чтобы увеличить ширину шрифта, укажите значение > 1; чтобы уменьшить ширину, укажите значение между 0 и 1.
"Скрывать значение" (HideValue)	Скрытие значений выражений, которые равны заданному значению. Подробнее об использовании этого свойства можно почитать в разделе "Форматирование".
"Скрывать нули" (HideZeros)	Скрытие нулевых значений выражений. Подробнее об использовании этого свойства можно почитать в разделе "Форматирование".

Свойство	Описание
"Условное выделение" (Highlight)	Изменение внешнего вида объекта в зависимости от заданных условий. Подробнее об использовании этого свойства можно почитать в разделе "Форматирование".
"Html теги" (HtmlTags)	Устарело, заменено на `TextRenderType`.
Выравнивание текста (HorzAlign, VertAlign)	Эти свойства задают выравнивание текста внутри объекта по горизонтали и вертикали.
"Межстрочный интервал" (LineHeight)	Межстрочный интервал, измеряемый в пикселях. Значение по умолчанию = 0, при этом используется стандартный межстрочный интервал.
"Слияние текстовых объектов" (MergeMode)	Объединение текстовых объектов с одинаковым содержимым, находящихся рядом. Содержит 2 значения: - `Horizontal` – объединение по горизонтали; - `Vertical` – объединение по вертикали. Допустимо использование как отдельно, как и совместно. *Важное замечание! Объекты должны находиться вплотную друг к другу.*
"Нулевое значение" (NullValue)	Строка, которая будет выводиться вместо null значения. Для использования необходимо отключить флажок "Преобразовывать null значения" в меню "Отчет|Настройки...".
"Отступы" (Padding)	Отступы текста от краев объекта, измеряемые в пикселях.
"Формат абзаца" (ParagraphFormat)	Настройки для отображения абзацев (межстрочный интервал, отступ красной строки). Подробнее об использовании этого свойства можно почитать на странице "Поддержка HTML тегов в тексте".
"Смещение абзаца" (ParagraphOffset)	Смещение абзаца, измеряемое в пикселях. Для `TextRenderType.HtmlParagraph` используйте свойство `ParagraphFormat.FirstLineIndent`.
"Справа налево" (RightToLeft)	Вывод текста справа налево.
"Набор позиций символов табуляции" (TabPositions)	Набор позиций символов табуляции, в пикселях. Отрицательные значения не оказывают влияния на это свойство.
"Ширина табуляции" (TabWidth)	Задание ширины символа табуляции, измеряемое в пикселях.
"Текст" (Text)	Текст объекта.
"Цвет текста" (TextFill)	Цвет текста. С помощью редактора этого свойства можно выбрать любую из доступных заливок.

Свойство	Описание
"Контур текста" (TextOutline)	Контур текста.
"HTML теги" (TextRenderType)	Это свойство позволяет использовать HTML теги в тексте объекта. Подробнее об использовании этого свойства можно почитать на странице "Поддержка HTML тегов в тексте".
"Отсечение" (Trimming)	Это свойство определяет, как показывать текст, который выходит за границы объекта. Используется только в том случае, если свойство "Перенос слов" отключено.
"Подчеркивание" (Underlines)	Линии подчеркивания под каждой строкой текста. Подчеркивание можно использовать только для текста, выровненного по верхнему краю.
"Перенос слов" (WordWrap)	Это свойство определяет, надо ли переносить текст по словам.
Wysiwyg	Изменение режима отображения текста таким образом, чтобы добиться максимального соответствия между отображением текста на экране и на распечатке. Этот режим неявно включается, если использовать выравнивание текста по ширине или нестандартный межстрочный интервал.

Объект "Форматированный текст"

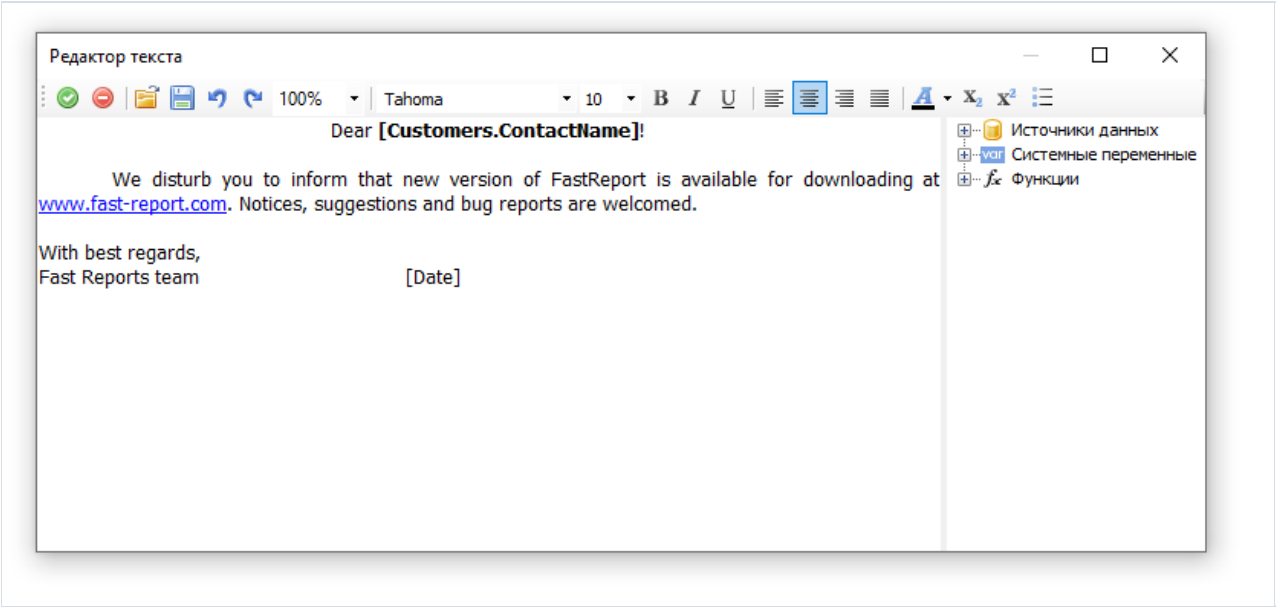
Этот объект позволяет отображать форматированный текст в формате RTF. Он выглядит следующим образом:



> По возможности старайтесь обходиться объектом "Текст" для отображения текста. При экспорте отчета в другие форматы объект "Форматированный текст" экспортируется в виде картинки.

> Объект поддерживает только сплошной тип заливки. Градиентные и прочие заливки не поддерживаются.

Для редактирования текста сделайте двойной щелчок на объекте:



Вы также можете использовать пакет Microsoft Word для создания текста. После того как вы создали текст, сохраните его в формате RTF. Затем вызовите редактор объекта и нажмите кнопку  , чтобы загрузить текст из файла RTF.

> Объект поддерживает далеко не все возможности Microsoft Word.

Подключить объект к данным из поля БД можно двумя способами:

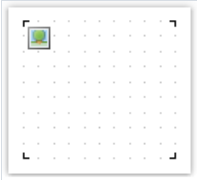
- в текст объекта можно вставлять выражения. Они обрабатываются таким же образом, как и в объекте "Текст". Вставьте ссылку на поле в текст объекта;
- используйте свойство `DataColumn`, чтобы напечатать содержимое поля БД целиком.

Объект имеет следующие свойства:

Свойство	Описание
"Выражения в тексте" (AllowExpressions)	Возможность отключать обработку выражений в тексте объекта. По умолчанию включено.
"Скобки" (Brackets)	Пара символов, которые используются для обозначения выражений в тексте объекта.
"Поле данных" (DataColumn)	Поле данных, из которого загружается текст объекта.
"Текст" (Text)	Текст объекта в формате RTF.
"Отступы" (Padding)	Отступы текста от краев объекта, измеряемые в пикселах.

Объект "Рисунок"

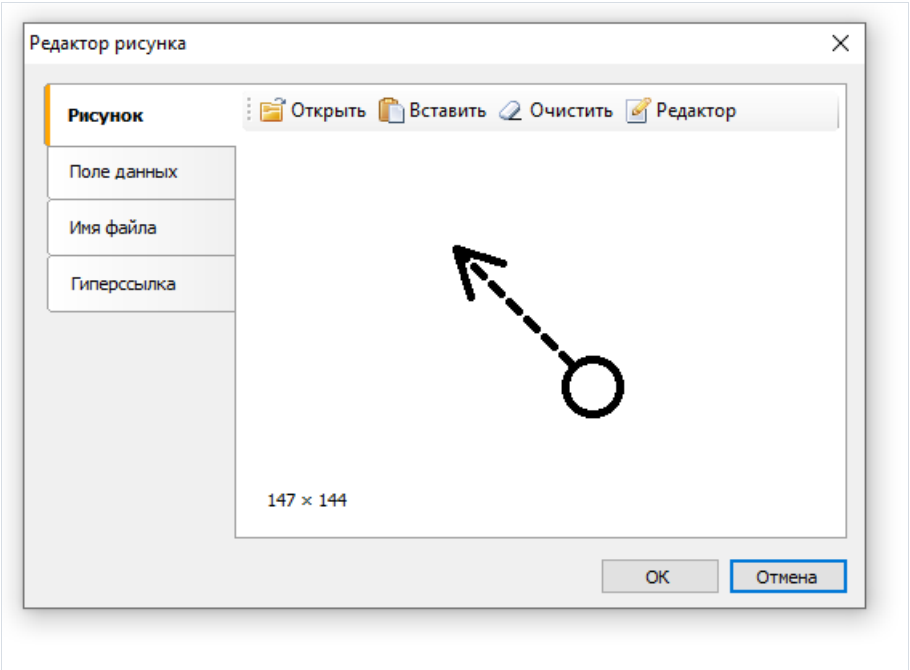
С помощью объекта "Рисунок" можно добавить в отчет логотип фирмы, фотографию сотрудника или любую другую графическую информацию. Объект может отображать графику в форматах BMP, PNG, JPG, GIF, TIFF, ICO, EMF, WMF. Выглядит следующим образом:



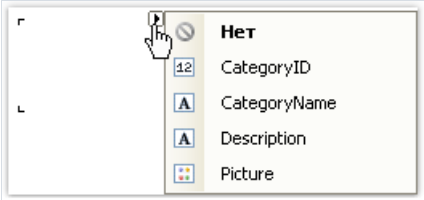
Объект может показывать данные из следующих источников:

Источник	Описание
Файл с картинкой	Рисунок загружается из файла и хранится внутри отчета. Рисунок хранится в свойстве `Image`.
Поле данных	Рисунок загружается из поля источника данных. Имя поля хранится в свойстве `DataColumn`.
Имя файла	Рисунок загружается из файла с указанным именем. Имя файла хранится в свойстве `ImageLocation`. Внутри отчета рисунок не хранится. Необходимо распространять файл рисунка вместе с отчетом.
URL	Рисунок загружается из Интернета каждый раз, когда отчет строится. Внутри отчета рисунок не хранится. Адрес хранится в свойстве `ImageLocation`.

Чтобы выбрать один из источников данных для рисунка, сделайте двойной щелчок мышью на объекте. Вы увидите редактор объекта "Рисунок":



Быстро подключить объект к полю данных можно, нажав мышкой на маленькой стрелочке в верхнем правом углу объекта и в открывшемся списке выбрав нужное поле:

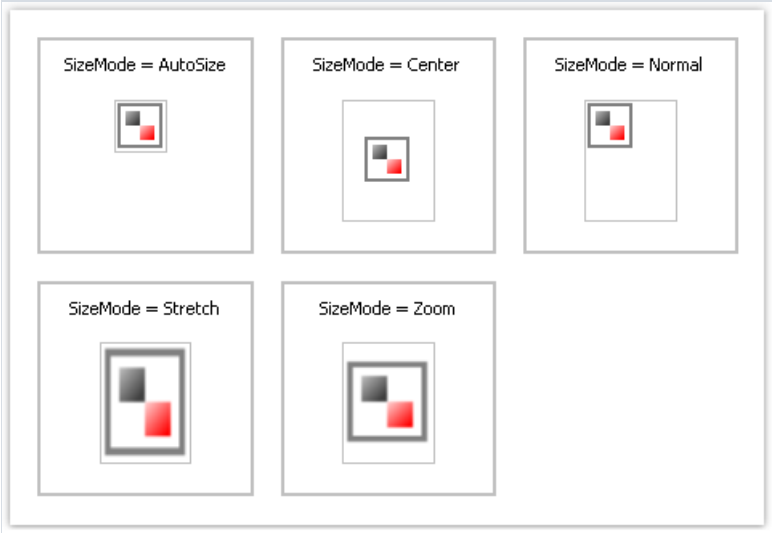


Также можно перетащить нужное поле из окна "Данные" на страницу отчета с помощью мыши (drag&drop). В данном случае создается объект "Рисунок", который содержит ссылку на выбранное поле.

В контекстном меню объекта можно настроить режим отображения рисунка:

- Авторазмер (AutoSize) - объект принимает размеры рисунка;
- Центрировать (CenterImage) - рисунок центрируется внутри объекта;
- Нормальный (Normal) - рисунок выводится в левом верхнем углу объекта без масштабирования;
- Растянуть (StretchImage) - рисунок растягивается до размеров объекта. Пропорции рисунка не соблюдаются.
- Масштабировать (Zoom) - рисунок растягивается до размеров объекта с соблюдением пропорций.

Различия между режимами показаны на следующем рисунке:



Объект имеет следующие свойства:

Свойство	Описание
"Угол поворота" (Angle)	Угол, на который нужно повернуть рисунок. Возможные значения для этого свойства - 0, 90, 180, 270.
"Режим отображения" (SizeMode)	Режим отображения рисунка.
"Прозрачность" (Transparency)	Степень прозрачности картинки. Свойство может иметь значение между 0 до 1. Значение 0 (по умолчанию) означает, что картинка непрозрачна.
"Прозрачный цвет" (TransparentColor)	Цвет, который будет прозрачным при отображении картинки.
"Рисунок" (Image)	Собственно рисунок.
"Поле данных" (DataColumn)	Поле данных, из которого загружать рисунок.

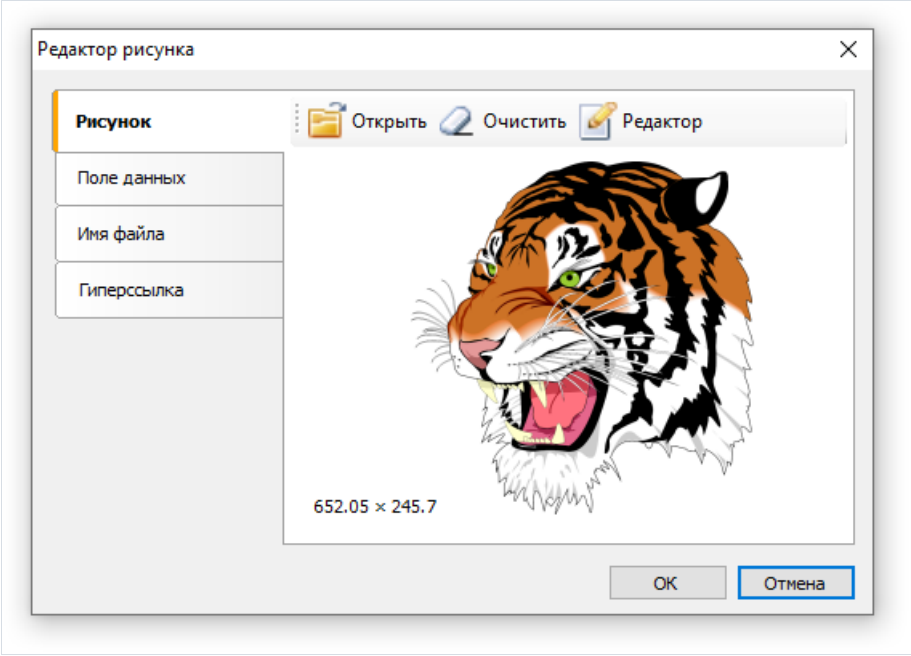
Свойство	Описание
"Месторасположение рисунка" (ImageLocation)	Свойство может содержать имя файла или URL. Рисунок будет загружен из указанного места при построении отчета.
"Отступы" (Padding)	Свойство позволяет задать отступы рисунка от краев объекта, в пикселах.
"Показывать ошибку" (ShowErrorImage)	Показывает значок "Нет рисунка" в случае, если рисунок пустой. Это свойство имеет смысл использовать, если рисунок загружается из Интернета.

Объект "SVG"

Этот объект предназначен для отображения векторных изображений в формате SVG. Пример такого изображения:

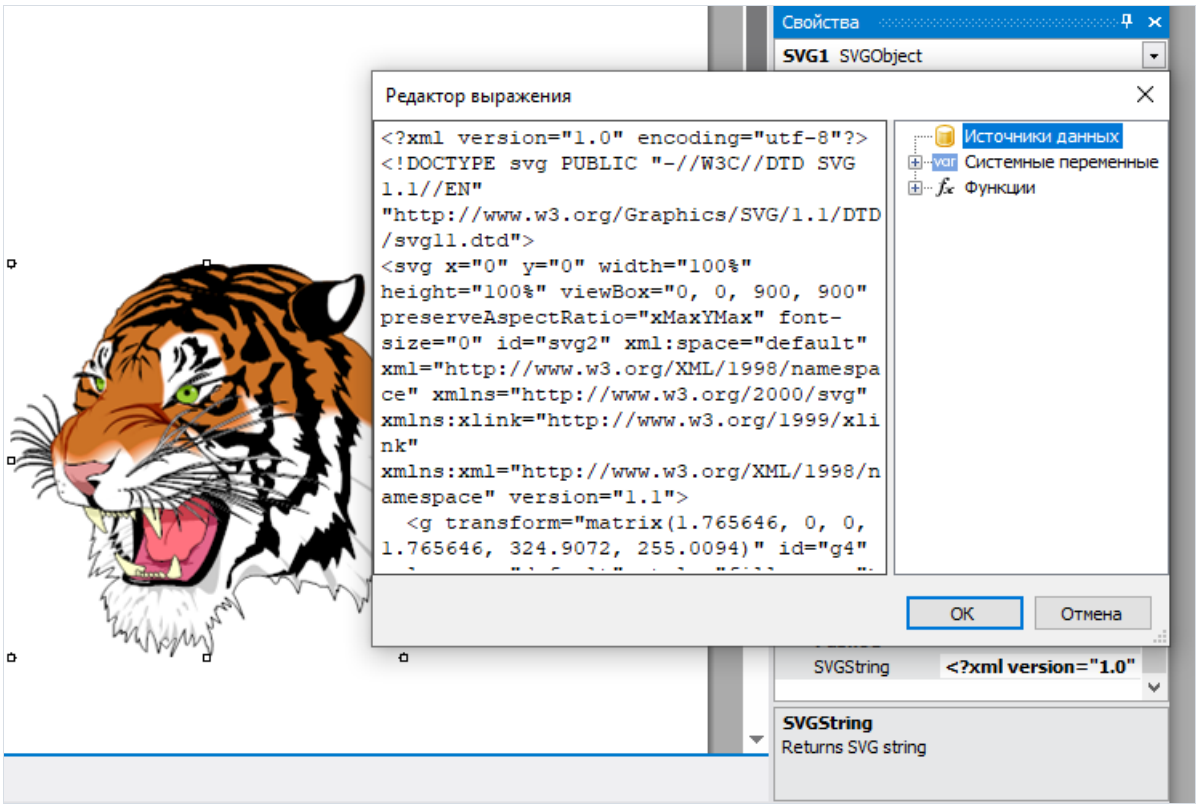
Чтобы добавить SVG-изображение, добавьте объект "SVG" в свой отчёт с помощью следующей кнопки на панели объектов: 

После этого нажмите на объект дважды, или в контекстном меню выберите "Редактировать". Это откроет редактор изображения, который аналогичен редактору объекта "Рисунок".



В этом редакторе вы можете задать изображение, которое будет показываться объектом. Вы можете включить изображение в шаблон отчёта (кнопка "Открыть" на вкладке "Рисунок"), выбрать SVG-изображение из таблицы данных (вкладка "Поле данных"), установить соединение со внешним SVG-файлом (вкладка "Имя файла", в этом случае вам нужно будет распространять этот файл вместе с шаблоном) или же задать гиперссылку, с адреса которой будет получаться изображение (вкладка "Гиперссылка").

Если вы включите изображение в шаблон отчёта, то его строковое представление будет храниться в свойстве `SVGString`. На скриншоте показан фрагмент такой строки:

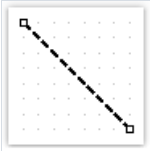


Объект SVG поддерживает такие же режимы размера, как и объект "Рисунок": Авторазмер (`AutoSize`), Центрировать (`CenterImage`), Нормальный (`Normal`), Растянуть (`StretchImage`), Масштабировать (`Zoom`).

Подробное описание этих режимов приведено в описании объекта "Рисунок".

Объект "Линия"

Объект "Линия" может отображать горизонтальную, вертикальную или диагональную линию. Линии могут использоваться для оформления отчета. Объект выглядит следующим образом:



> Там, где это возможно, используйте вместо линий рамки объектов. Это упростит отчет (в нем не будет лишних объектов), а также избавит от возможных проблем при экспорте отчета в различные форматы.

Дизайнер FastReport имеет удобные средства для рисования линий. Для того чтобы добавить в отчет линию, нажмите кнопку  на панели инструментов "Объекты" и в меню выберите объект "Линия" или "Диагональная линия". Поместите курсор мыши в то место, где будет начинаться линия. Затем нажмите и удерживайте левую кнопку мыши, чтобы нарисовать линию. После этого вы снова можете нарисовать линию. Когда все линии нарисованы, нажмите кнопку  на панели инструментов "Объекты".

Обычная линия отличается от диагональной тем, что ее можно сделать только вертикальной или горизонтальной.

> Для объекта "Линия" нельзя выбрать стиль линии – "Двойная". Хотя этот стиль и доступен в выпадающем списке на панели инструментов "Рамка и заливка", он относится только к рамке вокруг объектов.

Объект "Линия" имеет следующие свойства:

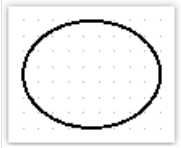
Свойство	Описание
"Диагональная" (Diagonal)	Свойство определяет, является ли линия диагональной. Обычную линию можно превратить в диагональную, включив это свойство.
Наконечники линии (StartCap, EndCap)	Эти свойства позволяет задать один из следующих типов наконечников для линии: - эллипс; - прямоугольник; - ромб; - стрелка. Размеры наконечника (ширина и высота) задаются в свойствах `Width`, `Height` наконечника. Можно настроить наконечник для каждого конца линии.

Объект "Фигура"

Объект "Фигура" используется для оформления отчета. Он позволяет отображать одну из следующих геометрических фигур:

- прямоугольник;
- скругленный прямоугольник;
- эллипс;
- треугольник;
- ромб.

Объект выглядит следующим образом:



Для вставки фигуры в отчет нажмите кнопку  на панели инструментов "Объекты" и выберите нужный тип фигуры из списка.

Фигура, как и другие объекты отчета, может иметь заливку и рамку. В отличие от объекта "Текст", вы не можете управлять отдельными линиями рамки. Также нельзя использовать стиль линии "Двойная".

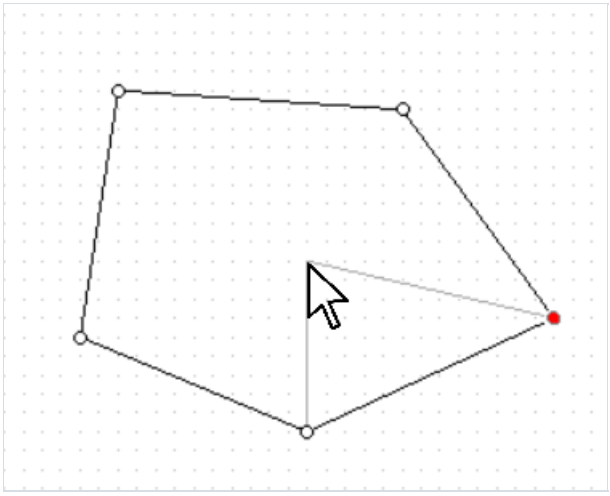
> Там, где это возможно, используйте для оформления отчета рамки объектов.

Объект имеет следующие свойства:

Свойство	Описание
"Тип фигуры" (Shape)	Это свойство позволяет выбрать тип фигуры.
"Закругление" (Curve)	Это свойство позволяет задать закругление для фигуры типа "Скругленный прямоугольник".

Объекты "Ломаная линия" и "Многоугольник"

Эти объекты используются для отображения разнообразных фигур, таких как многоугольники. Они находятся в той же категории, что и объекты "Фигура" и "Линия". Чтобы разместить объект "Многоугольник", откройте меню под кнопкой  и выберите один из многоугольников. Можно разместить готовый пяти-, шести-, семи- или восьмиугольник. Также можно выбрать "Многоугольник", чтобы нарисовать фигуру самостоятельно. Пример того, как может выглядеть многоугольник:

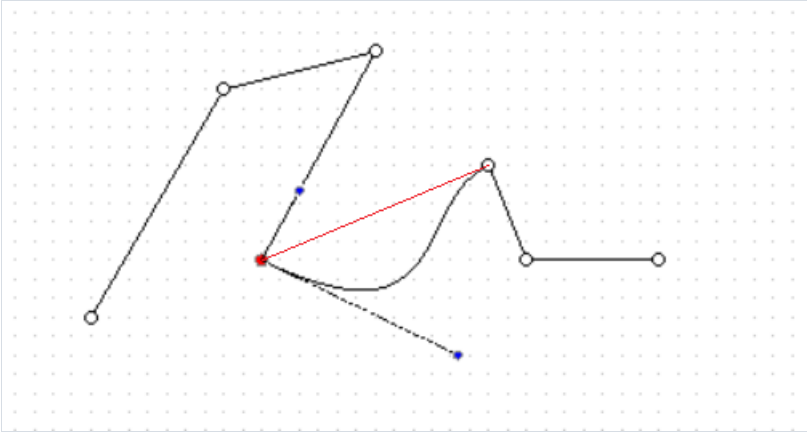


Серые линии на изображении помогают увидеть, как объект будет выглядеть после добавления новой точки.

После размещения первой точки, вы можете добавлять дополнительные. После того, как завершите рисовать многоугольник, нажмите `Escape` или переключите режим редактирования. Панель режимов редактирования для объектов "Ломаная линия" и "Многоугольник" находится на вкладке "Главная".

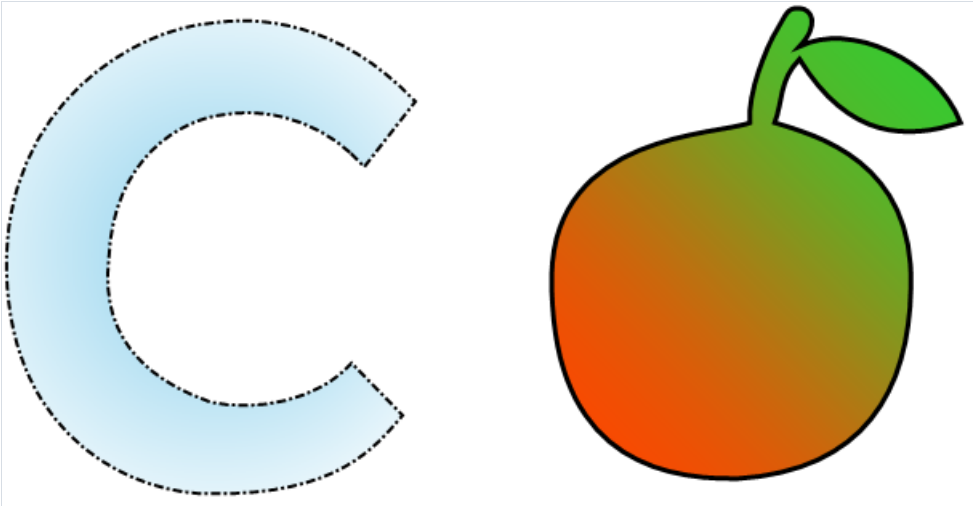
Режимы редактирования ломаной линии и многоугольника:

Кнопка	Режим	Описание
	Добавление точек	Включается сразу после добавления многоугольника. Чтобы добавить точку, выберите этот режим и выделите одну из точек объекта. После этого наведите на желаемое место размещения точки и нажмите левую клавишу мыши. Точка добавится рядом с той, которую вы выделили.
	Передвижение и изменение размера	В этом режиме можно перемещать объект целиком, а также сжимать его или растягивать.
	Удаление точек	Чтобы удалить точку из многоугольника или ломаной линии, выберите этот режим и нажмите на точку, которую хотите удалить. Также можно выделить её в другом режиме и нажать клавишу `Delete`.
	Указатель мыши	Этот режим позволяет просматривать и перемещать все точки объекта, добавленные ранее.
	Опорная точка кривой	В этом режиме можно добавить к линии, смежной с выбранной точкой, точку кривизны. Эта линия станет кривой Безье, а добавленная точка будет играть роль опорной.



Так выглядит ломаная линия с установленной точкой кривизны (синяя). Красная линия показывает состояние линии до того, как она была преобразована в кривую.

Оба объекта имеют возможность настроить их границу, а объект "Многоугольник" поддерживает те же режимы заливки, что и остальные объекты.



Объект "Штрихкод"

Объект позволяет отображать в отчете штрихкод. Он выглядит следующим образом:

Объект "Штрихкод" поддерживает следующие категории штрихкодов:

Двумерные

Название	Длина	Таблица символов
PDF417	Не фиксирована	Любые символы
Datamatrix	Не фиксирована	Любые символы
QR Code	Не фиксирована	Любые символы
Aztec	Не фиксирована	Любые символы
MaxiCode	Не фиксирована	138 цифровых символа или 93 алфавитных

EAN/UPC

Название	Длина	Таблица символов
EAN-8	8	0-9
EAN-13	13	0-9
UPC-A	12	0-9
UPC-E0	6	0-9
UPC-E1	6	0-9

Почтовые

Название	Длина	Таблица символов
Deutsche Identcode	12	0-9
Deutsche Leitcode	14	0-9
PostNet	Не фиксирована	0-9

Название	Длина	Таблица символов
Japan Post 4 State Code		
Intelligent Mail (USPS)	Не фиксирована	0-9, A-Z

GS1

Название	Длина	Таблица символов
GS1-128	Не фиксирована	0-9, A-Z
GS1 DataBar Omnidirectional		
GS1 DataBar Limited		
GS1 DataBar Stacked		
GS1 DataBar Stacked Omnidirectional		
GS1 Datamatrix		

Прочие

Название	Длина	Таблица символов
2 of 5 Interleaved	Не фиксирована	0-9
2 of 5 Industrial	Не фиксирована	0-9
2 of 5 Matrix	Не фиксирована	0-9
ITF-14	14	0-9
Codabar	Не фиксирована	0-9, `-`, `$`, `:`, `/`, `.`, `+`
Code 128	Не фиксирована	128 ASCII символов
Code 39	Не фиксирована	0-9, A-Z, `-`, `.`, `*`, `$`, `/`, `+`, `%`
Code 39 Extended	Не фиксирована	128 ASCII символов
Code 93	Не фиксирована	0-9, A-Z, `-`, `.`, `*`, `$`, `/`, `+`, `%`
Code 93 Extended	Не фиксирована	128 ASCII символов
MSI	Не фиксирована	0-9
2-Digit Supplement	2	0-9
5-Digit Supplement	5	0-9

Название	Длина	Таблица символов
Plessey	Не фиксирована	Шестнадцатеричные цифры (0-F)
Pharmacode	Не фиксирована	0-9

Данные штрихкода поступают в объект в виде строки. Строка может содержать любые символы, разрешенные для выбранного типа штрихкода. Некоторые типы кодов являются цифровыми, остальные могут отображать символьную информацию.

Выбрать тип штрихкода можно в контекстном меню объекта.

Подключить объект к данным можно одним из следующих способов:

- указать строку, содержащую текст объекта, в свойстве `Text`;
- подключить объект к полю данных с помощью свойства `DataColumn`;
- указать в свойстве `Expression` выражение, которое возвращает текст объекта.

Объект имеет следующие свойства:

Свойство	Описание
Код (Barcode)	Содержит настройки, специфичные для выбранного типа штрихкода.
Поворот (Angle)	Свойство позволяет задать поворот объекта на один из фиксированных углов – 0, 90, 180, 270 градусов.
Масштаб (Zoom)	Масштабирование штрихкода. Свойство используется только вместе со свойством "Авторазмер".
Авторазмер (AutoSize)	Если это свойство включено, объект будет растягиваться, чтобы показать штрихкод целиком. Если свойство отключено, штрихкод будет растянут до размеров объекта.
Показывать текст (ShowText)	Свойство определяет, надо ли показывать ли текст в нижней части штрихкода.
Поле данных (DataColumn)	Поле данных, из которого загружать текст объекта.
Выражение (Expression)	Выражение, которое возвращает текст объекта.
Текст (Text)	Текст объекта.
Отступы (Padding)	Свойство позволяет задать отступы от краев объекта, в пикселах.

Следующие свойства являются специфичными для выбранного типа штрихкода. Их можно изменить в окне "Свойства", раскрыв свойство `Barcode` у объекта "Штрихкод":

Свойство	Описание
Ширина полосок (WideBarRatio)	Это свойство имеется у всех линейных штрихкодов. Оно определяет относительный размер широких полосок штрихкода.
Контрольная сумма (CalcCheckSum)	Это свойство имеется у всех линейных штрихкодов. Оно определяет, надо ли считать контрольную сумму автоматически. Если это свойство отключено, контрольная сумма должна присутствовать в тексте объекта.
Автокодировка (AutoEncode)	Это свойство имеется у кода Code128. Этот тип штрихкода имеет три кодировки – A, B, C. Необходимо либо прямо указать кодировку в тексте, используя управляющие коды, либо включить это свойство, и кодировка будет подобрана автоматически. В тексте можно использовать следующие управляющие коды: &A; START A / CODE A &B; START B / CODE B &C; START C / CODE C &S; SHIFT &1; FNC1 &2; FNC2 &3; FNC3 &4; FNC4 Если включено свойство "Автокодировка", все управляющие коды будут игнорироваться. Пример использования управляющих кодов в тексте объекта: `&C;1234&B;ABC`
Отношение сторон (AspectRatio)	Относится к коду PDF417. Определяет отношение сторон штрихкода и используется при автоматическом вычислении размеров (если свойства `Columns` и `Rows` не заданы).
Кодовая страница (CodePage)	Относится к кодам PDF417 и Datamatrix. Определяет номер кодовой страницы, которая используется при кодировании символов. Например, для корректной работы с русскими символами необходимо задать значение свойства = 1251.
Колонки, Строки (Columns, Rows)	Относится к коду PDF417. Эти свойства определяют количество колонок и строк в штрихкоде. Если значения свойств равны 0, размер штрихкода будет подобран автоматически. В этом случае также используется свойство `AspectRatio`.
Режим упаковки (CompactionMode)	Относится к коду PDF417. Определяет режим упаковки информации.
Коррекция ошибок (ErrorCorrection)	Относится к коду PDF417. Определяет режим коррекции ошибок.
Размер точки (PixelSize)	Относится к коду PDF417. Определяет размер точки штрихкода, в пикселях. Как правило, высота пиксела должна быть больше его ширины как минимум в 3 раза.
Кодирование (Encoding)	Относится к коду Datamatrix. Определяет тип кодирования информации.
Размер точки (PixelSize)	Относится к коду Datamatrix. Определяет размер точки штрихкода, в пикселях.
Размер символа (SymbolSize)	Относится к коду Datamatrix. Определяет размер блока штрихкода.

PDF417

Этот распространенный двумерный штрихкод предназначен для кодирования больших объемов данных. Его аббревиатура расшифровывается как Portable Data File, а число 417 образовалось в результате сложения 4 и 17. Тут 4 – это четыре штриха и четыре пробела, а 17 – это количество модулей в кодовом слове.

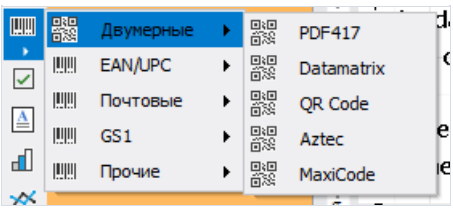
Как уже отмечалось выше, PDF417 – широко распространенный штрихкод, наряду с QR Code, Maxi Code и Data Matrix. Выглядит PDF417 как смесь классического линейного штрихкода и матричного, но на самом деле является сложным линейным кодом. По краям он имеет линии, как в обычных линейных кодах, а в средней части линии расположены в строках друг над другом. Это уплотнение линейного кода достигается за счет расположения строк с кодами друг над другом. Такая структура позволяет хранить большие объемы информации – от 3 до 90 строк, в которых можно закодировать до 1859 алфавитных символов или 2725 числовых символов.

Защита от повреждения кода предусматривает избыточность, которая может покрыть до 50% кода. Это очень высокий показатель, однако при этом размеры кода также будут увеличиваться. По сравнению с матричными кодами, PDF417 занимает в несколько раз больше места при кодировании такого же объема информации, что можно отнести к минусам.

Сфера применения PDF417 довольно обширна. Его используют транспортные компании для печати на пассажирских билетах и на грузовых отправлениях, в почтовых отправлениях, документах отчетности, различных удостоверениях личности, складском учете, и во многих других сферах где требуется маркировка и идентификация.

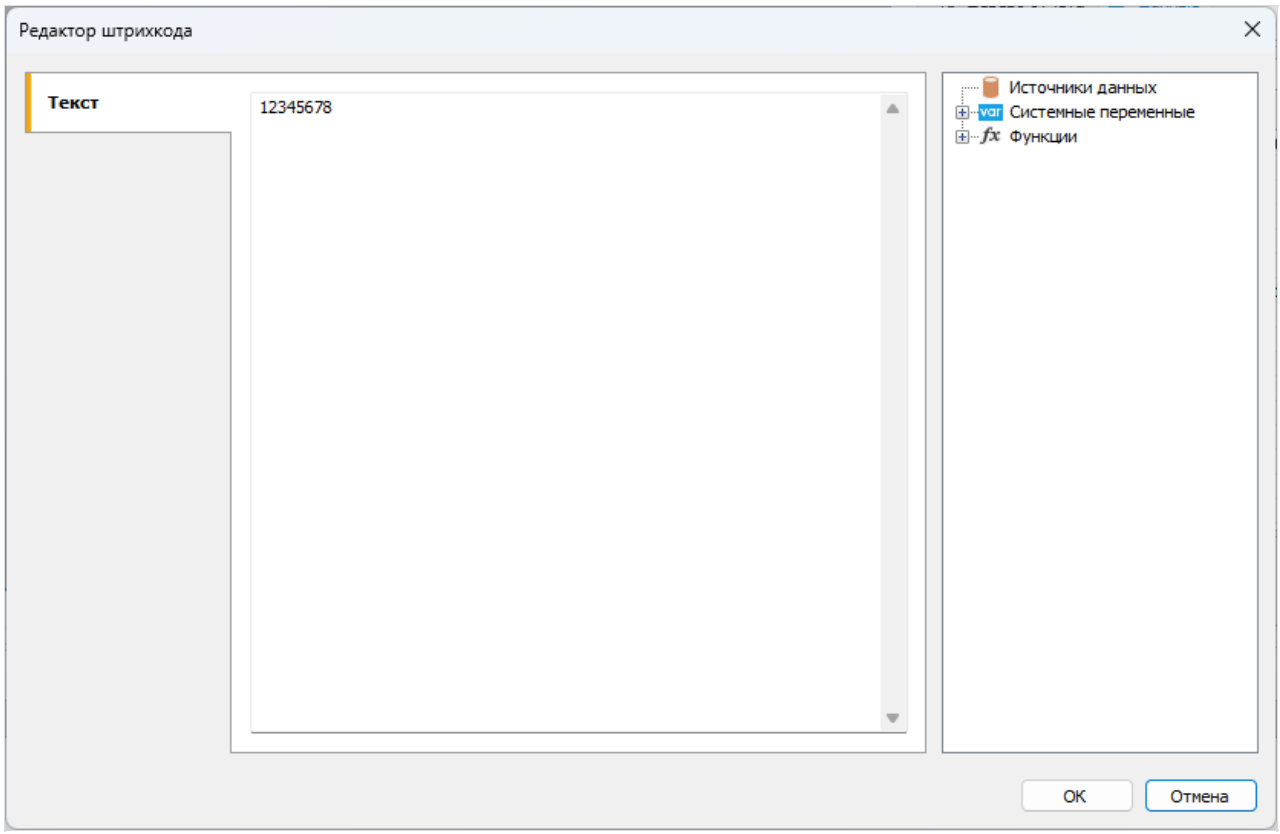
Для считывания этого кода используются лазерные сканеры, немного отличающиеся от обычных сканеров для линейных кодов. Как уже говорилось выше, по бокам кода присутствуют обычные линии, характерные для линейных штрихкодов. Они нужны для идентификации начала и конца кода.

Для формирования штрихкода PDF417 в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Двумерные", а затем PDF417:



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Если необходимо скрыть текст под штрихкодом, следует найти свойство `ShowText` в инспекторе свойств соответствующего штрихкода и установить его значение в `False`.

QR-код

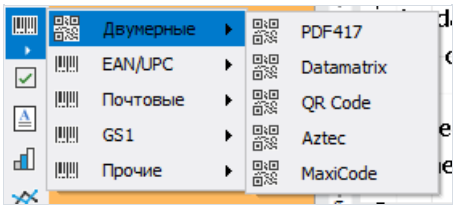
QR-код – двумерный штрихкод, предназначенный для хранения числовой, алфавитно-цифровой и двоичной информации.

Для корректного распознавания QR-кода с помощью камеры используются специальные маркеры по углам и по площади изображения. Это позволяет нормализовать изображение после считывания и преобразовать точечное кодирование в двоичные числа с проверкой контрольной суммы.

QR-код может содержать до 4296 символов, применяя алфавитно-цифровую кодировку.

Объект

Для формирования QR-кода в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Двумерные", а затем QR Code:

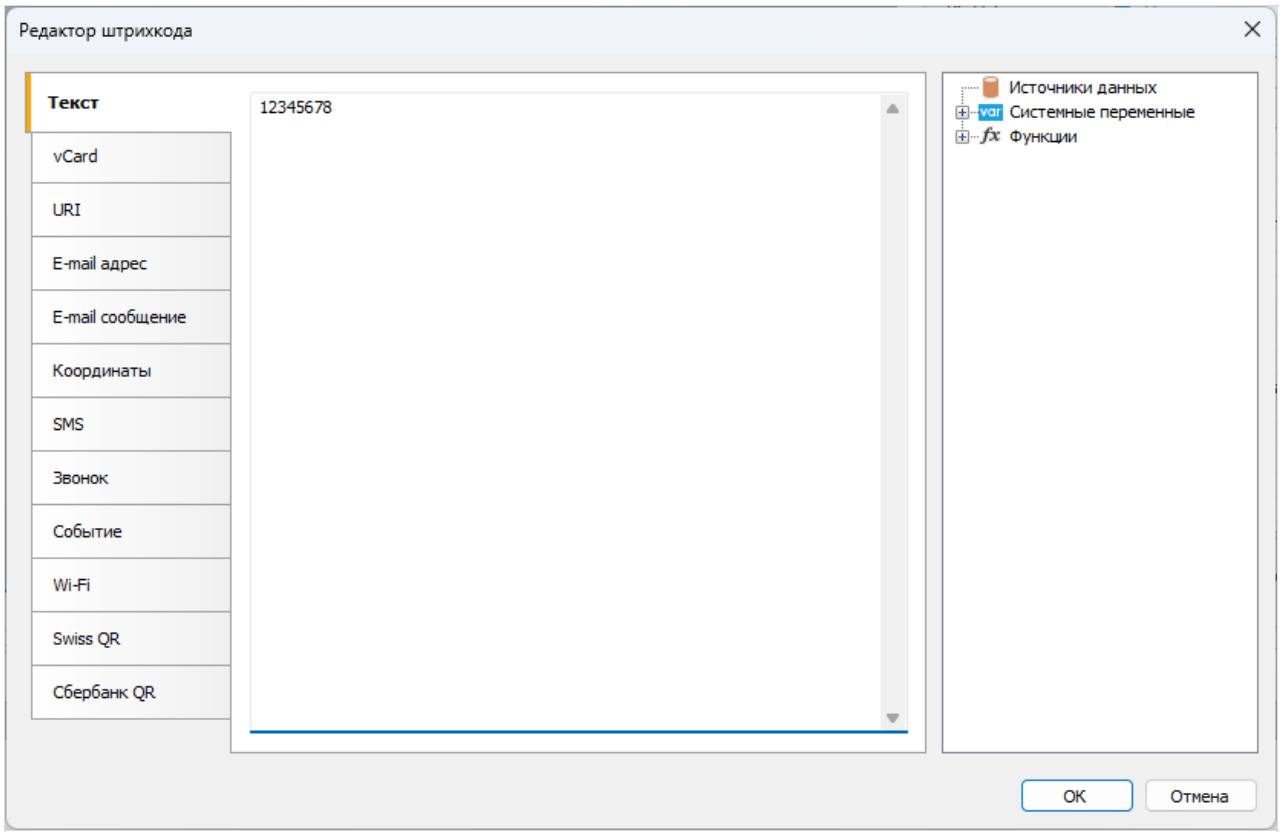


После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши.

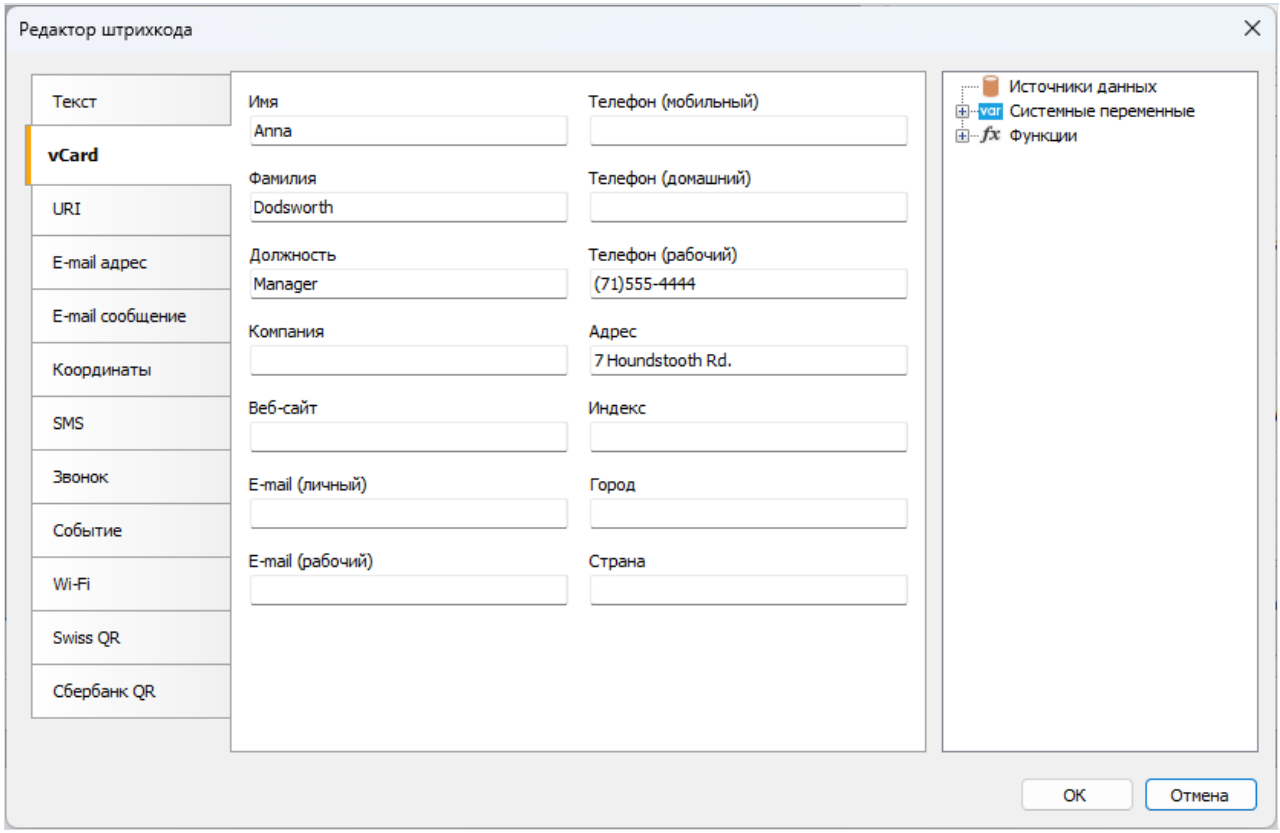
Редактор

Редактор QR-кода похож на обычный редактор выражений. Справа находится дерево данных, параметров и функций. Из него можно перетащить элементы в текстовый редактор.



Основное различие от редактора выражений состоит в наличии вкладок в левой части окна. Эти вкладки определяют тип содержимого QR-кода.

В зависимости от выбранного типа содержимого появляется соответствующий набор полей для заполнения.



Типы содержимого QR-кода

При работе с редактором текст содержимого (четвертая колонка таблицы) формируется автоматически. Поля редактора могут содержать любые выражения, в том числе и поля из источника данных (их можно перетащить из дерева справа).

Тип содержимого	Описание	Пример изображения	Пример содержимого
Text	Текст в буквенном и числовом представлении		`12345678`

vCard	Versitcard - формат представления электронной визитной карточки. Может содержать следующую информацию:
	- Фамилия;
	- Имя;
	- Должность;
	- Компания;
	- Веб-сайт компании;
	- E-mail персональный;
	- E-mail рабочий;
	- Телефон мобильный;
	- Телефон домашний;
	- Телефон рабочий;
	- Адрес;
	- Индекс;
	- Город;
	- Страна.
	Могут быть заполнены лишь некоторые поля.
			`BEGIN:VCARD`
			`VERSION:2.1`
			`FN:Anne Dodsworth`
			`N:Dodsworth;Anne`
			`TITLE:Manager`
			`TEL;WORK;VOICE(71) 555-4444`
			`ADR:;;7 HoundstoothRd.;;;;`
			`END:VCARD`

URI	Унифицированный идентификатор ресурса. Строка со ссылкой на файл, документ, изображение, электронную почту, веб-сайт и др.		`https://быстрыеотчеты.рф/ru/product/fast-report-net/`
E-mail Address	Адрес электронной почты		`support@fastreport.ru`
E-mail Message	Текст электронного письма		`MATMSG:TO:support@fastreport.ru;SUB:Вопрос о FastReport .NET;BODY:Здравствуйте, у меня есть вопрос о FastReport .NET;;`
Geolocation	Координаты для определения реального географического местоположения		`geo:-50.737563,-79.490016,120`

Тип содержимого	Описание	Пример изображения	Пример содержимого
SMS	Текстовое сообщение		`SMSTO:(71) 555-4444: Привет, Андрей! Я в порядке!`
Call	Телефонный номер		`tel:(71) 555-4444`

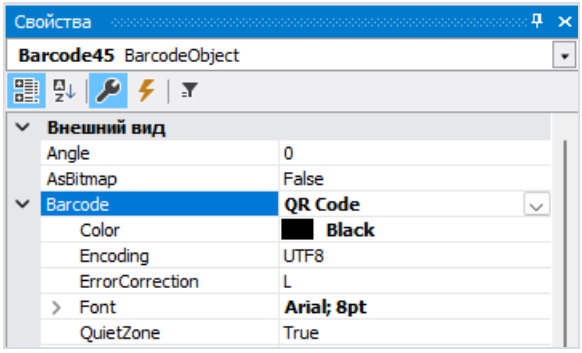
Event	Событие для добавления в календарь. Кроме времени и даты может содержать текстовое сообщение.		`BEGIN:VEVENT`
			`SUMMARY:Семейный пикник`
			`DTSTART:20240415T120000Z`
			`DTEND:20240415T160000Z`
			`DESCRIPTION:Семейный пикник на природе.`
			`END:VEVENT`

Wi-Fi	Информация для подключения к Wi-Fi сети		`WIFI:T:WPA;S:Honeypot;P:youarewelcome;H:true;`

Swiss	Специальный QR-код, содержащий платёжную информацию для Swiss Bill. Подробнее об этом баркоде можно прочесть в другой статье документации.		`SPC`
			`0200`
			`1`
			`CH4431999123000889012`
			`S`
			`Carl Ltd.`
			`Luber`
			`16`
			`123321`
			`Berlin`
			`GE`
			
			`50050.00`
			`EUR`
			`S`
			`Sigmunt Shuld`
			`Lunglen`
			`23`
			`123322`
			`Ferburg`
			`GE`
			`NON`
			
			`EPD`

Свойства QR-кода

Теперь рассмотрим свойства QR-кода. Они доступны в инспекторе объектов под свойством Barcode.



Свойство	Описание
Цвет (Color)	Определяет цвет отображения баркода. По умолчанию – Black (чёрный).
Кодировка (Encoding)	Кодировка текста содержимого баркода, например: UTF8, Windows_1251, CP_866 и др. По умолчанию: UTF8.
Избыточность (ErrorCorrection)	Избыточность для исправления ошибок с помощью кода Рида-Соломона. Оно может принимать значение: L (low – 7%), M (medium – 15%), Q (25%), H (high – 30%). По умолчанию – L.
Поле (QuietZone)	Определяет наличие белой рамки вокруг QR-кода. По умолчанию включено.

Избыточность нужна для корректного чтения данных при частично повреждённом изображении кода или при нанесённом поверх него изображении.

Например, при установленной избыточности в H (30%), баркод, изображённый ниже, без проблем считывается:

Если необходимо скрыть текст под штрихкодом, следует найти свойство `ShowText` в инспекторе свойств соответствующего штрихкода и установить его значение в `False`.

> Свойство `AutoSize` регулирует автоматическую подстройку размеров кода в зависимости от размера объекта. Если требуется ручное изменение размеров баркода с помощью мыши, необходимо отключить это свойство (установить в `False`). В таком случае важно самостоятельно следить за пропорциями по горизонтали и вертикали.

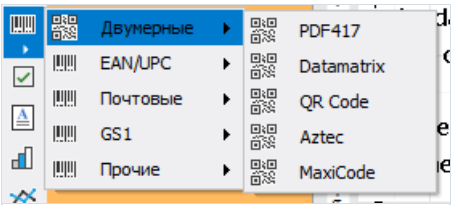
Swiss QR

Swiss QR является одной из разновидностей QR-кодов. В Швейцарии, как и во всем мире, QR-коды используются повсеместно. Однако, для использования этих кодов в сфере электронных платежей было принято решение о создании собственной разновидности QR. Теперь все квитанции об оплате и счета в Швейцарии должны иметь QR-код с швейцарским крестом в центре – это отличительный признак Swiss QR.

FastReport .NET поддерживает Swiss QR code. Сам код выглядит практически также, как и обычный QR, с той лишь разницей, что в его центре располагается швейцарский крест:

Суть этой разновидности кода в том, что в нем шифруется информация об оплате.

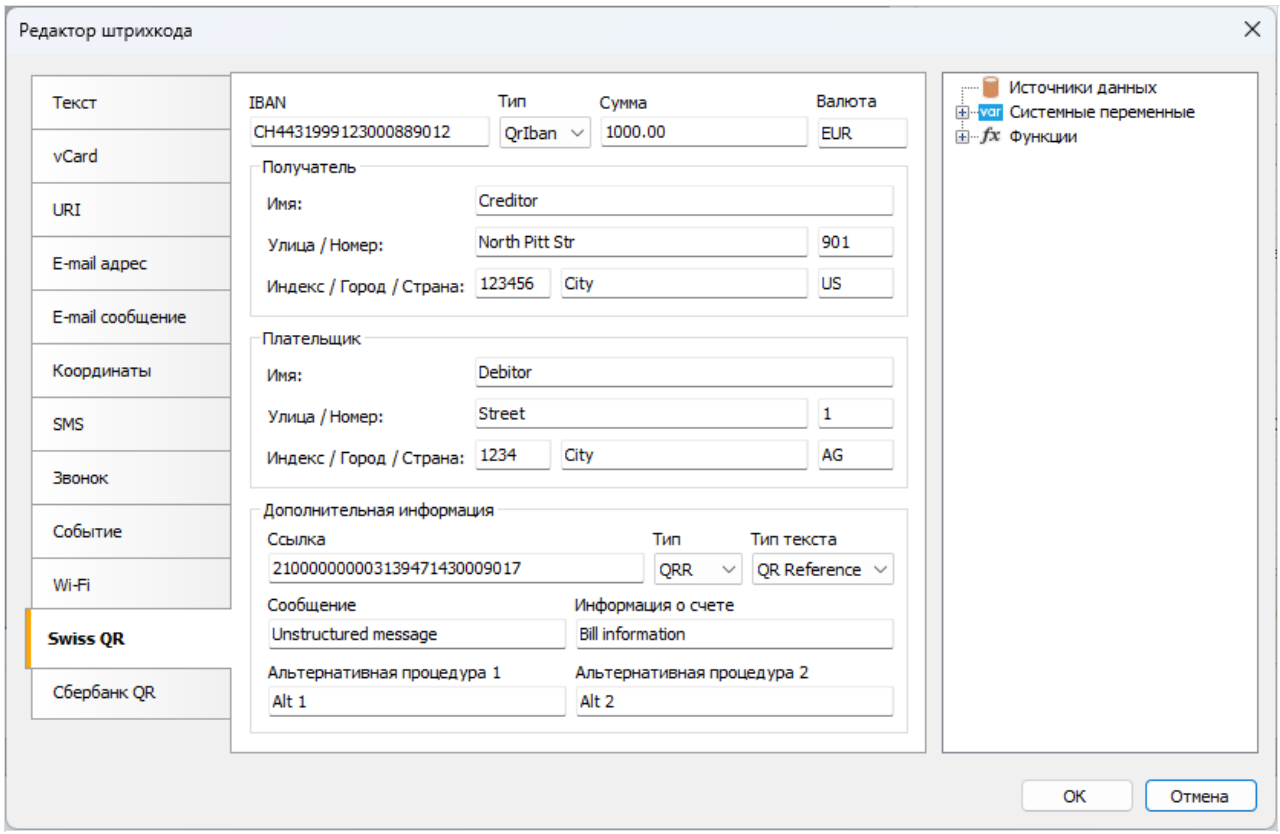
Для формирования Swiss QR в FastReport .NET выберите объект Штрихкод (Barcode) на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Двумерные", а затем QR Code:



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши.

В редакторе QR-кода необходимо перейти на вкладку Swiss QR. После этого будут отображены настройки для Swiss QR:



Рассмотрим параметры подробнее:

Iban

В Швейцарии используется стандарт IBAN (*International Bank Account Number*) для представления номера банковского счета. Это международный стандарт, который зарегистрирован в ISO под номером 13616.

Здесь вы можете выбрать один из двух типов `Iban` или `QRIban`.

`QRIban` должен использоваться для платежей с QR-ссылкой. При этом `QRIban` тоже соответствует стандарту ISO 13616. Каждое юридическое учреждение, которое участвует в схеме, получает свой идентификатор в диапазоне 30000 – 31999. Этот идентификатор называется QR-IID и он включен в состав `QR-Iban`.

Creditor

Заполняем данные о выставителе счета. Название организации и адрес.

Reference

Ссылка на платёж плательщика, которая нужна получателю платежа.

Type

- QRR – QR-ссылка: швейцарский стандарт ссылки длиной 26 символов (только цифры);
- SCOR – Ссылка кредитора: международный стандарт длиной от 5 до 25 символов;
- NON – ссылка может быть пустой.

Text type

- QR-Reference – используется с типом ссылки QRR;
- ISO 13616 – используется с типом ссылки SCOR.

QR-ссылка является заменой для используемой сейчас ссылки ISR. Она помогает прейти на новые QR счета с используемых ранее красных и оранжевых квитанций.

Debitor

Данные о плательщике – ФИО или наименование организации и адрес.

Additional Information

Выставитель счета может ввести какую-либо дополнительную структурированную/неструктурированную информацию для плательщика.

Currency

Так как система оплаты Швейцарская, то предполагается два типа валюты: евро (EUR) и швейцарские франки (CHF).

Alternative Procedure 1 и Alternative Procedure 2

Предполагается, что в будущем выставители счета могут предлагать альтернативные банковским переводам процедуры. Для этого предусмотрено два поля в Swiss QR.

Amount

В этом поле указывается сумма оплаты, разделителем является запятая.

Как правило Swiss QR используется в документе Swiss bill. Пример отчета с таким Swiss счетом вы можете найти в папке `Demos/Reports/` .

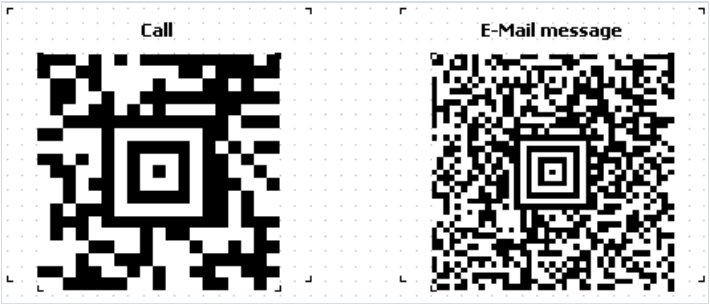
Aztec

Aztec Code является одной из разновидностей QR-кодов. Ацтеки – название племени индейцев из центральной Америки. Если внимательно посмотреть на код, то в его центре можно заметить квадрат, который похож на пирамиду Ацтеков, если смотреть на нее сверху. Это специальная мишень, по которой можно определить центр кода и его ориентацию.

Aztec Code сочетает в себе лучшие идеи двумерных штрихкодов: MaxiCode, SuperCode, CodeOne, DataMatrix, DotCode и PDF417. Несмотря на патент, эта разработка стала достоянием общественности. Стандарт, описывающий кодирование, изложен в ISO/IEC 24778:2008.

Размер кода зависит от объёма кодируемой информации. Например, минимальный размер 15x15 пикселей позволяет закодировать 6 байт, то есть 12 букв или 13 цифр. А максимальный размер 151x151 пиксель позволяет закодировать 1914 байт, 3067 букв или 3832 цифры.

Необходимо учесть, что код имеет два формата отображения: Compact (Компактный), в котором символ с мишенью состоит из двух квадратов, и Full-Range (Полный), в котором символ с мишенью состоит из трёх квадратов. Выбор формата зависит от объёма кодируемых данных.



Преимуществом данного типа кодирования перед другими является возможность считывания кода при любой его ориентации. Даже зеркально отраженный код будет без труда прочитан, благодаря использованию навигационных маркеров.

Использование мишени в центре кода позволяет считывать информацию даже с искаженных или растянутых изображений.

Благодаря использованию алгоритма кодирования Рида-Соломона, Aztec Code может быть прочитан даже при частичном повреждении. В этом случае в код специально закладывается избыточность. Пользователю

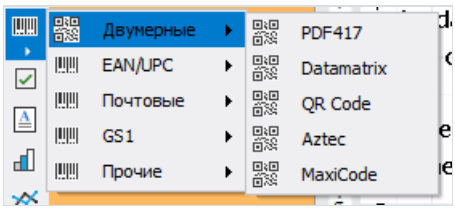
предоставляется возможность регулировать процент избыточного кода от 5 до 95, что обеспечивает высокую степень устойчивости к ошибкам чтения.

Послойная структура кода даёт возможность увеличивать объем хранимой информации путём увеличения области кодирования.

Все эти преимущества сделали Aztec Code очень привлекательным для применения в транспортных сетях в качестве электронных билетов, например, в авиа- и железнодорожных перевозках. В ряде стран этот код также используется в правительственной документации. Кроме того, как и другие коды высокой плотности, Aztec-коды популярны в коммерции, логистике, производстве и фармацевтике.

По сравнению с QR-кодом, Aztec Code имеет большую плотность записи и не требует наличия полей вокруг кода. Также, минимальный размер Aztec Code – 15x15, против 21x21 у QR.

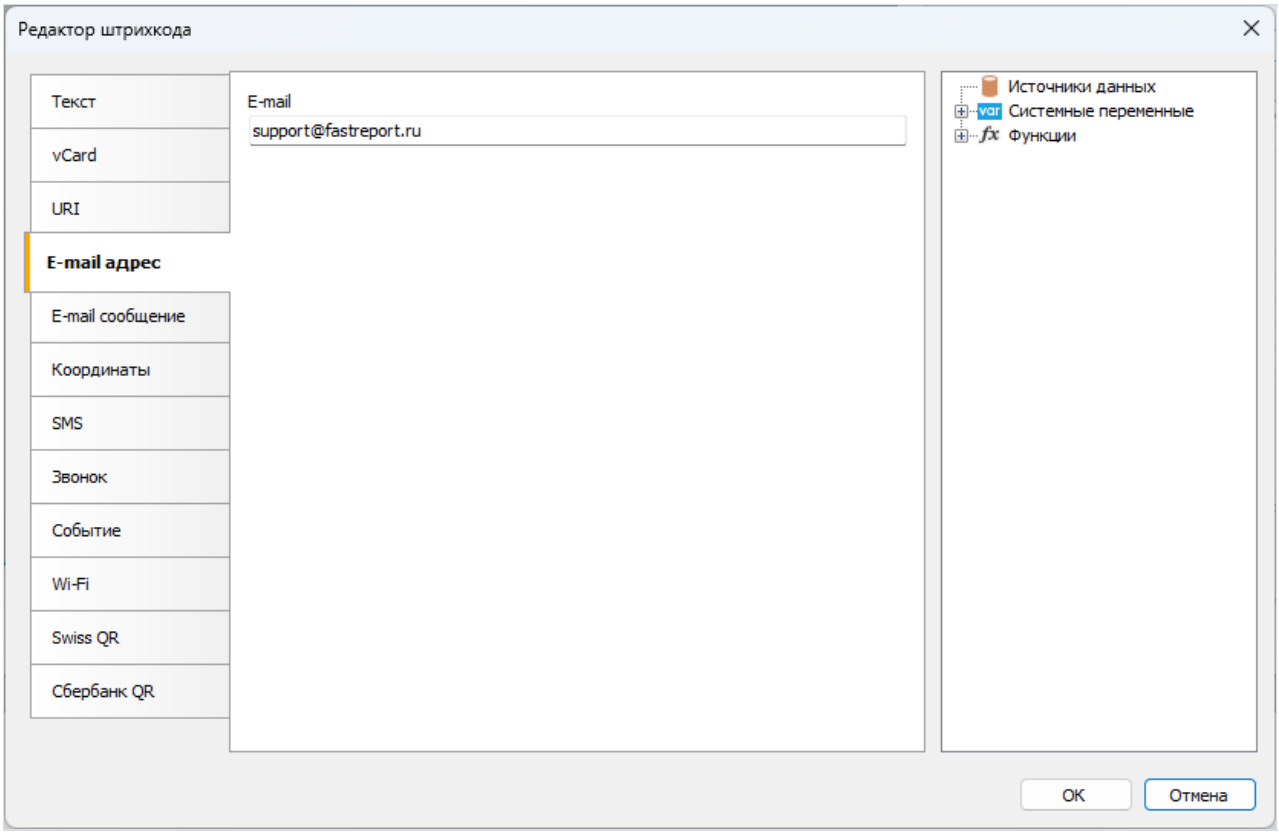
Для формирования штрихкода Aztec Code в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Двумерные", а затем Aztec:



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши.

В редакторе штрихкода можно выбрать шаблон для кодирования информации. Все шаблоны, кроме шаблона SwissQR, могут быть использованы в Aztec Code. Например, адрес электронной почты:



В итоге штрихкод будет выглядеть так:

MaxiCode

Двумерный штрихкод MaxiCode имеет уникальный дизайн, который невозможно перепутать с другими. Он отличается круглой мишенью для точного позиционирования при сканировании. Кодирующие элементы представлены точками, которые формируют специальные комбинации и размещены в гексагональной (сотовой) структуре. Эта сетка состоит из 888 ячеек, которые несут информацию.

MaxiCode позволяет закодировать 138 цифровых символа или 93 алфавитных. Штрихкод имеет фиксированный размер 1,11 x 1,054 дюйма. Предусмотрена коррекция ошибок чтения при повреждении кода, основанная на коде Рида-Соломона. Поскольку большинство таких штрихкодов находятся на упаковках, они должны быть устойчивы к повреждениям. Коррекция ошибок допускает повреждение до 1/8 части кода.

Каждый штрихкод Maxicode включает в себя два основных сообщения: первичное и вторичное. Первичное сообщение включает почтовый индекс, код страны и класс, а вторичное сообщение содержит данные адреса.

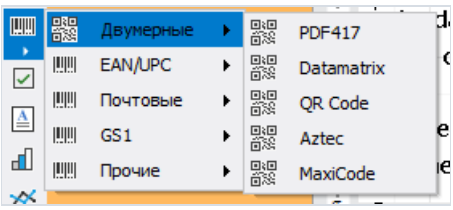
Для считывания штрихкода Maxicode требуется 2D-сканер штрихкода, предпочтительно тот, который выполняет эмуляцию клавиатуры и получает питание от USB-порта, поэтому внешний источник питания не требуется.

Основным применением MaxiCode является маркировка пакетов, поддонов и т. д. Он обеспечивает наличие критически важной информации в любой момент, когда груз обрабатывается.

MaxiCode был разработан для включения в существующий манифест систем доставки. Компактный размер позволяет также использовать MaxiCode для замены символов меньшей плотности, таких как линейные штрихкоды.

Одной из ключевых особенностей MaxiCode является его способность к быстрому считыванию в широком поле зрения. Это означает, что данный штрихкод можно успешно применять в автоматизированных системах обработки информации.

Для формирования штрихкода MaxiCode в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Двумерные", а затем MaxiCode:

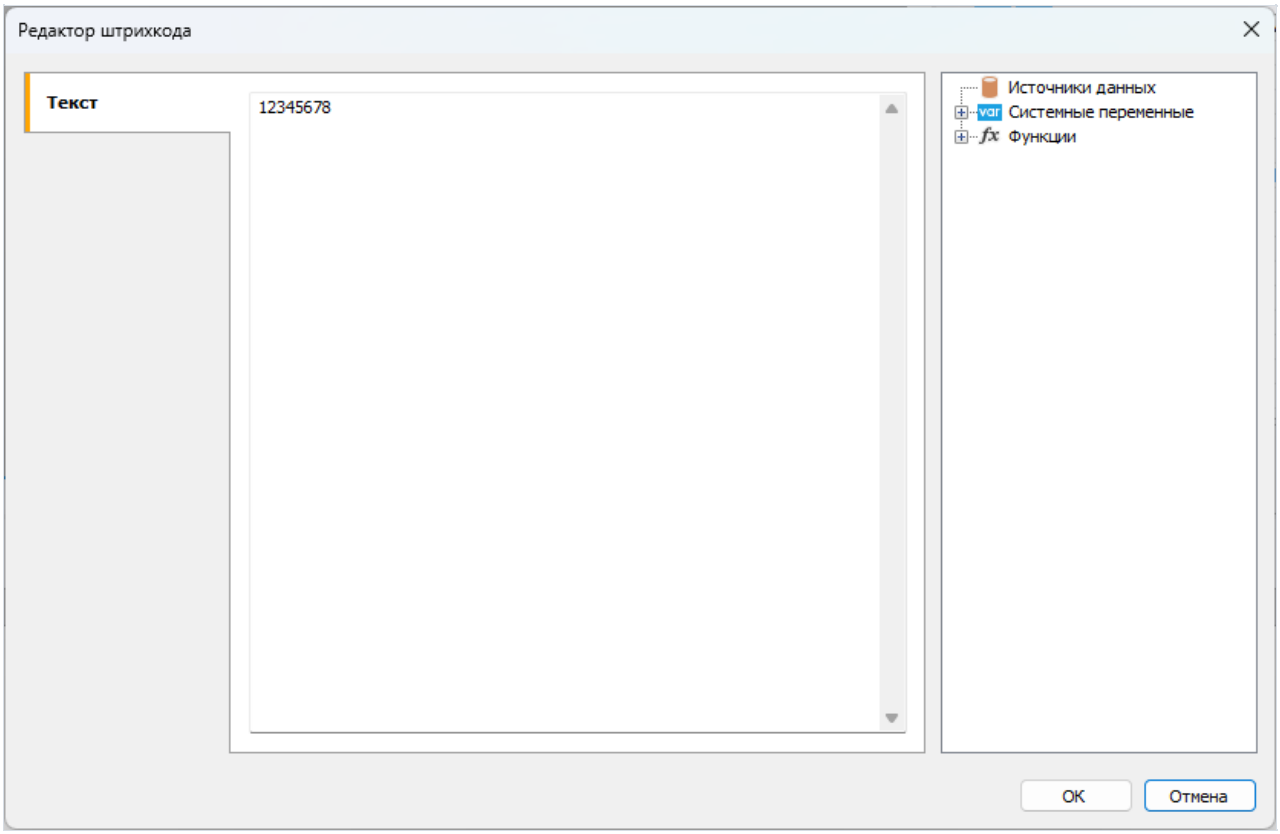


После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно

открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши.

В редакторе штрихкода можно ввести циферно-буквенную информацию:



Также, как и все другие штрихкоды в FastReport, MaxiCode может быть отображен как с текстовой информацией под кодом, так и без неё. За это отвечает свойство `ShowText`.

EAN-8

Код EAN-8 (European Article Number), известный также, как GTIN-8, представляет собой короткий штриховой код размерностью 8 цифр. Цифры разделены на 2 блока по 4 цифры. Первые 2 цифры идентифицируют страну происхождения товара, затем 5 цифр – кодируемая информация, а последний – это контрольная сумма для проверки целостности данных.

Этот код был создан на основе EAN-13 путем сокращения его размерности до 8 цифр. Это позволило размещать код на маленьких упаковках товаров. Благодаря наличию идентификации страны в коде, он приобрел интернациональное значение.

Область применения EAN-8: маркировка товаров и оборудования.

Каждая цифра кодируется 7 модулями, представленными вертикальными линиями и пробелами. Если принять, что 0 обозначается пробелом, а 1 – линией, то кодировка цифр выглядит следующим образом:

- 0 – 0001101;
- 1 – 0011001;
- 2 – 0010011;
- 3 – 0111101;
- 4 – 0100011;
- 5 – 0110001;
- 6 – 0101111;
- 7 – 0111011;
- 8 – 0110111;
- 9 – 0001011.

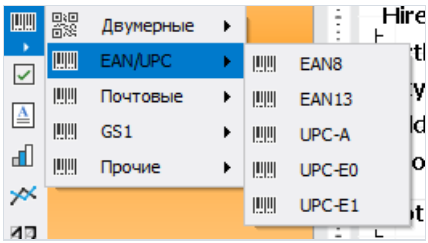
Убедимся на примере:

Напомним, что код начинается с последовательности 101, за которой следует код для цифры 3 и так далее. Как упоминалось ранее, код разделен на две части. Разделитель, подобно началу и концу кода, имеет более длинные линии. Он представлен кодом 01010.

Если присмотреться, можно заметить, что после разделителя линии не соответствуют цифрам из приведенного списка. Это происходит потому, что для второй части используется другая кодировка:

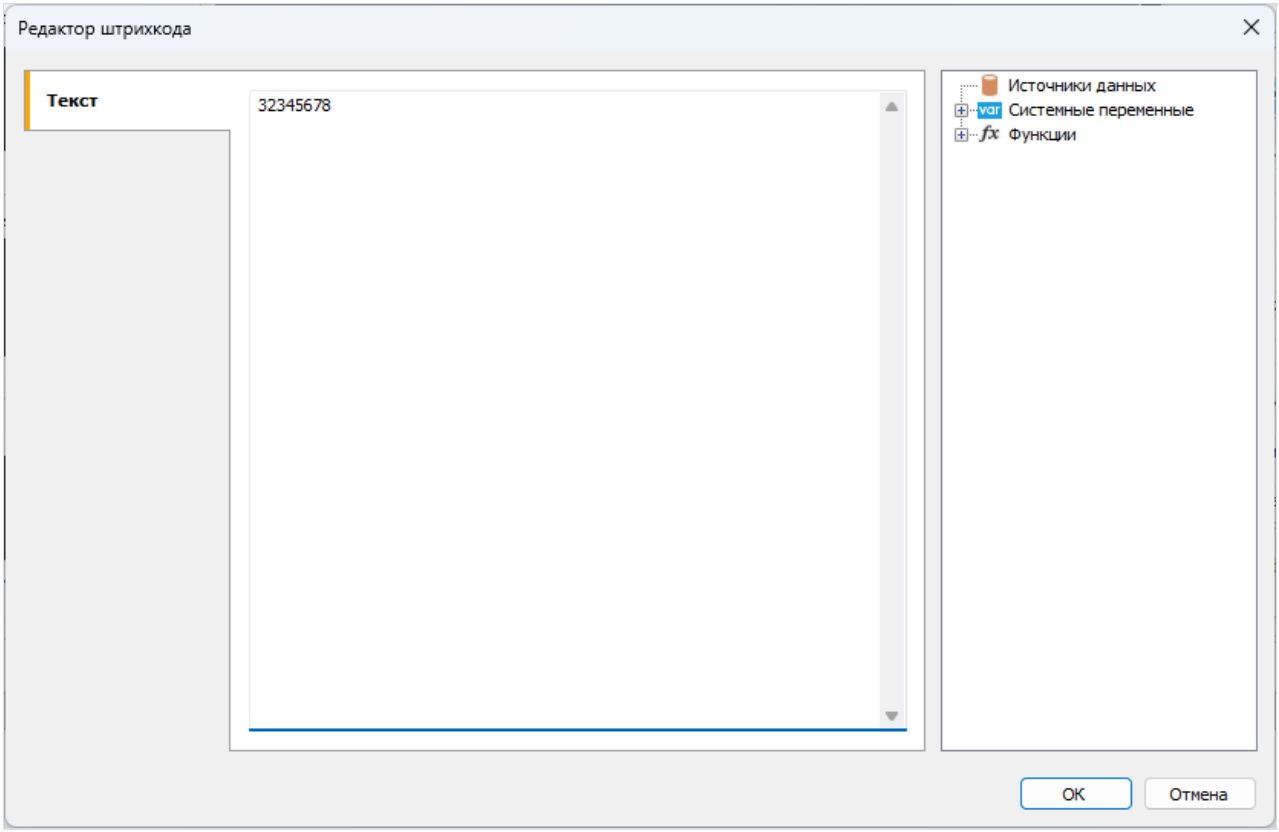
- 0 – 1110010;
- 1 – 1100110;
- 2 – 1101100;
- 3 – 1000010;
- 4 – 1011100;
- 5 – 1001110;
- 6 – 1010000;
- 7 – 1000100;
- 8 – 1001000;
- 9 – 1110100.

Для формирования штрихкода EAN-8 в FastReport .NET выберите объект Штрихкод (Barcode) на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "EAN/UPC", а затем EAN8:



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Если необходимо скрыть текст под штрихкодом, следует найти свойство `ShowText` в инспекторе свойств соответствующего штрихкода и установить его значение в `False`:

EAN-13

Штрихкод EAN-13 (European Article Number) является самым распространенным среди линейных символик. С помощью этой системы знаков можно закодировать 12 цифр, а 13-я цифра кода – контрольная сумма для проверки целостности кода. Допускается кодирование только цифр.

Структура кода такова:

- первые 2-3 цифры кода выделены для кодирования страны производителя товара;
- далее 4-5 цифр выделено под кодирование производителя товара;
- оставшиеся 3-5 цифр – номер продукта на предприятии.

Как уже говорилось, 13-я цифра – контрольная. Она рассчитывается автоматически на основе предыдущих 12 по особому алгоритму. Благодаря этой цифре можно определить правильно ли прочитан код.

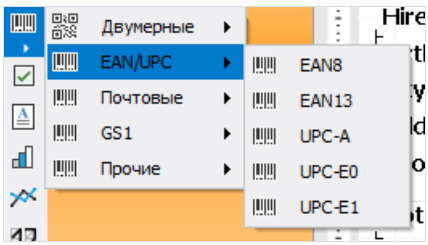
Если же рассматривать отображение кода, то можно заметить, что в начале и конце кода присутствует штриховая последовательность из двух линий. Если представлять код в виде двоичной последовательности, где линии обозначены как 1, а пробелы как 0, то эта штриховая последовательность выглядит как 101. Кроме того, штрихкод разделен на две части ровно посередине такой же последовательностью.

Каждая цифра кодируется 7 модулями (линиями и пробелами). Существуют три специальные таблицы с кодовыми последовательностями для цифр. Причем первая и вторая половина кода использует разные варианты таких таблиц.

Этот код может читаться в любой последовательности: как слева направо, так и справа налево. Его простота и удобство чтения быстро сделали его популярным в розничной торговле для маркировки товаров.

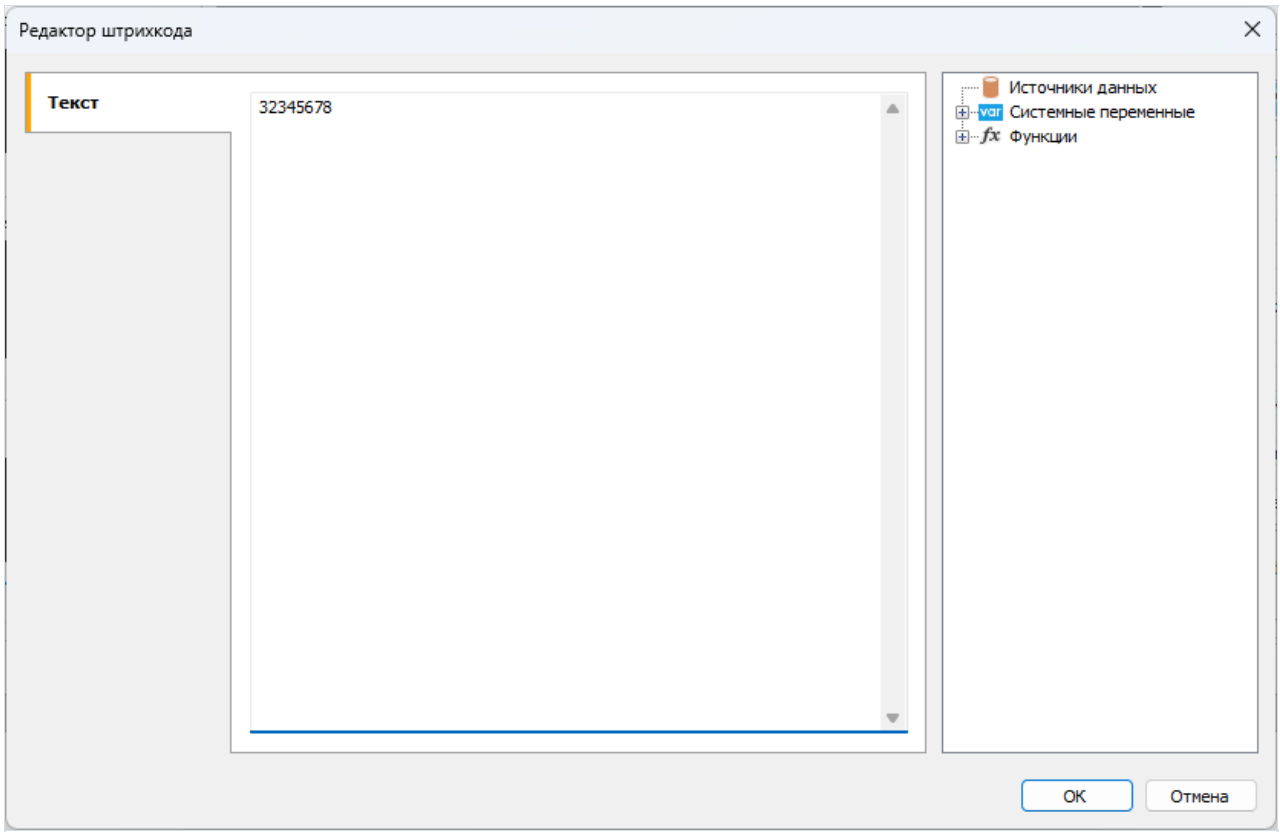
К недостаткам этого кода часто относят его малую емкость – длину кодируемой последовательности.

Для формирования штрихкода EAN-13 в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "EAN/UPC", а затем EAN13:



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Если необходимо скрыть текст под штрихкодом, следует найти свойство `ShowText` в инспекторе свойств соответствующего штрихкода и установить его значение в `False`:

UPC-A

Universal Product Code – это линейный штрихкод, который позволяет кодировать 12 символов. Он разработан совместно компаниями Uniform Grocery Product Code Council и IBM в 1973. По своей структуре и назначению он похож на известный код EAN-13. Коды UPC разработаны для Северной Америки, а EAN – для Европы.

Структура кода:

- старт-символ, обозначающий начало кода;
- префикс, который обозначает вид продукции – 1 символ;
- код производителя – 5 символов;
- код товара – 5 символов;
- контрольная цифра – 1 символ. Рассчитывается на основе предыдущих 11 цифр по формуле Modulo 10;
- стоп-символ, обозначающий конец кода.

Перед кодом и после него должно быть пустое пространство размером около 9 модулей. Это нужно для гарантии распознавания кода сканером.

Длина кода 12 символов, из которых только 11 кодированные данные, и еще один – контрольная цифра.

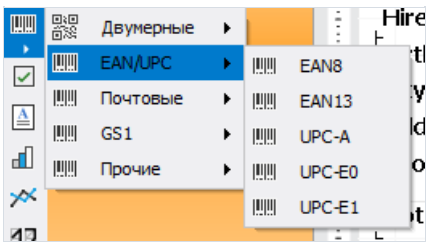
Каждый символ кодируется двумя штрихами и двумя пробелами. Штрих или пробел может быть шириной 1, 2, 3 или 4 модуля (один модуль равен 0,33мм).

Штрихкод UPC-A очень распространен в США и Канаде. Он используется в супермаркетах для маркировки продуктов.

Этот штрихкод заслужил большую популярность благодаря компактным размерам, простоте считывания сканерами и наличию контрольной цифры для защиты от ошибок считывания.

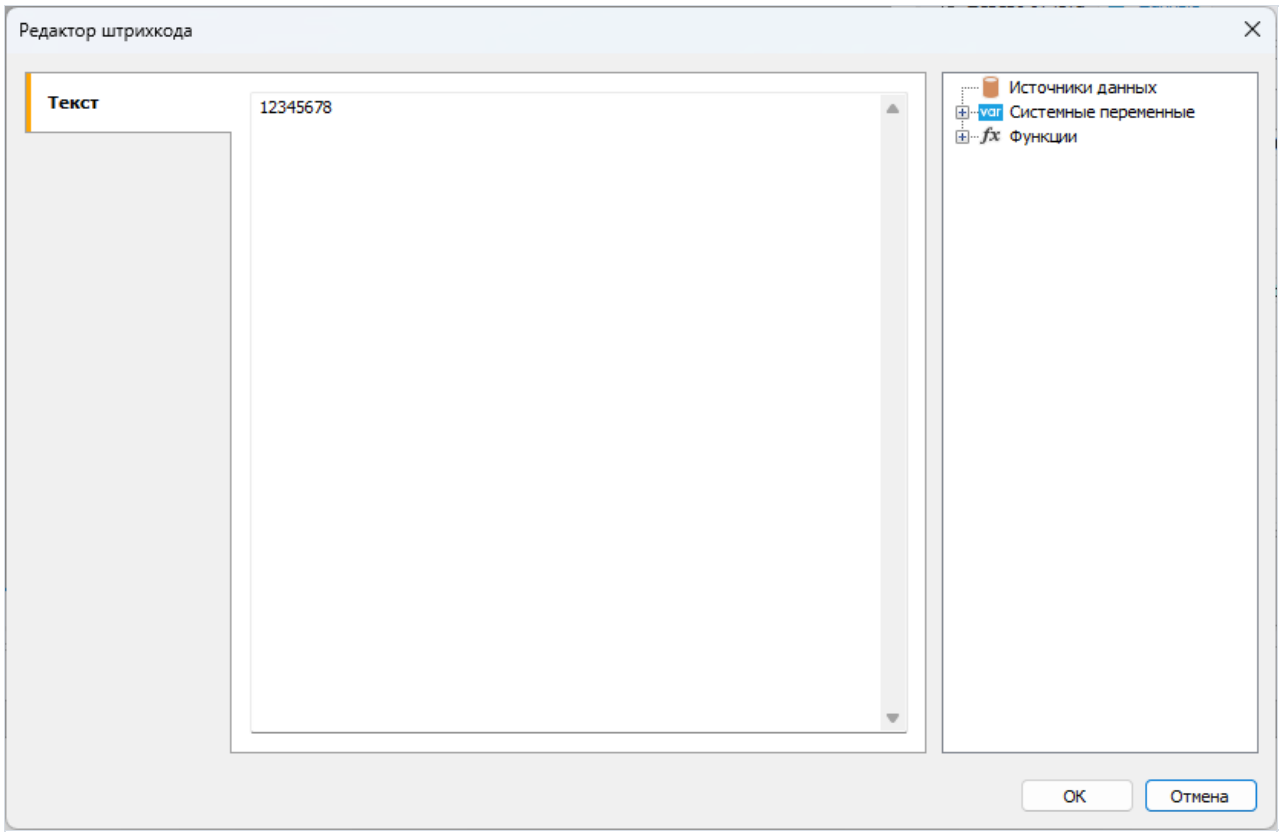
К недостаткам кода можно отнести возможность кодировать только цифры, также небольшую размерность кода, что ограничивает сферу его применения.

Для формирования штрихкода UPC-A в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "EAN/UPC", а затем UPC-A:



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:

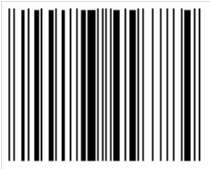


Все объекты типа Штрихкод (Barcode) имеют набор свойств. Вот наиболее часто используемые из них:

Свойство	Описание
Поворот (Angle)	Позволяет задать поворот объекта на один из фиксированных углов – 0, 90, 180, 270 градусов.
Масштаб (Zoom)	Задает масштабирование штрихкода. Это свойство используется только вместе со свойством "Авторазмер".
Авторазмер (AutoSize)	Если это свойство включено, объект будет растягиваться, чтобы показать штрихкод целиком. Если свойство отключено, штрихкод будет растянут до размеров объекта.
Показывать текст (ShowText)	Определяет, надо ли показывать ли текст в нижней части штрихкода.
Поле данных (DataColumn)	Поле данных, из которого загружать текст объекта.
Выражение (Expression)	Выражение, которое возвращает текст объекта.
Текст (Text)	Текст объекта.
Отступы (Padding)	Позволяет задать отступы от краев объекта, в пикселях.
Ширина полос (WideBarRatio)	Это свойство имеется у всех линейных штрихкодов. Оно определяет относительный размер широких полос штрихкода.
Контрольная сумма (CalcCheckSum)	Это свойство имеется у многих линейных штрихкодов. Оно определяет, надо ли считать контрольную сумму автоматически. Если это свойство отключено, контрольная сумма должна присутствовать в тексте объекта.

Свойство	Описание
Отображение вертикальных полос (DrawVerticalBearerBars)	Если это свойство включено, то у объекта будут отображаться боковые линии.

Если отключить свойство `ShowText`, код будет выглядеть следующим образом:



UPC-E

UPC-E – это линейный цифровой штрихкод, созданный для кодирования информации о товарах в розничной торговле. Как вы знаете, стандарт UPC основан на EAN. А именно, американский UPC-A является аналогом европейского EAN-13. Основное различие заключается лишь в длине кода: 12 символов против 13. У EAN-13 существует укороченная версия EAN-8, созданная для более компактного размещения на маленьких упаковках. Аналогично, у UPC-A есть «облегченная» версия UPC-E, состоящая из 6 символов. Как правило, коды продукта и товара, используемые в UPC-A, содержат много нулей. Путем удаления нулей из этих кодов удалось сократить длину штрихкода с 12 до 6 символов без потери информации.

Структура кода такова:

- первый символ-префикс обозначает систему счисления 0 или 1. От этого зависит как будет интерпретирован код;
- далее 6 символов производителя и кода продукта (в UPC-A они разделены);
- последний символ – контрольная цифра для проверки целостности кода.

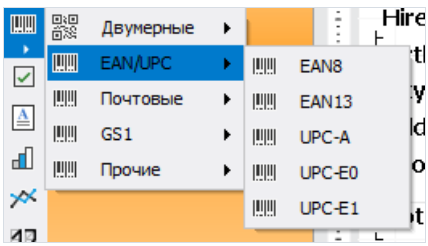
UPC-E фактически может кодировать от 6 до 8 символов. При кодировании минимального набора из 6 символов исключается первый символ, обозначающий систему счисления, а также последний символ – контрольная цифра. При кодировании 7 символов исключается только контрольная цифра. Восьмисимвольный код включает в себя всю вышеописанную структуру кода. Укороченная версия UPC лишена старт- и стоп-символов – в пользу компактности.

В зависимости от того, какая система счисления выбрана, автоматически рассчитывается контрольная цифра. Таким образом, стандарт UPC-E разделяется на UPC-E0 и UPC-E1.

Важно отметить, что UPC-E позволяет кодировать только структуры GTIN-12 с начальным символом 0 и последовательностью нулей в коде. Эта последовательность нулей в итоге и будет упразднена в коде UPC-E.

Также, как и в UPC-A, каждый символ данных в UPC-E кодируется двумя штрихами и двумя пробелами. Штрих или пробел может быть шириной 1, 2, 3 или 4 модуля (один модуль равен 0,33 мм).

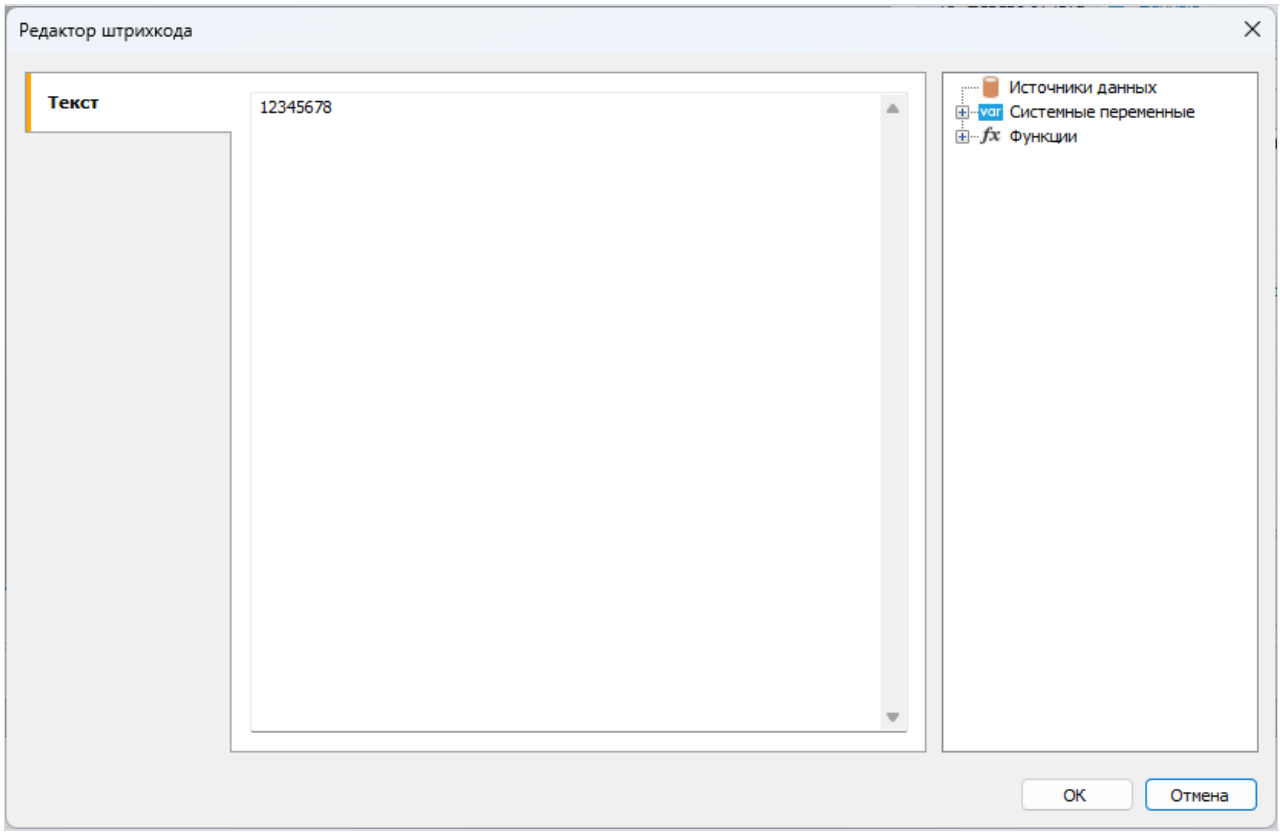
Для формирования штрихкодов UPC-E0 или UPC-E1 в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "EAN/UPC", а затем UPC-E0 или UPC-E1:



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием

правой кнопки мыши:

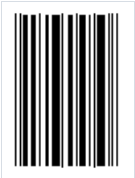


Все объекты типа Штрихкод (Barcode) имеют набор свойств. Вот наиболее часто используемые из них:

Свойство	Описание
Поворот (Angle)	Позволяет задать поворот объекта на один из фиксированных углов – 0, 90, 180, 270 градусов.
Масштаб (Zoom)	Задает масштабирование штрихкода. Это свойство используется только вместе со свойством "Авторазмер".
Авторазмер (AutoSize)	Если это свойство включено, объект будет растягиваться, чтобы показать штрихкод целиком. Если свойство отключено, штрихкод будет растянут до размеров объекта.
Показывать текст (ShowText)	Определяет, надо ли показывать ли текст в нижней части штрихкода.
Поле данных (DataColumn)	Поле данных, из которого загружать текст объекта.
Выражение (Expression)	Выражение, которое возвращает текст объекта.
Текст (Text)	Текст объекта.
Отступы (Padding)	Позволяет задать отступы от краев объекта, в пикселях.
Ширина полос (WideBarRatio)	Это свойство имеется у всех линейных штрихкодов. Оно определяет относительный размер широких полос штрихкода.

Свойство	Описание
Контрольная сумма (CalcCheckSum)	Это свойство имеется у многих линейных штрихкодов. Оно определяет, надо ли считать контрольную сумму автоматически. Если это свойство отключено, контрольная сумма должна присутствовать в тексте объекта.
Отображение вертикальных полос (DrawVerticalBearerBars)	Если это свойство включено, то у объекта будут отображаться боковые линии.

Если отключить свойство `ShowText`, код будет выглядеть следующим образом:



Deutsche Post Identcode

Линейный штрихкод Deutsche Post Identcode создан на основе популярного штрихкода 2 of 5 Interleaved и использует его символогию. Различия заключаются в назначении кода. Identcode создан специально для Deutsche Post с целью идентификации почтового отправителя. Исходя из необходимого размера данных об отправителе, длина кода уменьшена до 12 цифр по сравнению с 14 у Interleaved. Однако фактическая длина кодируемых данных составляет 13 символов, так как 13-й используется для контрольной суммы, необходимой для проверки правильности считывания кода.

Identcode имеет следующую структуру:

- старт-символ;
- 2 цифры для идентификатора первичного центра распределения;
- 3 цифры для идентификатора клиента;
- 6 цифр для почтового номера;
- контрольная цифра;
- стоп-символ.

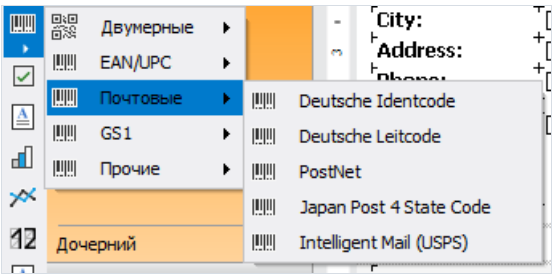
Если представить штрихи и пробелы в виде двоичных символов, то стартовый символ обозначается так: 1010. Этот символ помогает определить направление считывания.

Для кодирования цифр данных используется кодовая таблица 2 of 5 Interleaved. Каждый символ данных состоит из 5 штрихов или 5 пробелов, из которых два должны быть широкими, а остальные – узкими. Важно отметить, что все коды семейства 2 of 5 используют для кодирования как штрихи, так и пробелы. Поэтому длина кода всегда четная, так как один набор из 5 штрихов и 5 пробелов между ними кодирует сразу две цифры. Это обеспечивает высокую плотность и компактность кода.

После 11 цифр данных следует контрольная цифра, которая рассчитывается по алгоритму Modulo 10.

Завершает код стоп-символ, который кодируется последовательностью 101.

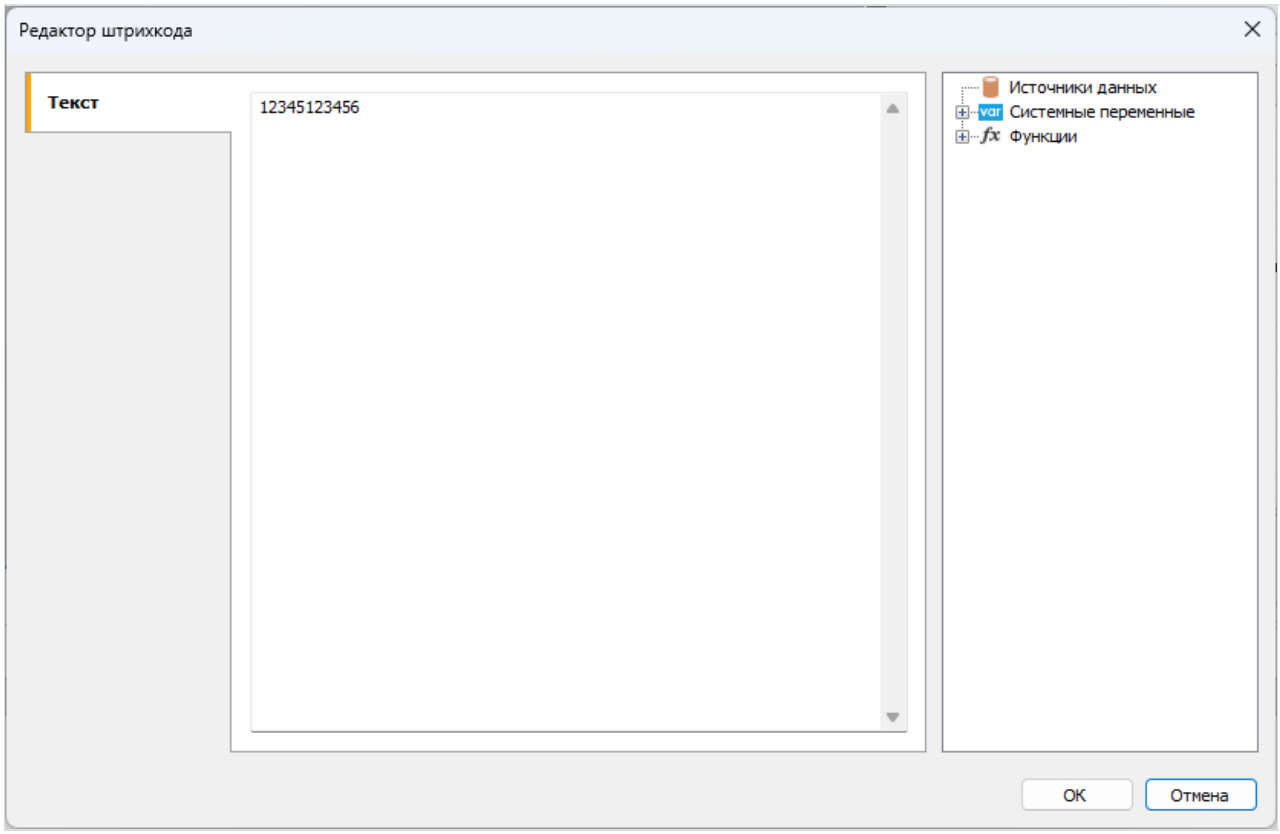
Для формирования штрихкода Deutsche Post Identcode в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Почтовые", а затем Deutsche Identcode:



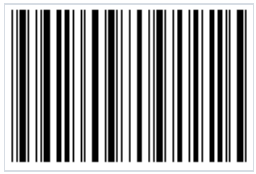
После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием

правой кнопки мыши:



Если необходимо скрыть текст под штрихкодом, следует найти свойство `ShowText` в инспекторе свойств соответствующего штрихкода и установить его значение в `False`:



Deutsche Post Leitcode

Линейный штрихкод Deutsche Post Leitcode, или просто LeitCode, разработан на основе широко известного кода 2 of 5 Interleaved. Он позволяет закодировать 13 цифровых символов (0-9), применяется в почтовой службе Германии (Deutsche Post) и в DHL для автоматизации сортировки почтовых отправлений. Отличительной особенностью Leitcode является его четко заданная структура данных, что выделяет его среди штрихкодов 2 of 5 Interleaved.

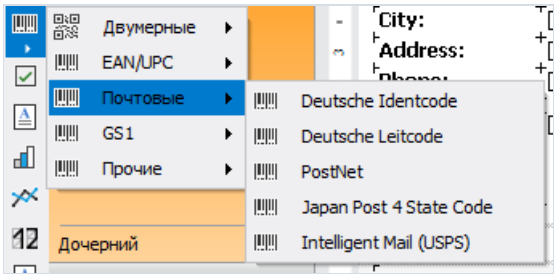
Структура кода:

- старт-символ;
- 1-5 – ZIP код;
- 6-8 – номер улицы;
- 9-11 – номер дома;
- 12-13 – код продукта;
- 14 – контрольная цифра;
- стоп-символ.

Другое отличие Leitcode от 2 of 5 Interleaved заключается в использовании алгоритма Modulo 10 для расчета обязательной контрольной суммы.

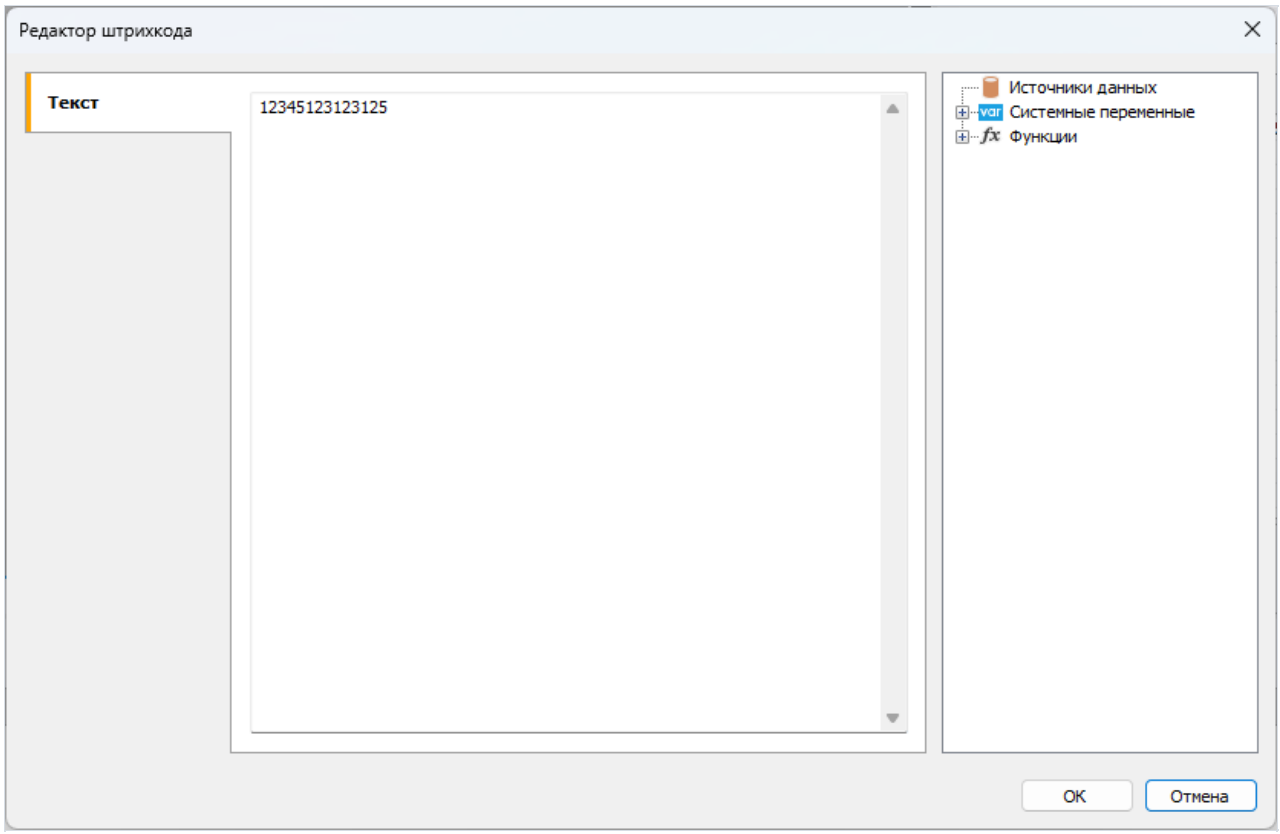
Как и в случае с штрихкодами 2 of 5, в Leitcode для кодирования данных используются как штрихи, так и пробелы. Этот подход помог уменьшить размер кода. Символ данных в Leitcode кодируется пятью модулями (штрихами/пробелами), где два штриха являются широкими, а три – узкими. То же самое относится и к пробелам. Именно благодаря этому сочетанию штрихкод получил название 2 of 5.

Для формирования штрихкода Deutsche Post Leitcode в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Почтовые", а затем Deutsche Leitcode:

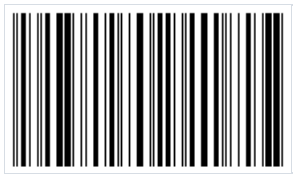


После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Если необходимо скрыть текст под штрихкодом, следует найти свойство `ShowText` в инспекторе свойств соответствующего штрихкода и установить его значение в `False`:



PostNet

Этот линейный числовой штрихкод создан специально для использования в почтовой службе Соединенных Штатов Америки и предназначен для машинной сортировки почты. Внешний вид этого штрихкода сильно отличается от привычных UPC и EAN, которые применяются для маркировки продуктов в розничной торговле. Штрихи имеют разную высоту, и в целом код также имеет небольшую высоту для удобства размещения на конверте.

Длина ZIP-кода в США изначально составляла 5 символов, но в 1983 году была увеличена до 9 символов из-за недостаточной емкости. Но затем код расширили еще на 2 символа. В итоге PostNet имеет три размера:

- 5 символов;
- 9 символов;
- 11 символов.

Структура кода:

- стартовый символ;
- данные;
- проверочный символ;
- стоп-символ.

Каждый символ кодируется 5 штрихами. Старт- и стоп-символы обозначаются одним длинным штрихом.

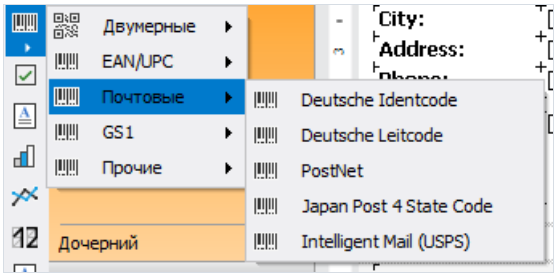
Далее приводится список соответствия чисел и штрихкодов:

1. ;
2. ;
3. ;
4. ;
5. ;
6. ;
7. ;
8. ;
9. ;
10. .

Из-за достаточно большого количества штрихов, необходимых для кодирования одного символа, размер штрихкода PostNet довольно велик. Кроме того, он позволяет кодировать только числа. Эти недостатки привели к снижению его популярности, и в настоящее время он постепенно вытесняется более современным решением – Intelligent Mail.

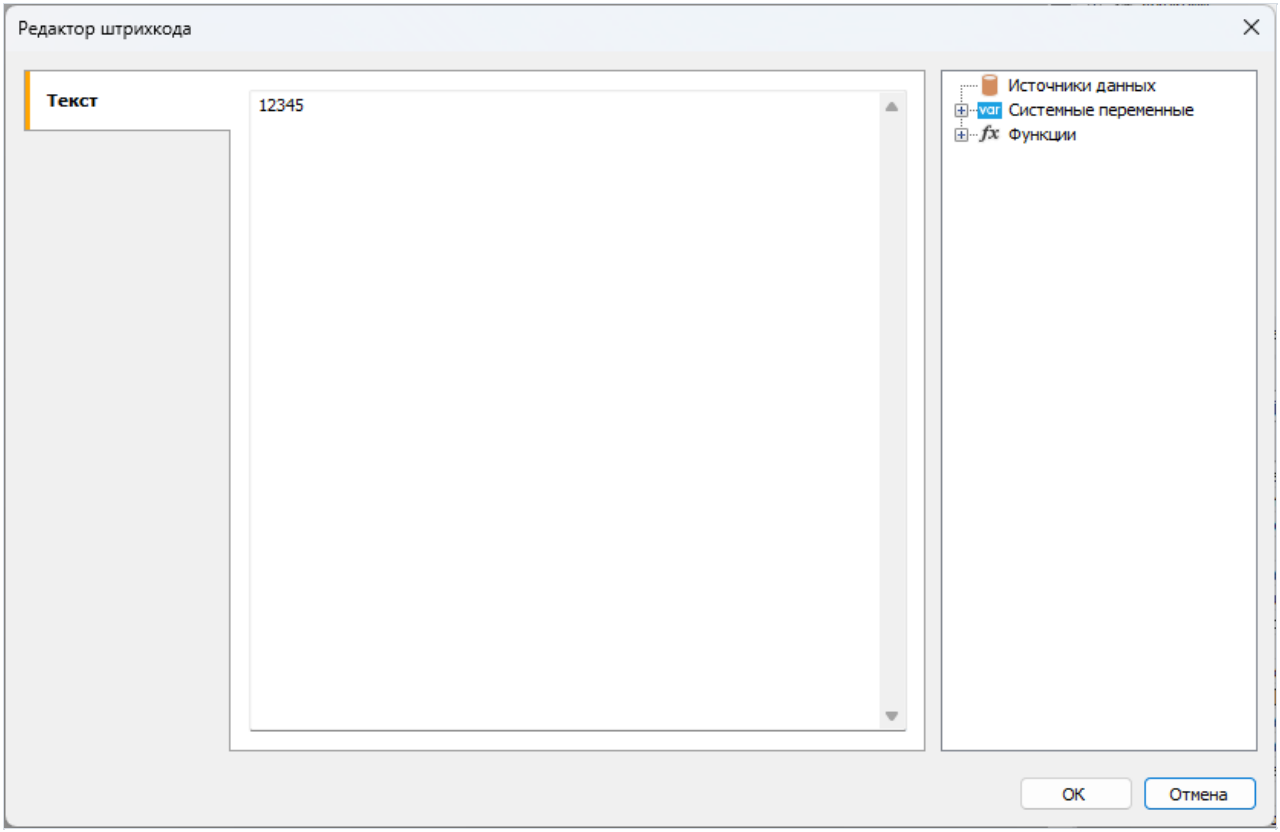
В FastReport .NET для штрихкода PostNet нет ограничений на количество символов. Однако стоит помнить, что стандарт предусматривает три варианта размерности кода, которые были рассмотрены ранее.

Для формирования штрихкода PostNet в FastReport .NET выберите объект Штрихкод (Barcode) на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Почтовые", а затем PostNet:



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Как и все штрихкоды в FastReport .NET, PostNet код имеет ряд свойств, которые можно отредактировать в инспекторе свойств объекта:

Свойство	Описание
Поворот (Angle)	Позволяет задать поворот объекта на один из фиксированных углов – 0, 90, 180, 270 градусов.
Масштаб (Zoom)	Задает масштабирование штрихкода. Это свойство используется только вместе со свойством "Авторазмер".
Авторазмер (AutoSize)	Если это свойство включено, объект будет растягиваться, чтобы показать штрихкод целиком. Если свойство отключено, штрихкод будет растянут до размеров объекта.

Свойство	Описание
Показывать текст (ShowText)	Определяет, надо ли показывать ли текст в нижней части штрихкода.
Поле данных (DataColumn)	Поле данных, из которого загружать текст объекта.
Выражение (Expression)	Выражение, которое возвращает текст объекта.
Текст (Text)	Текст объекта.
Отступы (Padding)	Позволяет задать отступы от краев объекта, в пикселях.
Ширина полос (WideBarRatio)	Это свойство имеется у всех линейных штрихкодов. Оно определяет относительный размер широких полос штрихкода.
Контрольная сумма (CalcCheckSum)	Это свойство имеется у многих линейных штрихкодов. Оно определяет, надо ли считать контрольную сумму автоматически. Если это свойство отключено, контрольная сумма должна присутствовать в тексте объекта.
Отображение вертикальных полос (DrawVerticalBearerBars)	Если это свойство включено, то у объекта будут отображаться боковые линии.

Intelligent Mail

IMb (Intelligent Mail barcode), также известный как USPS OneCode, представляет собой модулированный по высоте штрихкод, который кодирует до 31 цифры данных почтового отправления в 65 вертикальных полосах, используя символику с 4 состояниями. Эта символика использует четыре разных состояния «баров», что позволяет закодировать больше информации в одном штрихкоде.

Стандарт Intelligent Mail был создан на основе стандартов POSTNET и PLANET, которые использовались в США для почты ранее.

Эти стандарты позволяют кодировать почтовые индексы и служат, главным образом, для сортировки и отслеживания почты.

POSTNET способен кодировать пятизначный почтовый индекс, четырехзначный плюс-код и двузначный код точки доставки.

Штрихкоды POSTNET имеют переменную длину от 32 до 62 штриха, а PLANET – 62 или 72 штриха. Они модулированы по высоте (вертикальные столбцы имеют разную длину) и имеют два состояния (короткие и длинные бары). Каждая цифра закодированных данных представлена группой из пяти полос. У штрихкодов POSTNET в каждой группе из пяти полос содержится ровно два полных штриха, в то время как у штрихкода PLANET их три.

Технология IMb эффективно объединяет функциональные возможности PLANET и POSTNET в одном штрихкоде. Это позволяет почтовым пользователям использовать один штрихкод для участия в нескольких почтовых службах, что расширяет возможности отслеживания отдельных почтовых отправлений и обеспечивает более широкую видимость потока почты. Применение этого штрихкода позволяет Почтовой службе предоставлять несколько услуг для индустрии рассылки, а также дополнительные функции отслеживания почты и мониторинга ее производительности, что способствует сокращению расходов. Как и в случае с POSTNET, IMb имеет контрольную сумму для проверки целостности кода и возможности восстановления поврежденного кода.

По сравнению с POSTNET, IMb обладает гораздо большей емкостью данных (31 символ против 11). Кроме того, помимо кода маршрутизации, в штрихкоде Intelligent Mail содержатся четыре дополнительных поля: идентификатор штрихкода, идентификатор типа услуги (STID), идентификатор почтовой программы (MID) и серийный номер. Эти дополнительные поля позволяют почтовым пользователям определять класс почты, идентифицировать сервисы, которые они хотят получить (например, отслеживание и исправление адреса), а также обеспечивают однозначную идентификацию почтовых отправлений.

К недостаткам можно отнести достаточно большую длину кода и возможность кодировать только числа.

Intelligent Mail позволяет кодировать следующую информацию:

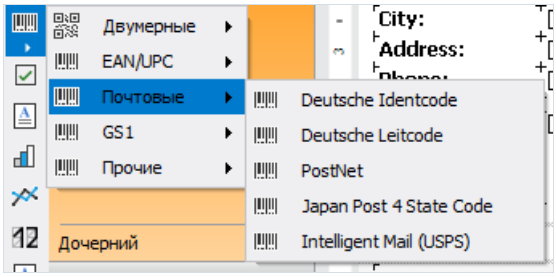
Информация	Описание
Идентификатор штрихкода (2 символа)	Присваивается почтовой службой Соединенных Штатов Америки.

Информация	Описание
Идентификатор типа услуги (3 символа)	Класс почты или другой сервис.
Идентификатор отправителя (6 или 9 символов)	Идентификатор конкретной компании, присвоенный почтовой службой Соединенных Штатов Америки.
Серийный номер (6 или 9 символов)	Почтовому отправителю назначается номер для идентификации конкретного получателя или домашнего хозяйства.
Почтовый индекс точки доставки (11 символов)	Необязательное поле.

Штрихи имеют различную высоту, направление (вверх, вниз) и толщину.

Такой штрихкод может быть напечатан не только на конверте, но и непосредственно на документе, после чего его можно будет считать через специальное прозрачное окно конверта.

Для формирования штрихкода Intelligent Mail в FastReport .NET выберите объект Штрихкод (Barcode) на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Почтовые", а затем Intelligent Mail (USPS):



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши.

В FastReport минимальная длина кода 20 символов. Это обусловлено тем, что с 21-го символа начинается Почтовый индекс точки доставки, который является необязательным.

Изменение всего одной цифры полностью меняет штрихкод:

GS1-128

Этот штрихкод также известен как EAN-128 и UCC-128. Он представляет собой одномерный линейный штрихкод, с возможностью кодировать и числа, и буквы. Наибольшую популярность он получил в сфере логистики при маркировке упаковок, но также применяется и в других сферах. Стандарт GS1 Logistic Label описывает логистическую грузовую этикетку в которой используется код GS1-128.

Для построения кода используется набор идентификаторов Code-128, известных как Application Identifiers (AI, идентификаторы применения). Каждый AI обозначает определённый тип информации и указывается в скобках, например, `(37)`. Знание типа данных, заключённого в скобках, позволяет корректно интерпретировать последующие данные.

Сначала в скобках указывается код идентификатора, затем его значение. Далее, без пробелов следующий код идентификатора и значение. Таким образом код будет иметь значение вида:

```
(01)12646846874672(10)ABC11(15)100420
```

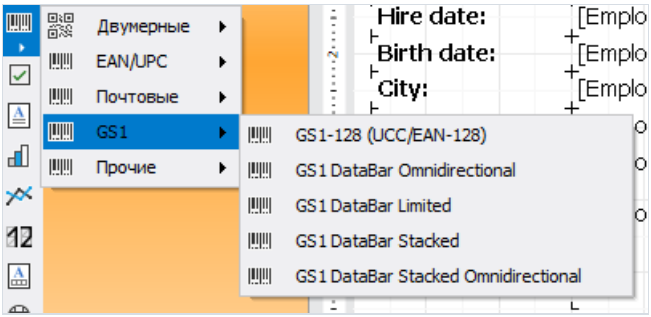
Длина кода ограничивается 48 символами без учета скобок.

Список поддерживаемых идентификаторов типа данных достаточно большой, приведем пример некоторых из них:

Код	Описание	Формат
00	Глобально-уникальный код грузовых контейнеров (Serial Shipping Container Code (SSCC))	N2+N18
01	Глобально-уникальный номер торговых продуктов (Global Trade Item Number (GTIN))	N2+N14
02	GTIN содержащихся в грузе торговых продуктов (GTIN of Contained Trade Items)	N2+N14
10	Номер партии/лота (Batch/Lot Number)	N2+X..20
11	Дата производства (Production Date (YYMMDD))	N2+N6
12	Дата, до которой необходимо оплатить счёт (Due Date (YYMMDD))	N2+N6
13	Дата упаковки (Packaging date (YYMMDD))	N2+N6
15	Дата, до которой лучше всего использовать товар (Best Before Date (YYMMDD))	N2+N6
37	Количество содержащихся торговых единиц (Number of Units Contained)	N2+N..8
240	Дополнительный идентификатор продукта (Additional Product Identification)	N3+X..30
8002	Идентификатор сотового телефона (Cellular mobile telephone identifier)	N4+X..20

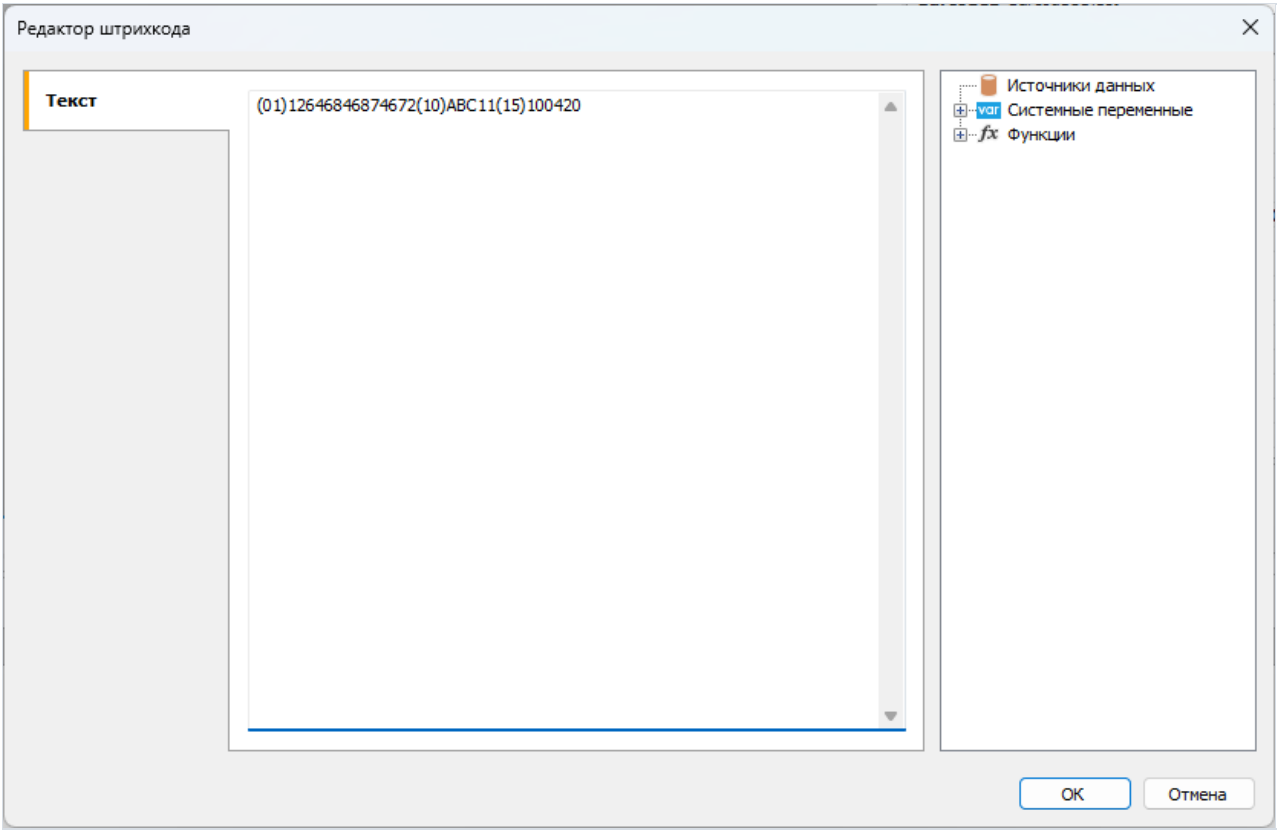
Если использовать приведенный пример кода при его формировании в FastReport .NET, то получим:

Для формирования штрихкода GS1-128 в FastReport .NET выберите объект Штрихкод (Barcode) на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "GS1", а затем GS1-128 (UCC/EAN-128):



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Interleaved 2 of 5

Штрихкод Interleaved 2 of 5 (ITF 2/5) – классический линейный код для кодирования числовой информации любой длины. Он был разработан Дэвидом Аллеем в 1972 году.

Особенностью этого штрихкода является обязательное условие – четное количество кодируемых символов. Это обусловлено тем, что кодирование символов происходит парами. Если код имеет нечетное количество цифр, впереди штрихкода добавляется ноль.

Первый символ из пары кодируется штрихами, а второй – пробелами. Штрихи и пробелы бывают узкими или широкими. Каждый символ кодируется пятью штрихами или пробелами. Два штриха широкие, а три – узкие. То же самое и относительно пробелов.

Благодаря такому способу кодирования штрихкод имеет высокую плотность и занимает меньше места, чем, например, Code 39.

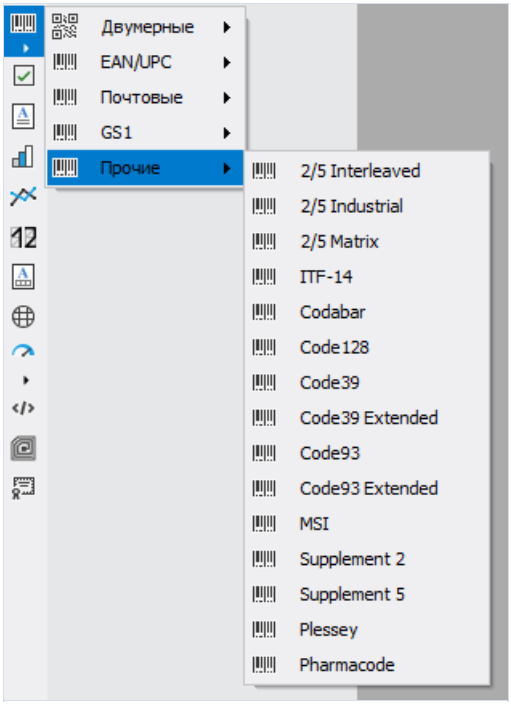
Штрихкод 2 of 5 Interleaved используется для маркировки продукции на складах, а также в судоходстве.

Структура кода:

- стартовый символ – указывает на начало кода;
- кодированные данные;
- необязательная контрольная цифра;
- стоп символ.

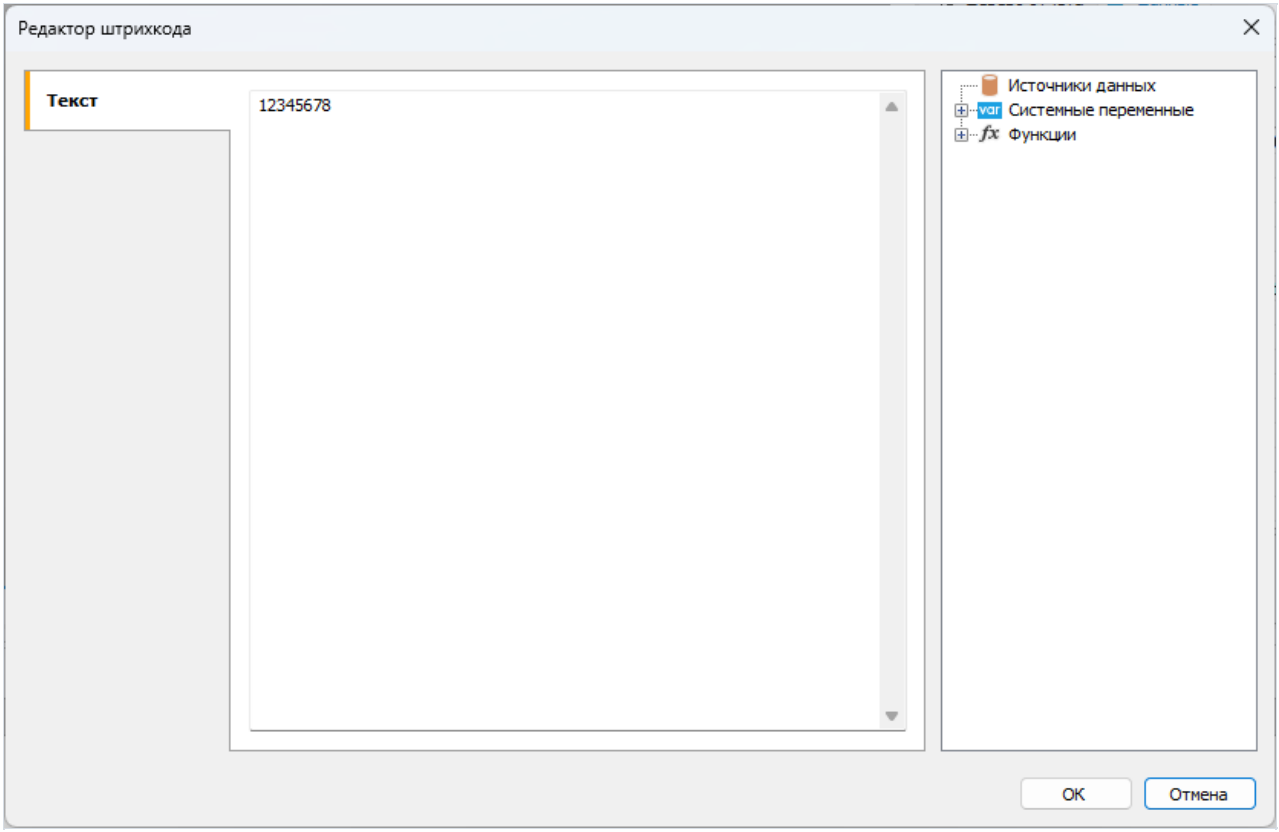
Штрихкод предусматривает самоконтроль, поэтому наличие контрольной цифры не обязательно.

Для формирования штрихкода Interleaved 2 of 5 в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Прочие", а затем 2/5 Interleaved:

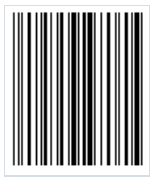


После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Если необходимо скрыть текст под штрихкодом, следует найти свойство `ShowText` в инспекторе свойств соответствующего штрихкода и установить его значение в `False`:



Industrial 2 of 5

Industrial 2 of 5 (также известен как Standard 2 of 5) – линейный штрихкод низкой плотности, предназначенный для кодирования чисел произвольной длины. Для кодирования используются только линии различной толщины, а пробелы служат только как разделители между ними. Из-за этого штрихкод имеет низкую плотность и, следовательно, большой размер.

Один символ кодируется пятью линиями, две из которых имеют большую толщину. Именно эта особенность и определяет название кода.

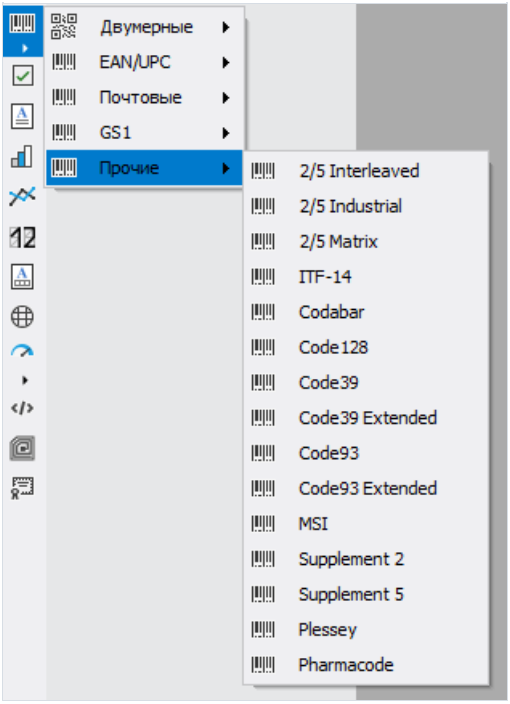
На данный момент этот штрихкод считается устаревшим. Он использовался в складском учете, в фотолабораториях и для нумерации авиабилетов.

Industrial 2 of 5 предусматривает контрольную цифру по модулю 10, но ее наличие не является обязательным.

Стандартный штрихкод 2 of 5 имеет следующую физическую структуру:

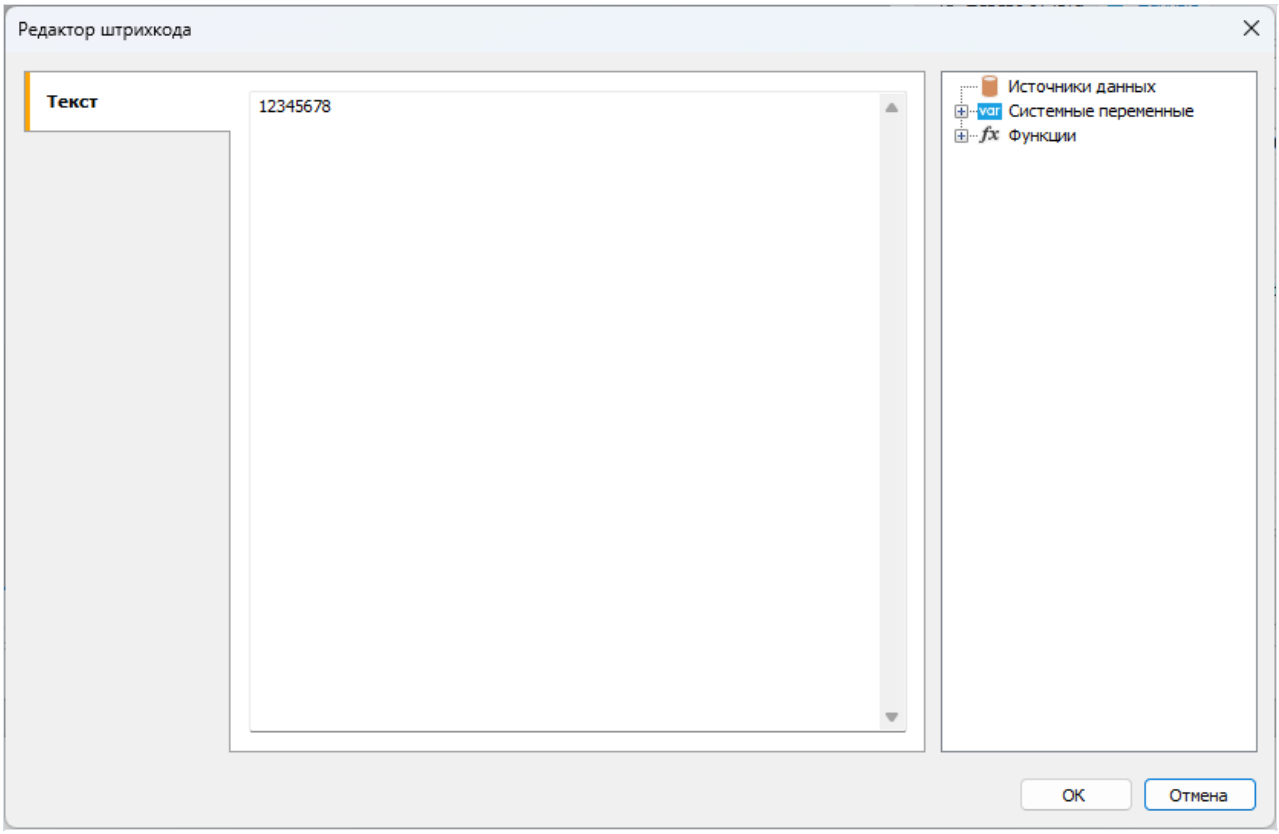
- стартовый символ;
- символы данных;
- необязательный контрольный символ;
- стоп-символ.

Для формирования штрихкода Industrial 2 of 5 в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Прочие", а затем 2/5 Industrial:

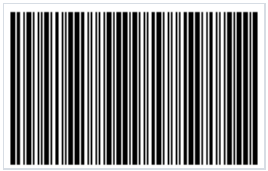


После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Если необходимо скрыть текст под штрихкодом, следует найти свойство `ShowText` в инспекторе свойств соответствующего штрихкода и установить его значение в `False`:



2 of 5 Matrix

2 of 5 Matrix - линейный штрихкод высокой плотности, предназначенный для кодирования числовых данных (0-9) произвольной длины. Он был разработан компанией Identicon в 1968 году. Этот вид кода обеспечивает высокую плотность благодаря использованию пробелов между линиями для кодирования информации, в результате чего в процессе кодирования участвуют как линии, так и пробелы.

Код обладает встроенной самопроверкой и поддерживает двунаправленное чтение, то есть его можно прочитать как слева направо, так и справа налево.

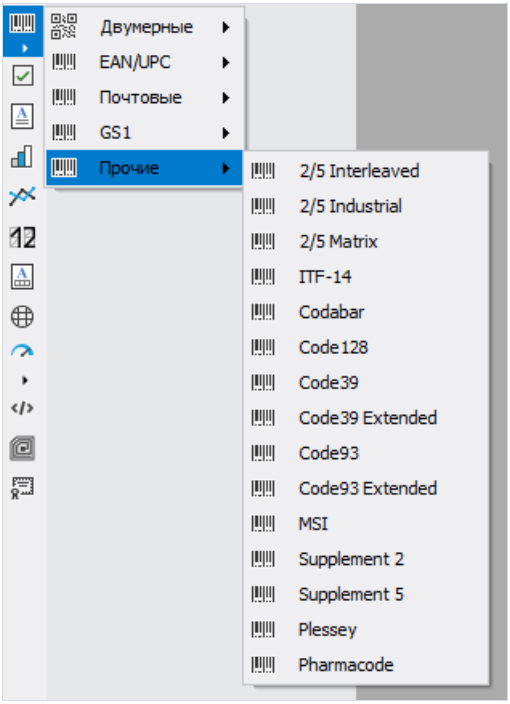
Структура кода:

- стартовый символ;
- данные;
- опциональный (необязательный) символ проверки. Рассчитывается по формуле `mod 10`;
- стоп-символ.

Символ состоит из пяти линий, две из которых широкие. Это символика с двумя ширинами, то есть в ней определена ширина тонкой линии и все, что шире – считается широкой линией. Это позволяет печатать коды с большим допуском на не очень качественном оборудовании.

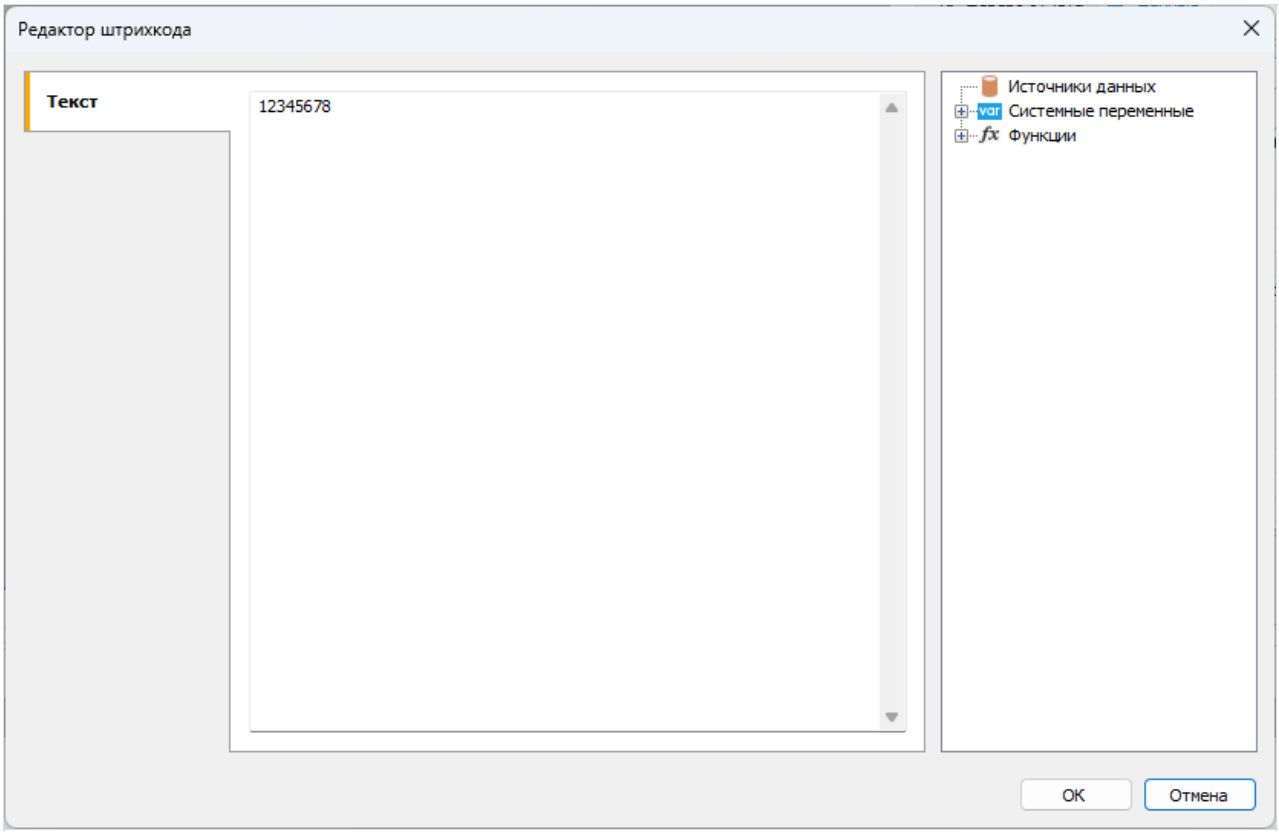
2 of 5 Matrix применяется для маркировки товаров на складах, маркировки авиабилетов, в фотолабораториях.

Для формирования штрихкода 2 of 5 Matrix в FastReport .NET выберите объект Штрихкод (Barcode) на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Прочие", а затем 2/5 Matrix:

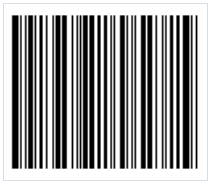


После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Если необходимо скрыть текст под штрихкодом, следует найти свойство `ShowText` в инспекторе свойств соответствующего штрихкода и установить его значение в `False`:



ITF-14

ITF-14 – двухполосный числовой код, также известный как код высокой плотности. Он способен кодировать только числа в чётном количестве. В этом типе штрихкода каждая нечётная цифра обозначается тёмной линией, а каждая чётная цифра – пробелом между ними. Для кодирования нечётного числа цифр необходимо дополнить самую левую (старшую) цифру нулем.

Штрихкод имеет следующую структуру:

- первый символ-индикатор обозначает уровень упаковки для конкретной картонной коробки. Этот однозначный префикс может варьироваться от 0 до 8. (например, 1 – коробка, 2 – ящик и т.д.);
- далее 2-3 цифры – региональный код (префикс) страны, где зарегистрирован данный номер;
- следующие 4-5 цифр представляют регистрационный номер предприятия внутри национальной организации;
- следующая группа цифр обозначает порядковый номер продукции внутри предприятия;
- последняя 13-я цифра является контрольной суммой или контрольным числом. Она вычисляется из предыдущих двенадцати по алгоритму Module 10.

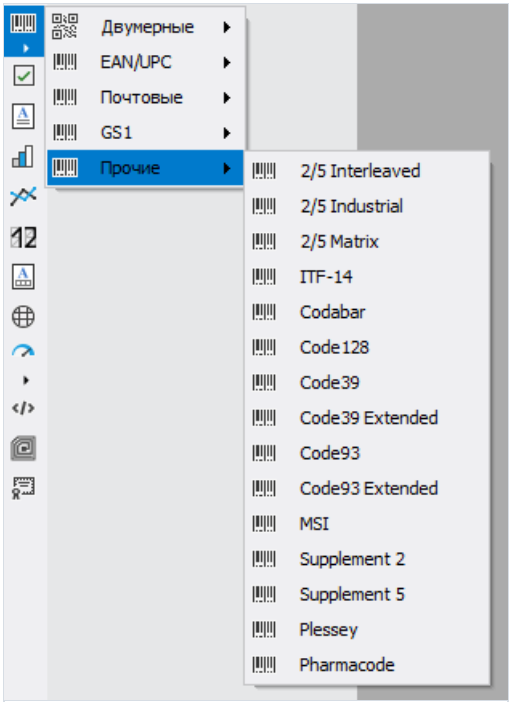
Штрихкод ITF-14 применяется для автоматизации складского учета товаров, помещенных в индивидуальную или групповую транспортную упаковку. По нему компьютерная система учета определяет не только вид товара, находящийся в упаковке, но и его количество.

ITF-14 обычно используются для печати на гофрированном картоне, для маркировки картонных коробок, ящиков или поддонов. Эти штрихкоды широко применяются розничными торговцами, производителями и дистрибьюторами для точного контроля логистики и управления запасами. Кроме того, они могут использоваться для идентификации багажа в аэропортах, нумерации авиабилетов и идентификации почтовых отправлений.

Поскольку штрихкод ITF-14 предназначен для обозначения товаров в транспортных упаковках – он не предусматривает обработку на кассовых терминалах.

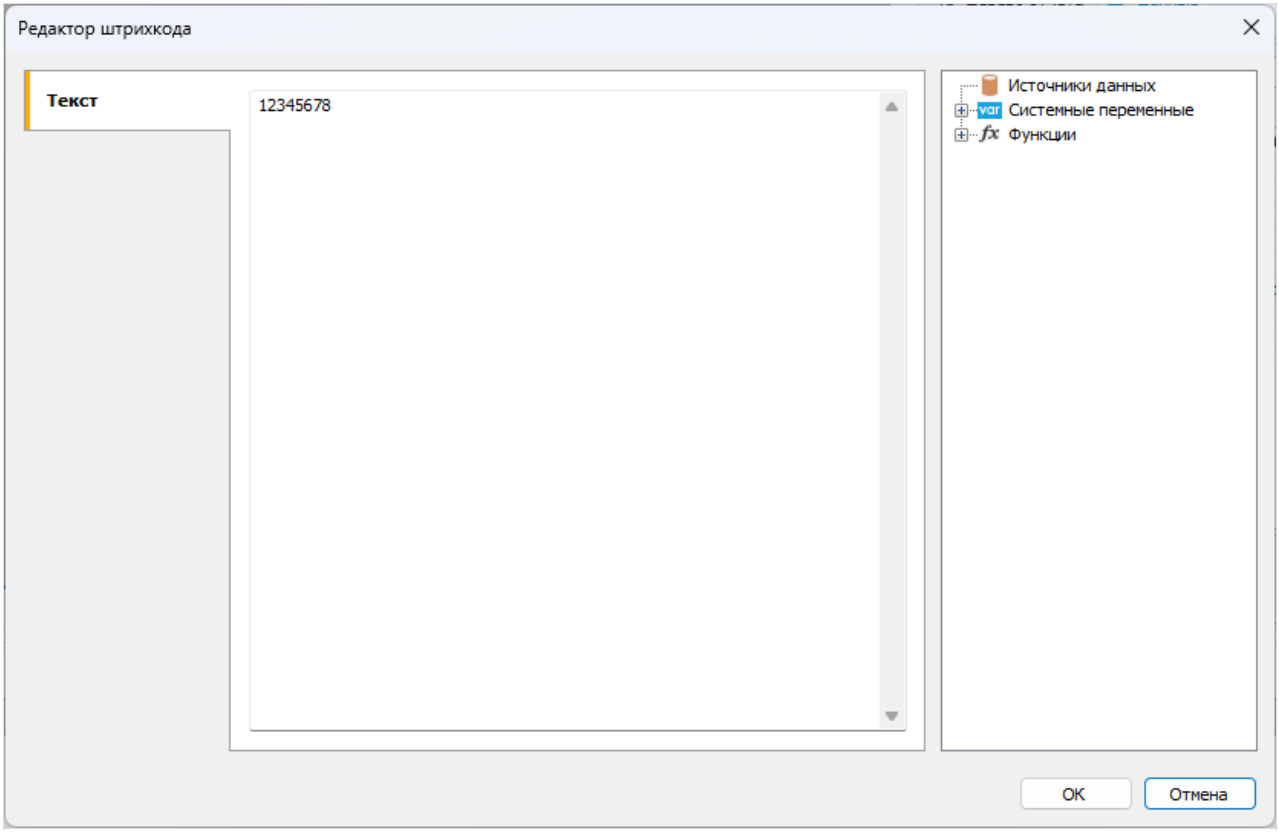
Толстая черная рамка вокруг штрихкода называется опорной полосой (Bearer Bar). Эта полоса уравнивает давление, создаваемое печатающей пластиной по всей поверхности штрихкода и улучшает читаемость, сокращая вероятность неполного сканирования символа. ITF-14 может быть с видимыми или скрытыми вертикальными опорными полосами.

Для формирования штрихкода ITF-14 в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Прочие", а затем ITF-14:



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Рассмотрим свойства объекта Штрихкод (Barcode):

Свойство	Описание
Поворот (Angle)	Позволяет задать поворот объекта на один из фиксированных углов – 0, 90, 180, 270 градусов.

Свойство	Описание
Масштаб (Zoom)	Задает масштабирование штрихкода. Это свойство используется только вместе со свойством "Авторазмер".
Авторазмер (AutoSize)	Если это свойство включено, объект будет растягиваться, чтобы показать штрихкод целиком. Если свойство отключено, штрихкод будет растянут до размеров объекта.
Показывать текст (ShowText)	Определяет, надо ли показывать ли текст в нижней части штрихкода.
Поле данных (DataColumn)	Поле данных, из которого загружать текст объекта.
Выражение (Expression)	Выражение, которое возвращает текст объекта.
Текст (Text)	Текст объекта.
Отступы (Padding)	Позволяет задать отступы от краев объекта, в пикселях.
Ширина полос (WideBarRatio)	Это свойство имеется у всех линейных штрихкодов. Оно определяет относительный размер широких полос штрихкода.
Контрольная сумма (CalcCheckSum)	Это свойство имеется у многих линейных штрихкодов. Оно определяет, надо ли считать контрольную сумму автоматически. Если это свойство отключено, контрольная сумма должна присутствовать в тексте объекта.
Отображение вертикальных полос (DrawVerticalBearerBars)	Если это свойство включено, то у объекта будут отображаться боковые линии.

Если свойство `DrawVerticalBearerBars` отключено, штрихкод будет иметь следующий вид:

Codabar

Codabar – линейный штрихкод переменной длины, который чаще всего применяется для кодирования серийных номеров. Он создан еще в 1972 году компанией Pitney Bowes. Его штрихи обеспечивают точное считывание кода даже, если код напечатан на матричном принтере.

Симвология:

- цифры 0-9;
- буквы A, B, C, D;
- специальные символы `-`, `$`, `/`, `.`, `+`.

Буквы можно использовать только как стартовый и финишный символ кода. Их можно использовать как дополнительную кодируемую информацию.

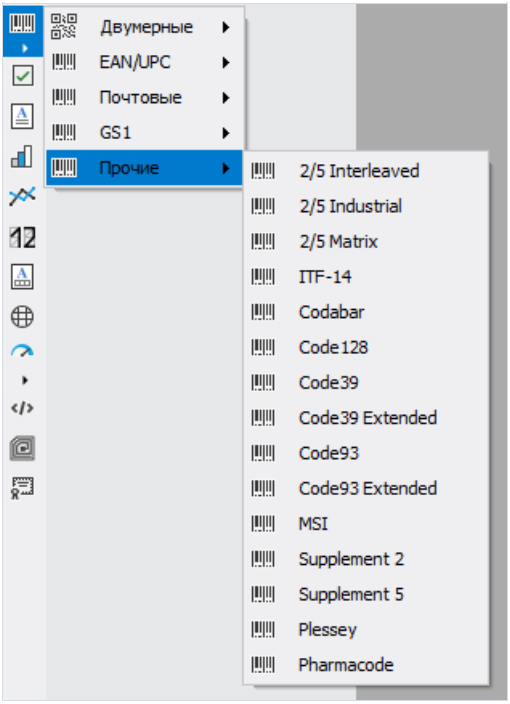
Структура кода крайне проста:

- старт-символ обозначает начало чтения данных;
- кодируемые данные;
- стоп-символ – обозначает окончание кода.

Символ контрольной суммы не предусмотрен, так как код имеет встроенную самопроверку.

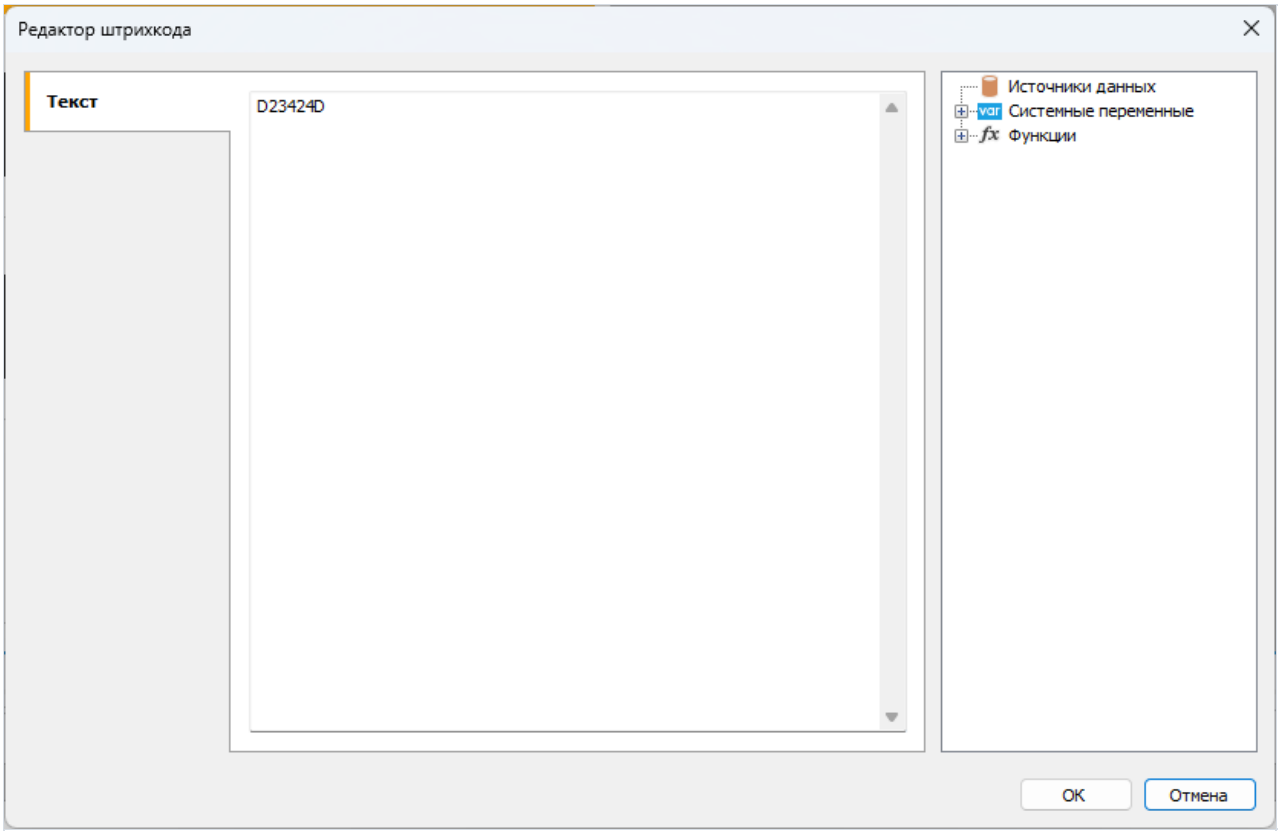
Каждый символ в Codabar кодируется 4 линиями и 3 пробелами.

Для формирования штрихкода Codabar в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Прочие", а затем Codabar:



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Если необходимо скрыть текст под штрихкодом, следует найти свойство `ShowText` в инспекторе свойств соответствующего штрихкода и установить его значение в `False`:

Code 128

Линейный штрихкод Code 128 получил широкое распространение во многих сферах. Из его названия можно понять, что он позволяет закодировать 128 символов. Однако, это не ограничивается только цифрами, так как он также способен кодировать буквы и некоторые специальные символы. Именно это является его главным достоинством и отличием от кодов стандарта EAN.

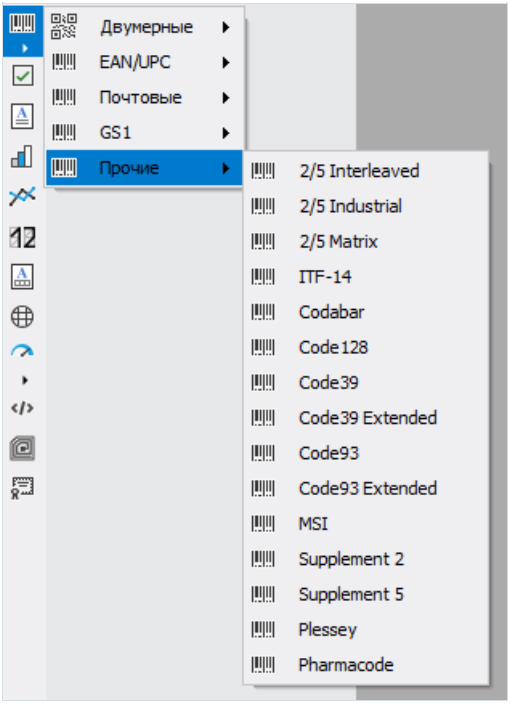
Структура кода, как и у большинства линейных штрихкодов проста. С обеих сторон, код имеет пустое пространство, которое позволяет четко определить начало кода. Далее, начало кода обозначается специальным символом – последовательностью линий и пробелов. После стартового символа следуют кодируемые данные. Затем – символ контрольной суммы для проверки целостности кода. Завершает – код остановки сигнализирующий окончание кодированных данных.

Кодирование разной информации осуществляется с помощью разных наборов символов для букв, цифр и специальных символов. Каждый символ кодируется тремя линиями и тремя пробелами. Линии и пробелы могут иметь разную ширину. Ширина определяется количеством модулей – от 1 до 4.

В результате Code 128 обладает следующими преимуществами:

- кодировать можно как числа, так и буквы (заглавные);
- чисто цифровой код очень компактный благодаря применению двойной упаковки.

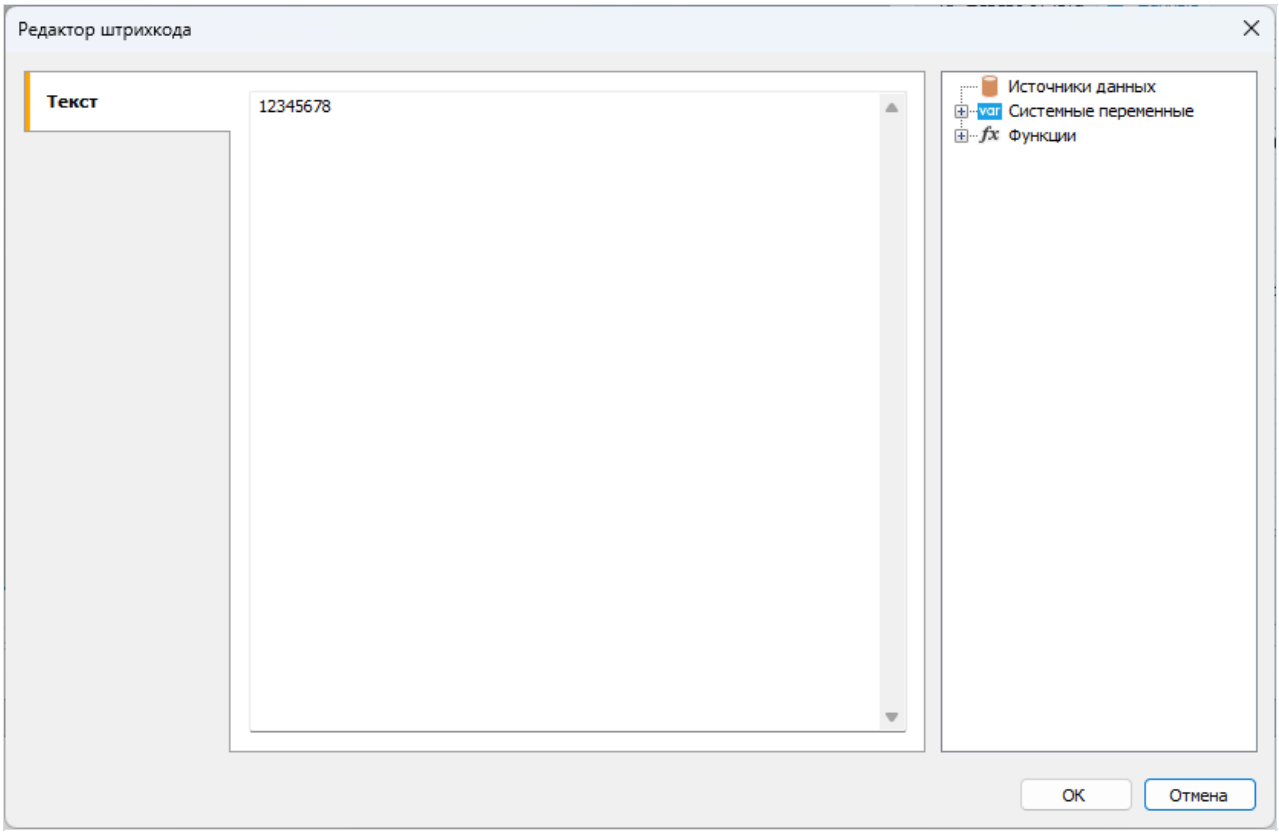
Для формирования штрихкода Code 128 в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Прочие", а затем Code128:



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно

открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Если необходимо скрыть текст под штрихкодом, следует найти свойство `ShowText` в инспекторе свойств соответствующего штрихкода и установить его значение в `False`:

Code 39

Code 39 – линейный штрихкод, разработанный в 1975 году компанией Intermec. Этот код получил большую популярность благодаря возможности кодировать буквенно-цифровые данные. Активно используется до сих пор.

Важной особенностью этого кода является возможность кодировать данные произвольной длины. Ширина кода ограничивается возможностью чтения сканера.

Каждый символ в коде представлен 9 элементами: 5 линиями и 4 пробелами. Такой большой набор элементов обусловлен встроенной самопроверкой. Это означает, что каждый символ в коде проверяется, и нет необходимости в контрольной цифре. Однако её добавление все же допускается для большего обеспечения целостности штрихкода.

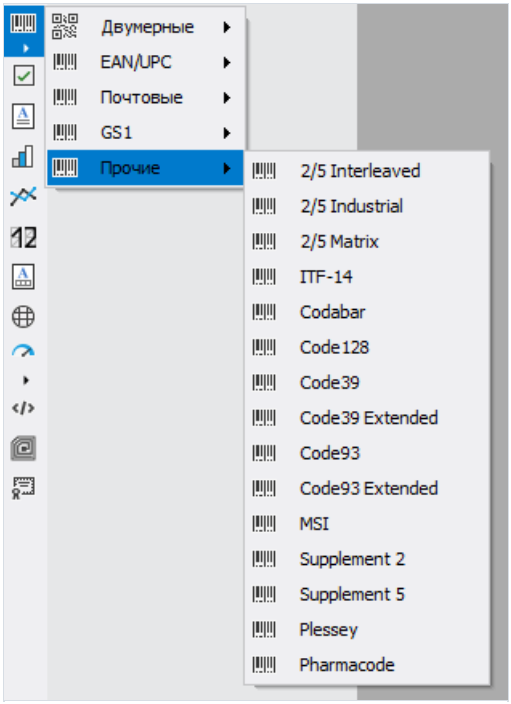
Как уже отмечалось выше, этот штрихкод позволяет кодировать буквенно-цифровую информацию. Допускаются:

- латинские буквы в верхнем регистре (A-Z);
- цифры (0-9);
- специальные символы (`-` , `,` , `.` , `$` , `/` , `+` , `%` , `:`).

Штрихкод Code 39 имеет следующую структуру:

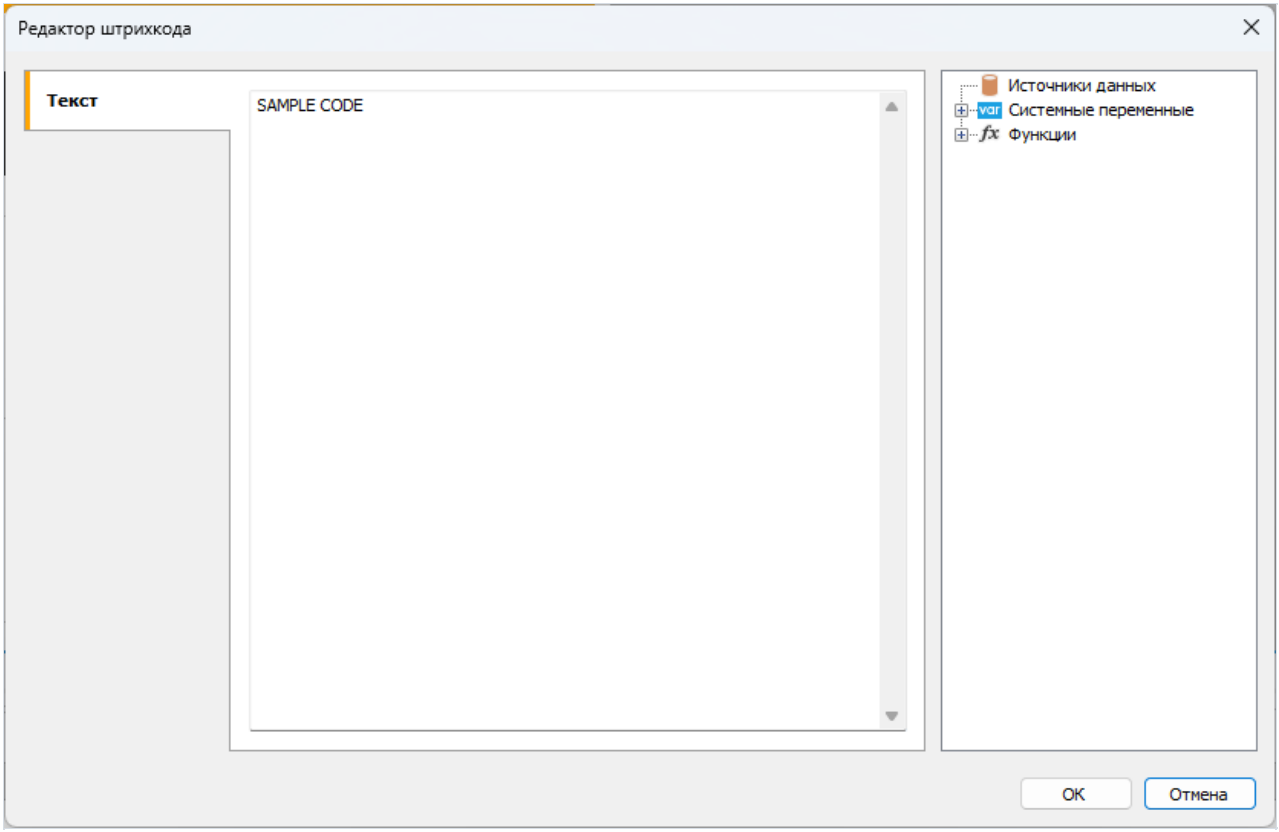
- старт-символ кодируемый `*` . Обозначает начало кода;
- закодированные данные;
- необязательная цифра контрольной суммы;
- стоп-символ, также кодируется `*` .

Для формирования штрихкода Code 39 в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Прочие", а затем Code39:



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Если необходимо скрыть текст под штрихкодом, следует найти свойство `ShowText` в инспекторе свойств соответствующего штрихкода и установить его значение в `False`:

Code 39 Extended

Из названия становится ясно, что это расширенная версия Code 39. Она поддерживает полный набор символов ASCII, включая заглавные буквы A-Z, строчные a-z и большой набор спецсимволов. Как и обычный Code 39, расширенная версия может содержать контрольный символ, который рассчитывается по формуле модуля 43.

Штрихкод позволяет кодировать буквенно-цифровую информацию. Допускаются:

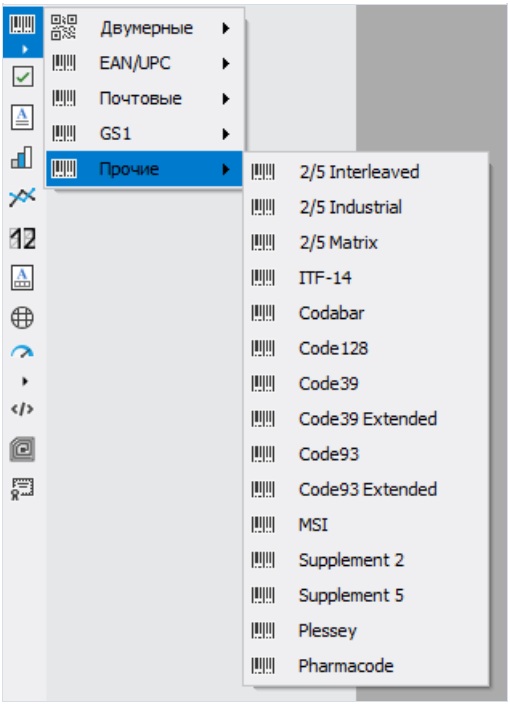
- латинские буквы в верхнем (A-Z) и в нижнем регистре;
- цифры (0-9);
- специальные символы.

Code 39 Extended имеет следующую структуру:

- старт-символ кодируемый символом `*`;
- закодированные данные;
- необязательная цифра контрольной суммы;
- стоп-символ, также кодируется символом `*`.

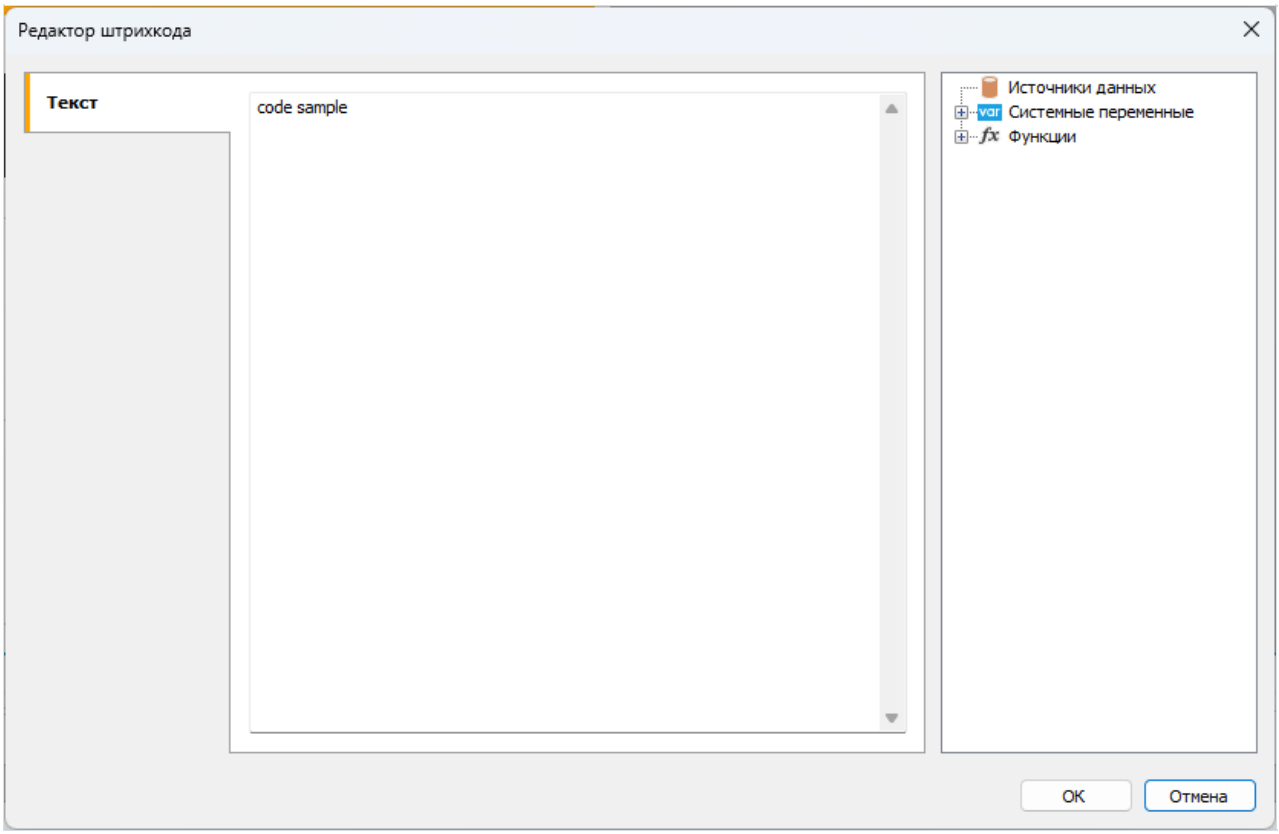
Для создания строчной буквы используется два символа: `+` и соответствующая заглавная буква. Для кодирования некоторых спецсимволов также используется сочетание двух символов: `%` и заглавная буква, или `/` и заглавная буква.

Для формирования штрихкода Code 39 Extended в FastReport .NET выберите объект Штрихкод (Barcode) на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Прочие", а затем Code39 Extended:

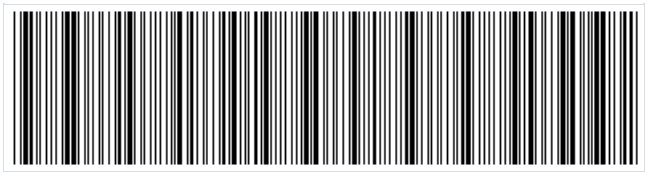


После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Если необходимо скрыть текст под штрихкодом, следует найти свойство `ShowText` в инспекторе свойств соответствующего штрихкода и установить его значение в `False`:



Code 93

Code 93 (спецификация AIM-BC5-2000) – линейный штрихкод, который позволяет кодировать как цифры, так и буквы. Разработан в 1982 году компанией Intermec и создан на основе Code 39. Главным преимуществом по сравнению с предшественником является более компактный размер.

Code 93 применяется почтой Канады для кодирования информации об отправлениях, а также в автомобильной промышленности.

Длина кода ограничивается 48 символами, из которых: 26 букв в верхнем регистре, 10 цифр, 7 специальных символов и 5 служебных символов, таких как начало и конец кода, а также символы для расширения символики. Этот код может представлять полный набор символов ASCII с использованием сочетания с двумя служебными символами.

Структура кода:

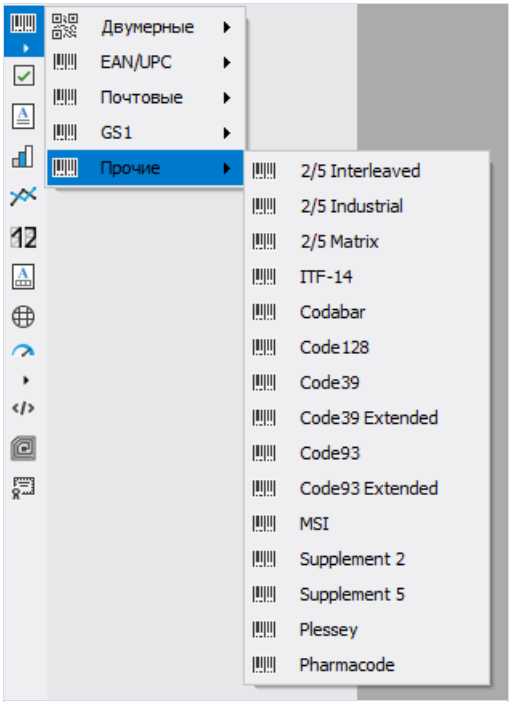
- стартовый символ;
- кодируемая информация;
- цифра контрольной суммы, рассчитываемая по модулю "C";
- цифра контрольной суммы, рассчитываемая по модулю "K";
- стоп-символ.

Стартовый символ представляется как `*`.

Каждый символ данных состоит из трех линий и трех пробелов. Ширина линии или пробела может быть от 1 до 4 модулей.

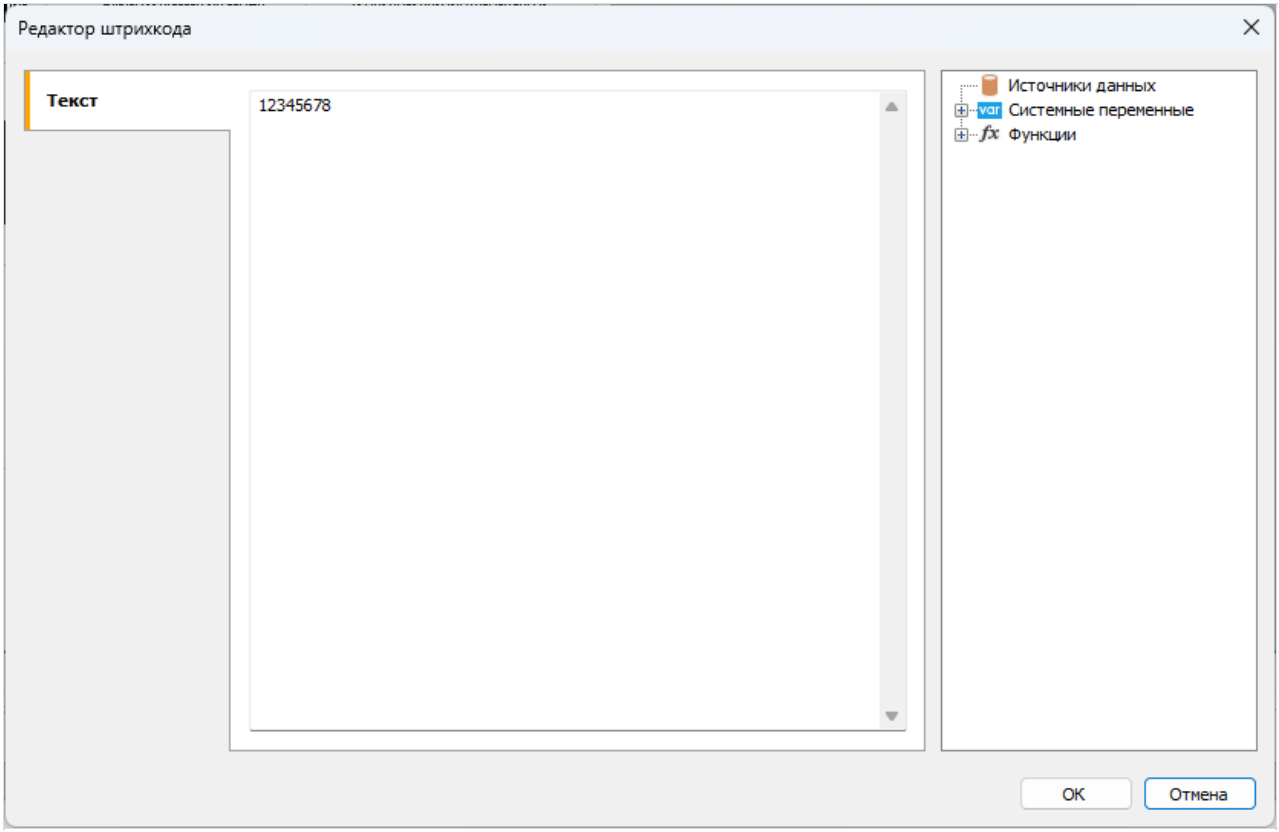
Цифры контрольной суммы не отображаются в тексте под кодом. Контрольная сумма состоит из двух цифр, которые рассчитываются разными способами: по модулю "C" и по модулю "K". Формула расчета представляет собой остаток по модулю 47 от суммы весов значений данных. Отличие между "C" и "K" заключается в методах взвешивания значений.

Для формирования штрихкода Code 93 в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Прочие", а затем Code93:

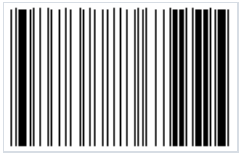


После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Если необходимо скрыть текст под штрихкодом, следует найти свойство `ShowText` в инспекторе свойств соответствующего штрихкода и установить его значение в `False`:



Code 93 Extended

Code 93 Extended – расширенная версия штрихкода Code 93. В отличие от базовой версии, он способен кодировать 128 ASCII символов, вместо 48.

Набор символов:

- цифры 0-9;
- прописные буквы A-Z;
- специальные символы: `$`, `%`, `/`, `+`.

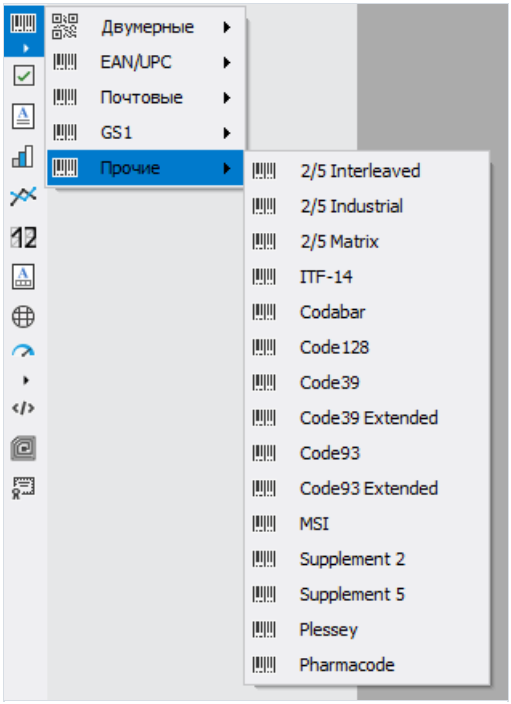
Структура кода:

- стартовый символ `*`;
- кодируемая информация – символ данных состоит из трех линий и трех пробелов. Ширина линии или пробела может быть от 1 до 4 модулей;
- цифра контрольной суммы, рассчитываемая по модулю "C";
- цифра контрольной суммы, рассчитываемая по модулю "K";
- стоп-символ `*`.

Цифры контрольной суммы не отображаются в тексте под кодом. Контрольная сумма состоит из двух цифр, которые рассчитываются разными способами: по модулю "C" и по модулю "K". Формула расчета представляет собой остаток по модулю 47 от суммы весов значений данных. Отличие между "C" и "K" заключается в методах взвешивания значений.

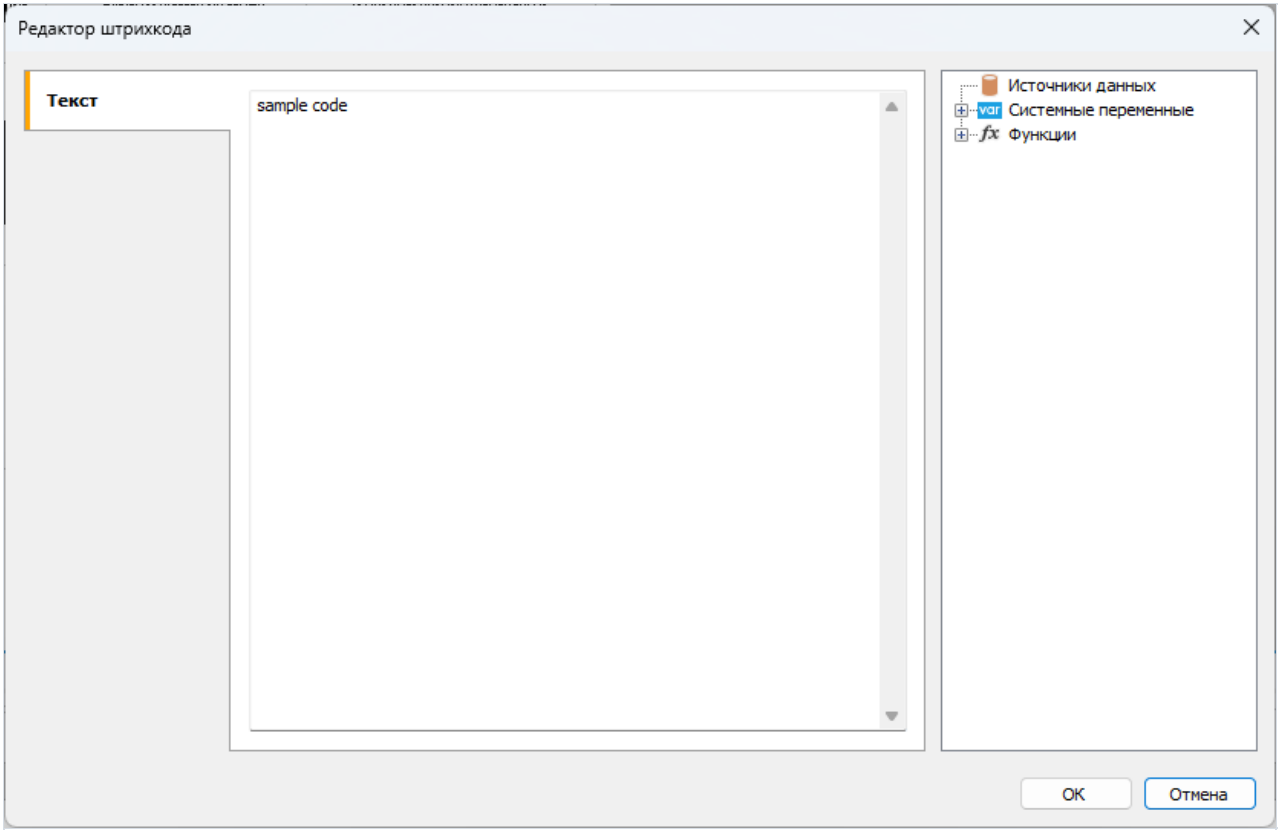
Как уже отмечалось, Code 93 Extended позволяет закодировать 128 символов. Однако, для расширения стандартной таблицы символов Code 93 требуется использовать дополнительные символы (`$`, `/`, `%`, `+`). Таким образом, для кодирования строчных букв необходимо использовать сочетание символа `+` и соответствующей прописной буквы. Остальные символы-модификаторы нужны для расширения набора специальных символов.

Для формирования штрихкода Code 93 Extended в FastReport .NET выберите объект Штрихкод (Barcode) на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Прочие", а затем Code93 Extended:



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Если необходимо скрыть текст под штрихкодом, следует найти свойство `ShowText` в инспекторе свойств соответствующего штрихкода и установить его значение в `False`:

MSI

Линейный штрихкод MSI был создан в 1971 году на основе штрихкода Plessey и является по сути его усовершенствованной версией. Он позволяет кодировать только числа от 0 до 9.

Структура кода:

- стартовый символ, означающий начало чтения данных (110);
- данные;
- опциональный контрольный символ;
- стоп-символ, означающий окончание чтения данных (1001).

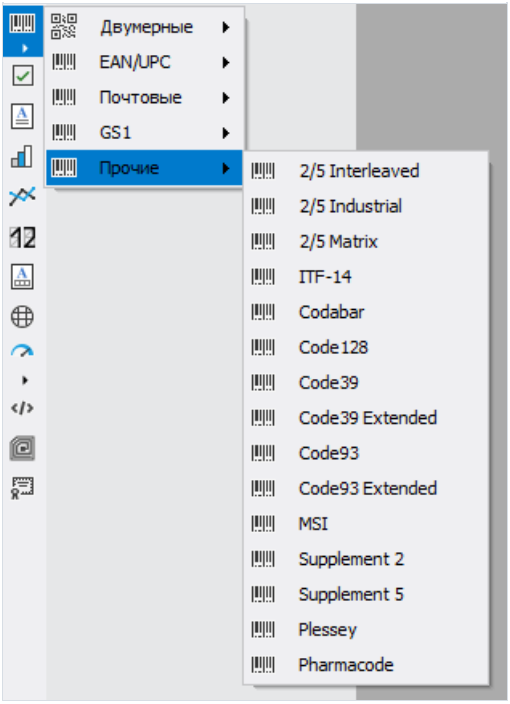
Каждый символ в коде представлен в двоичном виде с помощью штрихов и пробелов. Штрих – 1, а пробел – 0.

Контрольный символ может быть рассчитан по одному из четырех типов: Modulo 10, Modulo 11, Modulo 1010, Modulo 1110. Самый распространенный тип – Modulo 10. По определенному алгоритму рассчитывается контрольный символ. Прочитанные сканером данные складываются по алгоритму и полученный результат сравнивается с контрольным символом. При положительном результате, код считается прочитанным правильно.

MSI – код произвольной длины. Она ограничивается лишь возможностями сканера.

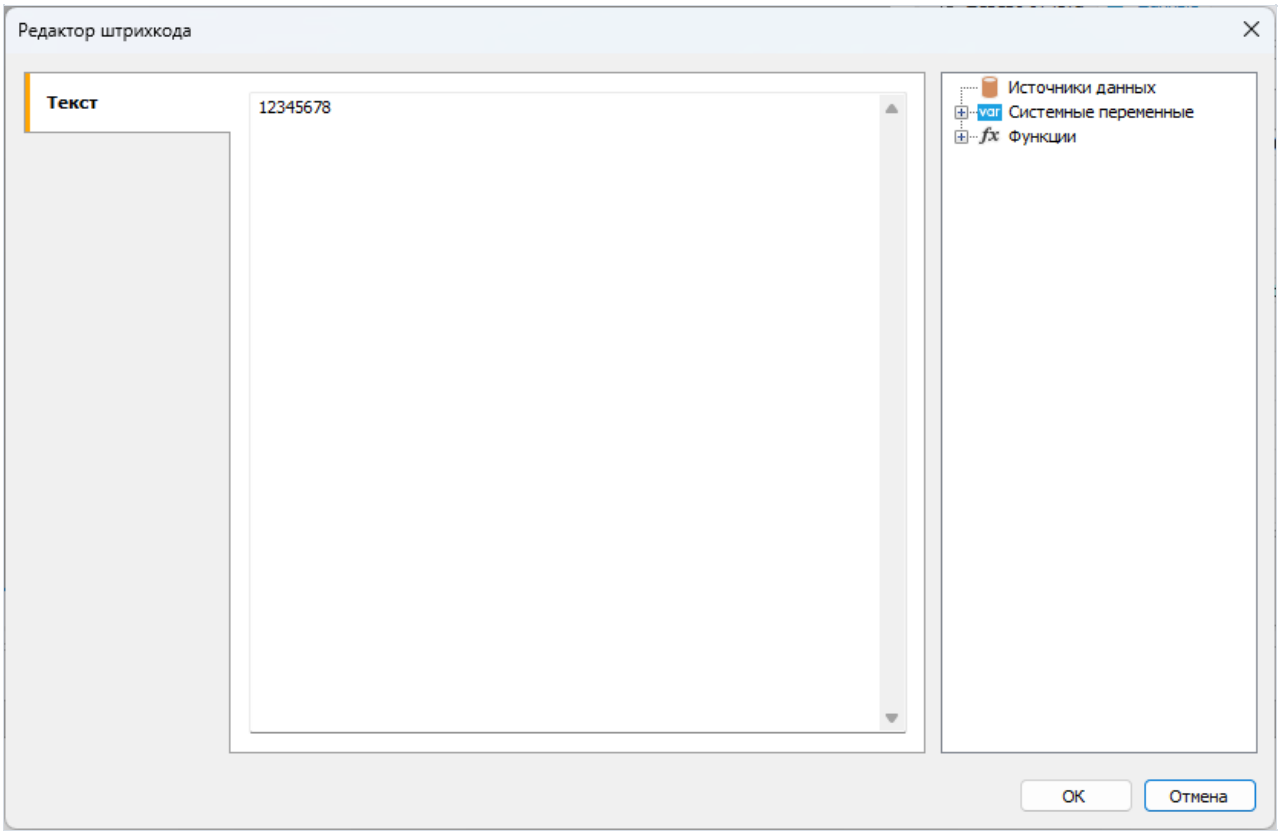
Сейчас этот штрихкод считается устаревшим и практически не используется. Ранее он применялся для маркировки товаров на складах и в супермаркетах.

Для формирования штрихкода MSI в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Прочие", а затем MSI:

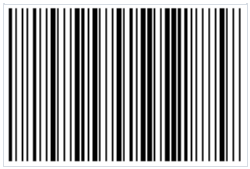


После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Если необходимо скрыть текст под штрихкодом, следует найти свойство `ShowText` в инспекторе свойств соответствующего штрихкода и установить его значение в `False`:



Supplement

Распространенные одномерные штрихкоды, такие как EAN в Европе и UPC в США, могут быть расширены с помощью дополнительного кода, размещаемого справа от основного кода.

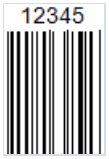
UPC-A, UPC-E, EAN-13 и EAN-8 могут включать дополнительный штрихкод справа от основного штрихкода. Этот второй штрихкод, который обычно не такой высокий, как основной штрихкод, используется для кодирования дополнительной информации для газет, книг и других периодических изданий.

Дополнительный штрихкод может закодировать 2 или 5 цифр информации.

Двухзначный дополнительный код обычно применяется для кодирования информации о номере выпуска в периодических изданиях (журналы, газеты).

Дополнительный код позволяет не включать номер выпуска в основной штрихкод, оставляя его неизменным для всех выпусков. Так, нет необходимости считывать большой код, когда нужна лишь информация о выпуске.

Пятизначный дополнительный код часто применяется на книгах и содержит информацию о рекомендуемой розничной цене экземпляра. Из пяти символов первый обозначает код валюты, остальные 4 – цену.



Существуют определенные коды для обозначения без цены:

- код 90000 – для книги розничная цена не определена;
- код 99991 – книга распространяется бесплатно.

Но дополнительный код может хранить и другую информацию, для внутреннего применения в издательстве.

Каждая цифра кодируется 7 модулями (линиями и пробелами).

Структура кода:

- стартовый символ (1011 если провести аналогию между штрихами и цифрами);
- первый символ данных;
- разделитель (01);
- второй символ данных либо 4 символа в случае 5-символьного кода.

Явного контрольного символа и стоп-символа не предусмотрено. После разделителя считывается допустимое количество символов.

Кодирование данных проводится по наборам "левый четный" и "левый нечетный", применяемым в EAN.

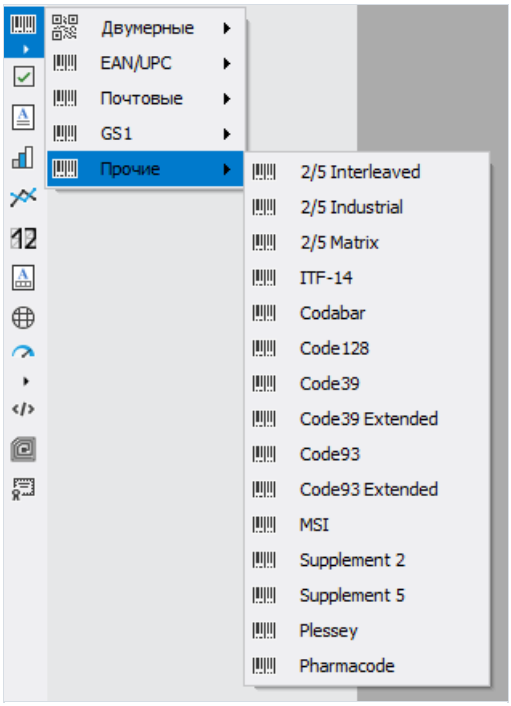
В зависимости от размерности дополнительного штрихкода применяются разные шаблоны четности символов. Четность символа определяет контрольную сумму.

Если дополнительный штрихкод двузначный, то двузначное число, которое образуется в результате сложения первого и второго числа, необходимо разделить на 4. Далее, если остаток от деления четное число, то первый символ кодируется с четностью, а второй с нечетностью. Это означает, что для кодирования первой цифры будет использован четный набор значений, а второго – нечетный.

Теперь, при сканировании будет определяться четность прочитанного значения. Если она не соответствует расчетной четности, которая ожидается – значит штрихкод прочитан неправильно.

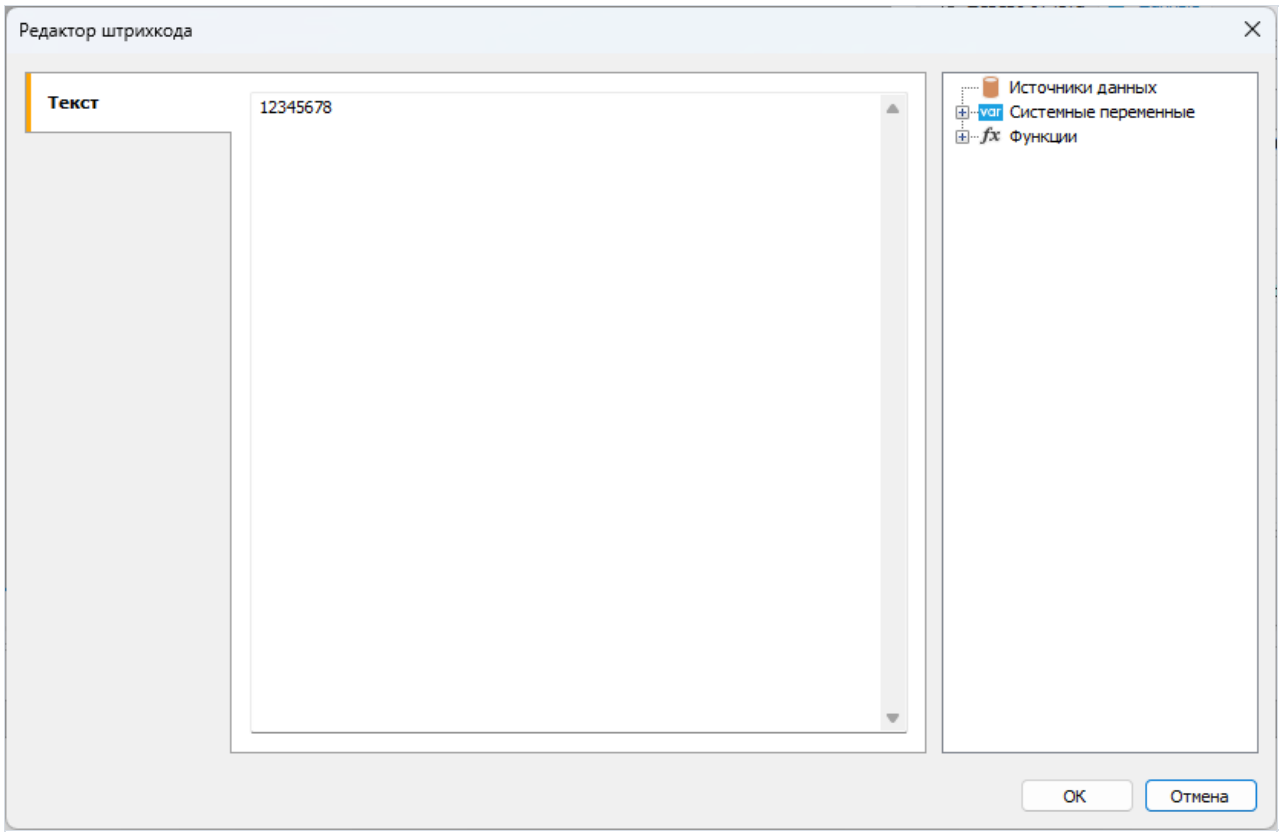
В случае с пятизначным кодом расчет контрольной суммы сложнее. Считается, что последняя цифра кода находится в нечетной позиции. Начиная с последней цифры и до первой по очереди назначаются четные и нечетные позиции. Затем, берется сумма всех нечетных цифр и умножается на 3. Сумма всех четных цифр умножается на 9. Далее берется единица измерения от суммы двух предыдущих вычислений, то есть крайняя правая цифра. Это и есть контрольная цифра по которой определяется шаблон четности в специальной таблице.

Для формирования штрихкода Supplement 2 (для двухзначного кода) или Supplement 5 (для пятизначного кода) в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Прочие", а затем Supplement 2 или Supplement 5:



После выбора штрихкода разместите его на странице отчёта. Позиционируйте дополнительный код рядом с основным:

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:



Как и все штрихкоды в FastReport .NET Supplement имеет ряд свойств, которые вы можете отредактировать в инспекторе свойств объекта:

Свойство	Описание
Поворот (Angle)	Позволяет задать поворот объекта на один из фиксированных углов – 0, 90, 180, 270 градусов.
Масштаб (Zoom)	Задает масштабирование штрихкода. Это свойство используется только вместе со свойством "Авторазмер".
Авторазмер (AutoSize)	Если это свойство включено, объект будет растягиваться, чтобы показать штрихкод целиком. Если свойство отключено, штрихкод будет растянут до размеров объекта.
Показывать текст (ShowText)	Определяет, надо ли показывать ли текст в нижней части штрихкода.
Поле данных (DataColumn)	Поле данных, из которого загружать текст объекта.
Выражение (Expression)	Выражение, которое возвращает текст объекта.
Текст (Text)	Текст объекта.
Отступы (Padding)	Позволяет задать отступы от краев объекта, в пикселях.
Ширина полос (WideBarRatio)	Это свойство имеется у всех линейных штрихкодов. Оно определяет относительный размер широких полос штрихкода.
Контрольная сумма (CalcCheckSum)	Это свойство имеется у многих линейных штрихкодов. Оно определяет, надо ли считать контрольную сумму автоматически. Если это свойство отключено, контрольная сумма должна присутствовать в тексте объекта.

Свойство	Описание
Отображение вертикальных полос (DrawVerticalBearerBars)	Если это свойство включено, то у объекта будут отображаться боковые линии.

Plessey

Штрихкод Plessey был разработан в 1971 году одноименной компанией. Это классический одномерный линейный штрихкод, который применялся в основном для маркировки товаров на полках магазинов и при складском учете. Основное преимущество данного кода на момент создания – простота печати на матричном принтере. В настоящее время он считается устаревшим и практически не встречается.

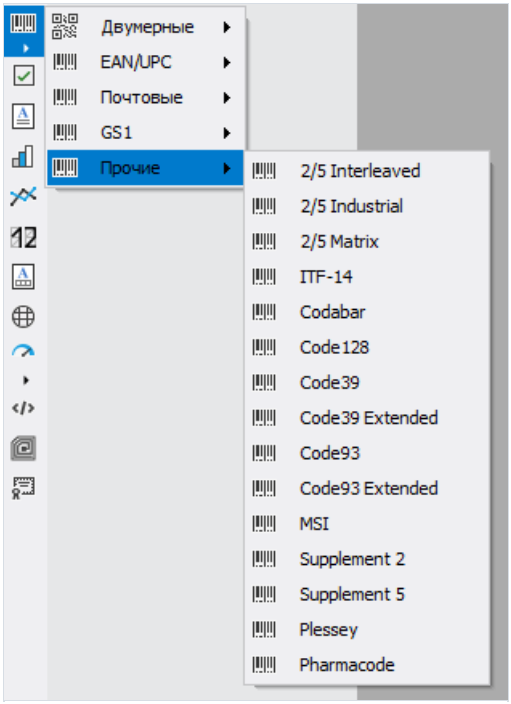
Plessey позволяет кодировать шестнадцатеричные цифры (0-F). Каждая цифра представлена четырьмя битами – полосами. Ноль – тонкая линия, один – толстая. Помимо цифр, могут быть закодированы и буквы от A до F. Структура штрихкода включает в себя стартовый код, кодированные данные, код контрольной суммы, финишную метку и код для реверсного чтения, что позволяет читать код в любом направлении.

Одной из разновидностей Plessey стал MSI. В отличие от обычного Plessey, он позволяет кодировать только цифры, а также отсутствует код для реверсного чтения. MSI поддерживает несколько разновидностей кода контрольной суммы, например: Mod-10, Mod-11, Mod-1010, Mod-1110.

И MSI, и стандартный Plessey не ограничивают длину кода, однако слишком длинный код может просто не поместиться на упаковке, и сканер может быть не рассчитан на такую большую длину.

Вот как будет выглядеть такой длинный код:

Для формирования штрихкода Plessey в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Прочие", а затем Plessey:



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши.

Pharmacode

Pharmacode – двоичный код, который был разработан немецкой компанией LAETUS GMBH специально для фармацевтических упаковок. Он является подмножеством CODE39. Этот код широко применяется в фармацевтической промышленности, как часть системы контроля упаковки продукции.

В рамках автоматизированной системы упаковки, Pharmacode позволяет легко сканировать и регистрировать фармацевтические поставки с использованием универсальных идентификаторов. Также, с помощью сканеров легко определить была ли партия препаратов перемешана с другой.

Штрихкод Pharmacode гарантирует прочтение кода, несмотря на возможные ошибки печати. Также, для гарантии, что остальная часть упаковки, кроме кода, напечатана правильно, Pharmacode могут быть напечатаны в разных цветах (код и фон), в отличие от штрихкодов, предназначенных для чтения с помощью лазера или эмуляции лазера. Это возможно потому, что Pharmacode сканируется специальными сканерами белого цвета LAETUS. Это делает Pharmacode очень практичным форматом для печати на упаковке или документах, не содержащих черных чернил.

Как выше было отмечено, Pharmacode может быть напечатан разными цветами. Причём и сам код, и цвет фона могут быть отличными от белого и чёрного. Существует специальные спецификации сочетаний цвета кода и фона, применяемые в зависимости от типа сканера для чтения. Например, стандартные черно-белые сканеры воспринимают только контрастные код и фон, а специальные сканеры, распознающие цвет, не имеют жёстких ограничений.

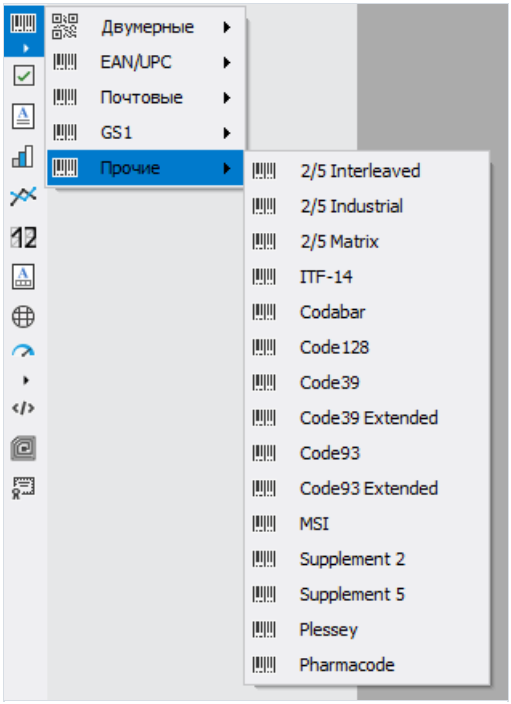
В отличие от других 1D штрихкодов, Pharmacode хранит данные в двоичной системе, а не десятичной. Кроме того, Pharmacode может представлять только единичные целые числа от 3 до 131070. Минимальное количество линий равно 2 для числа 3, а максимальное значение равно 16 для 131070. Уникальность Pharmacode заключается также в том, что он читается справа налево, в отличие от других линейных штрихкодов, которые имеют старт- и стоп-символ. Если читать код слева направо, то получится совсем другая последовательность чисел.

Стандарт Pharmacode регламентируется разработчиком LAETUS и описан в документе PharmaCode Guide.

Вот пример Pharmacode:

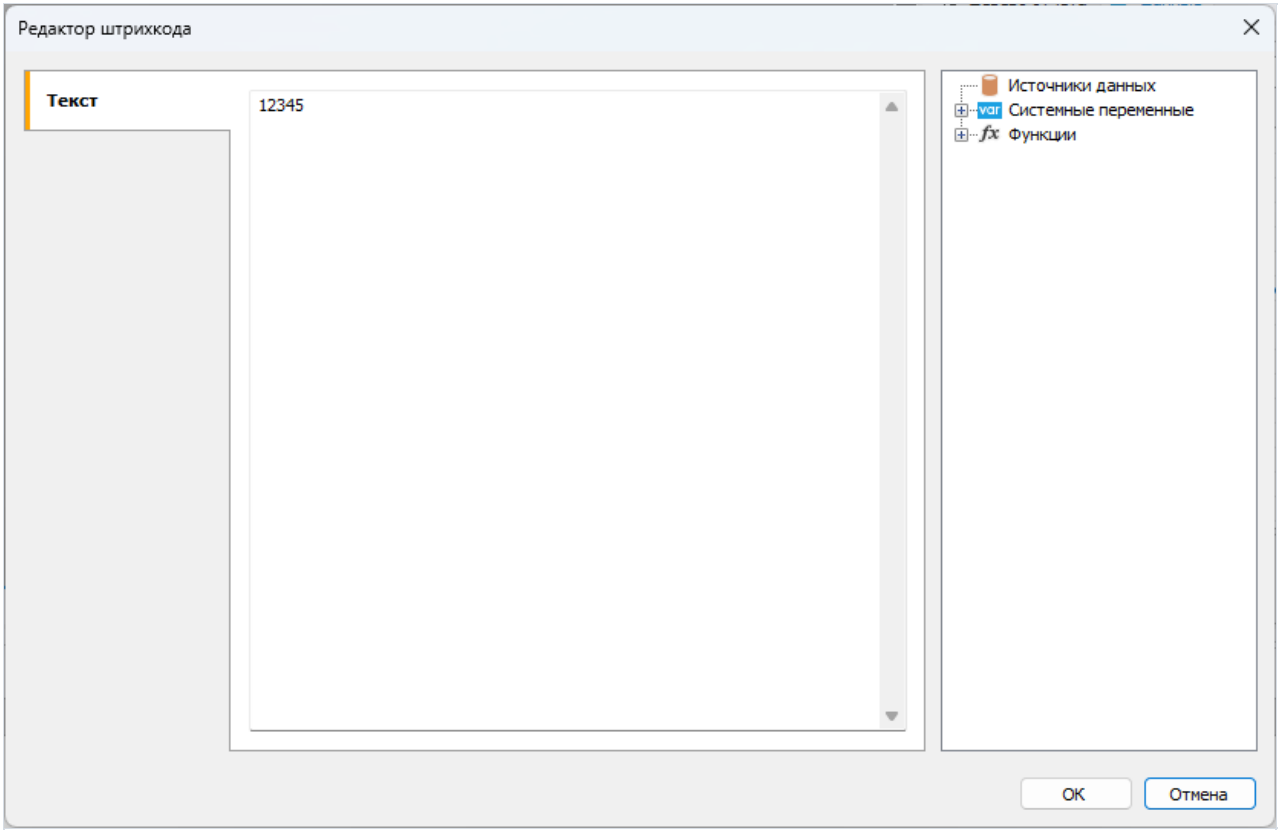
FastReport .NET позволяет создавать такие коды в своих отчётах. Вы можете разработать дизайн упаковки в генераторе сразу же со штрихкодом.

Для формирования штрихкода Pharmacode в FastReport .NET выберите объект Штрихкод (Barcode)  на панели компонентов в дизайнере отчётов. В выпадающем списке перейдите в категорию "Прочие", а затем Pharmacode:



После выбора штрихкода разместите его на странице отчёта.

С помощью двойного клика по добавленному штрихкоду открывается редактор. Также редактор кода можно открыть нажатием кнопки  в контекстном меню добавленного объекта, которое вызывается нажатием правой кнопки мыши:

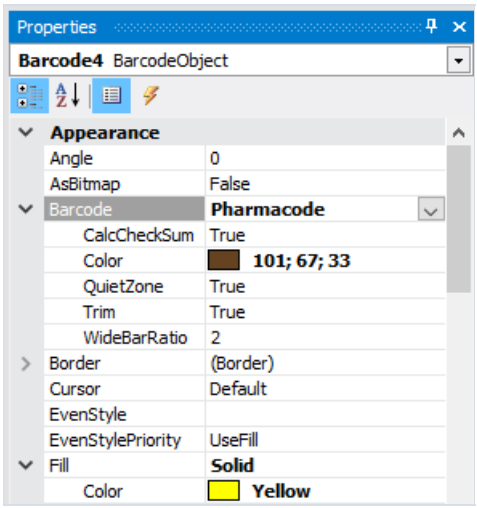


В качестве значения кода можно ввести числовую последовательность, задать функцию, переменную отчёта, значение из базы данных.

В свойствах штрихкода можно изменить промежуток между линиями (`WideBarRatio`), высоту кода (`Height`), отображение цифр под кодом (`ShowText`).

По умолчанию штрихкод имеет чёрный цвет на белом фоне. Изменить цвет кода можно в свойстве `Barcode`

-> `Color`. А цвет фона – в свойстве `Fill` -> `Color`:



В результате настройки цвета Pharmacode может выглядеть так:

Объект "Флажок"

Объект позволяет отображать в отчете флажок. Он выглядит следующим образом:

Объект может отображать два состояния: "включен" и "выключен". Вы можете задать состояние объекта следующим образом:

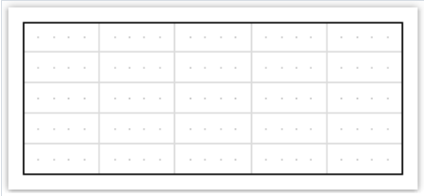
- указать состояние в логическом свойстве `Checked`;
- подключить объект к полю данных с помощью свойства `DataColumn`;
- указать в свойстве `Expression` выражение, возвращающее "истину" или "ложь".

Объект имеет следующие свойства:

Свойство	Описание
"Символы флажка" (CheckedSymbol, UncheckedSymbol)	Эти свойства определяют тип символа, который отображается во включенном и выключенном состоянии.
"Цвет флажка" (CheckColor)	В этом свойстве можно указать цвет символа флажка.
"Ширина флажка" (CheckWidthRatio)	Ширина флажка зависит от размеров объекта. Вы можете установить любое значение из диапазона 0,2 – 2. По умолчанию свойство равно 1.
"Прятать, если выключен" (HideIfUnchecked)	Позволяет скрыть объект, если он имеет состояние "Выключен".
"Включен" (Checked)	Позволяет напрямую управлять состоянием объекта.
"Поле данных" (DataColumn)	Поле данных, из которого загружать состояние объекта. Поле данных должно быть логического типа (`Boolean`).
"Выражение" (Expression)	Выражение, которое возвращает состояние объекта. Выражение должно возвращать `true` или `false` .

Объект "Таблица"

Объект "Таблица" состоит из строк, колонок и ячеек и представляет собой упрощенный аналог таблицы Microsoft Excel. Он выглядит следующим образом:



Подробнее об использовании этого объекта можно прочитать в главе "Построение отчетов".

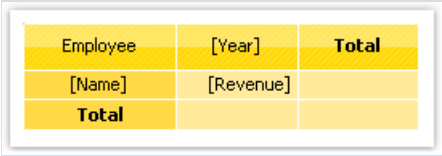
Объект имеет следующие свойства:

Свойство	Описание
"Количество колонок" (ColumnCount)	Позволяет быстро создать нужное количество колонок. Если колонок в таблице меньше, они добавляются, если больше – удаляются.
"Количество строк" (RowCount)	Позволяет быстро создать нужное количество строк. Если строк в таблице меньше, они добавляются, если больше – удаляются.
"Фиксированные колонки" (FixedColumns)	Определяет, сколько колонок в начале таблицы являются фиксированными. Фиксированные колонки образуют заголовок таблицы. Печатью заголовков управляет свойство "Повторять заголовки". Это свойство работает только для таблиц, которые строятся динамически.
"Фиксированные строки" (FixedRows)	Определяет, сколько строк в начале таблицы являются фиксированными. Фиксированные строки образуют заголовок таблицы. Печатью заголовков управляет свойство "Повторять заголовки". Это свойство работает только для таблиц, которые строятся динамически.
"Повторять заголовки" (RepeatHeaders)	Позволяет печатать заголовки таблицы на каждой новой странице. Это свойство работает только для таблиц, которые строятся динамически.

Объект "Матрица"

Объект "Матрица" является разновидностью таблицы и, как и объект "Таблица", состоит из строк, колонок и ячеек. Причем заранее неизвестно, сколько строк и столбцов будет в матрице - это зависит от данных, к которым она подключена.

Объект выглядит следующим образом:



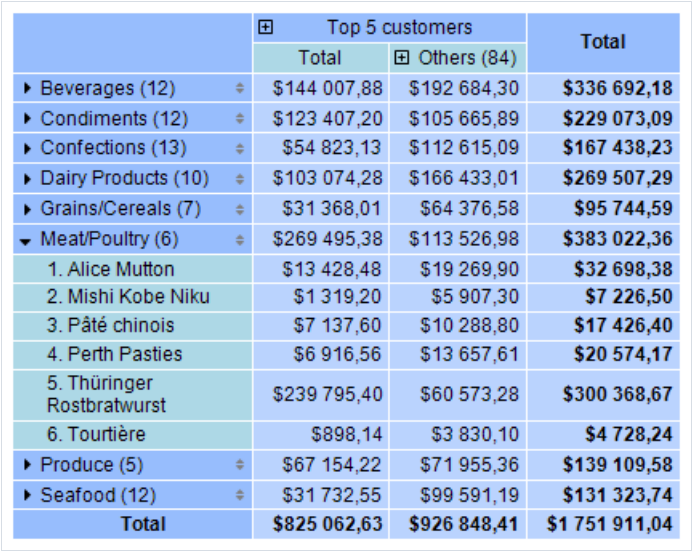
Подробнее об использовании этого объекта можно прочитать в главе "Построение отчетов".

Объект имеет следующие свойства:

Свойство	Описание
"Повторять заголовки" (RepeatHeaders)	Если матрица при печати разбивается на несколько страниц, это свойство позволяет печатать заголовки матрицы на каждой новой странице.
"Ячейки одной строкой" (CellsSideBySide)	Свойство определяет, как будут располагаться ячейки матрицы, если матрица имеет несколько значений в ячейках данных. Возможные варианты: - ячейки выводятся рядом (одной строкой); - ячейки выводятся друг под другом.
"Стиль" (Style)	Используя это свойство, можно задать стиль для всей матрицы. Вы можете выбрать один из предустановленных стилей.
"Авторазмер" (AutoSize)	Это свойство, будучи включенным, позволяет подбирать размеры матрицы автоматически. Выключите его, если вы хотите управлять размером объекта вручную.
"Источник данных" (DataSource)	Свойство позволяет подключить матрицу к источнику данных. Это свойство заполняется автоматически при перенесении поля данных в матрицу. Однако, если вы используете выражения в ячейках, проверьте, чтобы это свойство было установлено корректно.
"Фильтр" (Filter)	Это свойство содержит выражение для фильтрации данных, которое будет применено к источнику данных матрицы (см. свойство `DataSource`).

Объект "Улучшенная матрица"

Данный объект, как и объект "Матрица", позволяет строить сводные отчеты.

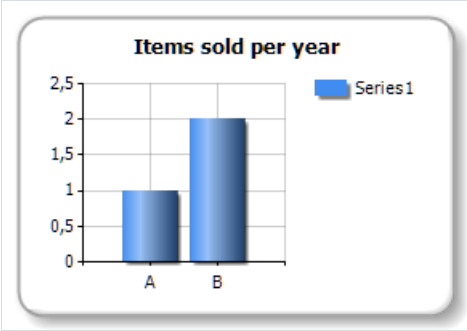


	Top 5 customers		Total
	Total	Others (84)	
Beverages (12)	$144 007,88	$192 684,30	**$336 692,18**
Condiments (12)	$123 407,20	$105 665,89	**$229 073,09**
Confections (13)	$54 823,13	$112 615,09	**$167 438,23**
Dairy Products (10)	$103 074,28	$166 433,01	**$269 507,29**
Grains/Cereals (7)	$31 368,01	$64 376,58	**$95 744,59**
Meat/Poultry (6)	$269 495,38	$113 526,98	**$383 022,36**
1. Alice Mutton	$13 428,48	$19 269,90	**$32 698,38**
2. Mishi Kobe Niku	$1 319,20	$5 907,30	**$7 226,50**
3. Pâté chinois	$7 137,60	$10 288,80	**$17 426,40**
4. Perth Pasties	$6 916,56	$13 657,61	**$20 574,17**
5. Thüringer Rostbratwurst	$239 795,40	$60 573,28	**$300 368,67**
6. Tourtière	$898,14	$3 830,10	**$4 728,24**
Produce (5)	$67 154,22	$71 955,36	**$139 109,58**
Seafood (12)	$31 732,55	$99 591,19	**$131 323,74**
Total	**$825 062,63**	**$926 848,41**	**$1 751 911,04**

Подробнее об использовании этого объекта можно почитать в главе Объект "Улучшенная матрица".

Объект "Диаграмма"

Объект "Диаграмма MS Chart" позволяет строить диаграммы. Всего доступно более 30 различных видов диаграмм - гистограммы, области, линии, пузырьки, круговые, лепестковые, финансовые, пирамидальные, диапазоны. Объект выглядит следующим образом:



Подробнее об использовании этого объекта можно прочитать в главе "Построение отчетов".

Объект имеет следующие свойства:

Свойство	Описание
"Диаграмма" (Chart)	Ссылка на объект Microsoft Chart.
"Выравнивать значения" (AlignXValues)	Свойство позволяет выравнивать значения в нескольких сериях (вставлять недостающие значения). Используется при выводе двух и более серий в одной диаграмме.
"Автосерии" (AutoSeriesColumn, AutoSeriesColor, AutoSeriesSortOrder)	Эти свойства позволяют настроить автоматически создаваемые серии. Подробнее см. в главе "Построение отчетов".
"Источник данных" (DataSource)	Свойство позволяет подключить диаграмму к источнику данных.
"Фильтр" (Filter)	Это свойство содержит выражение для фильтрации данных, которое будет применено к источнику данных (см. свойство DataSource).

Объект "Почтовый индекс"

Объект "Почтовый индекс" предназначен для печати почтового индекса на конвертах. Он способен отображать цифры от 0 до 9. Объект соответствует требованиям, изложенным в стандарте ГОСТ Р 51506-99.

Объект выглядит следующим образом:

Вы можете подключить объект к данным одним из следующих способов:

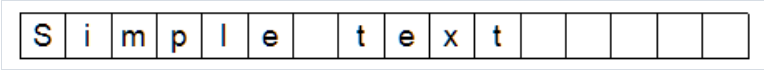
- указать строку, содержащую текст объекта, в свойстве `Text`;
- подключить объект к полю данных с помощью свойства `DataColumn`;
- указать в свойстве `Expression` выражение, которое возвращает текст объекта.

Объект имеет следующие свойства:

Свойство	Описание
"Количество сегментов" (SegmentCount)	Определяет количество сегментов в почтовом индексе. По умолчанию оно равно 6.
"Размер сегмента" (SegmentWidth, SegmentHeight)	Размер одного сегмента. Согласно стандарту, размер сегмента равен 0,5x1 см.
"Расстояние между сегментами" (Spacing)	Расстояние между левыми краями двух соседних сегментов. Согласно стандарту, расстояние равно 0,9 см.
"Показывать сетку" (ShowGrid)	Свойство определяет, надо ли показывать точечную сетку.
"Показывать маркеры" (ShowMarkers)	Свойство определяет, надо ли показывать маркеры (жирные горизонтальные линии над почтовым индексом).
"Поле данных" (DataColumn)	Поле данных, из которого загружать текст объекта.
"Выражение" (Expression)	Выражение, которое возвращает текст объекта.
"Текст" (Text)	Текст объекта.

Объект "Текст в ячейках"

Объект предназначен для печати текста, каждый символ которого выводится в отдельной ячейке. Такой объект часто используется при печати форм, например, налоговой отчетности. Он выглядит следующим образом:



В сущности, объект является наследником объекта "Текст". Вы можете подключить его к данным аналогичным образом. Для этого в текст объекта поместите ссылку на поле данных, например:

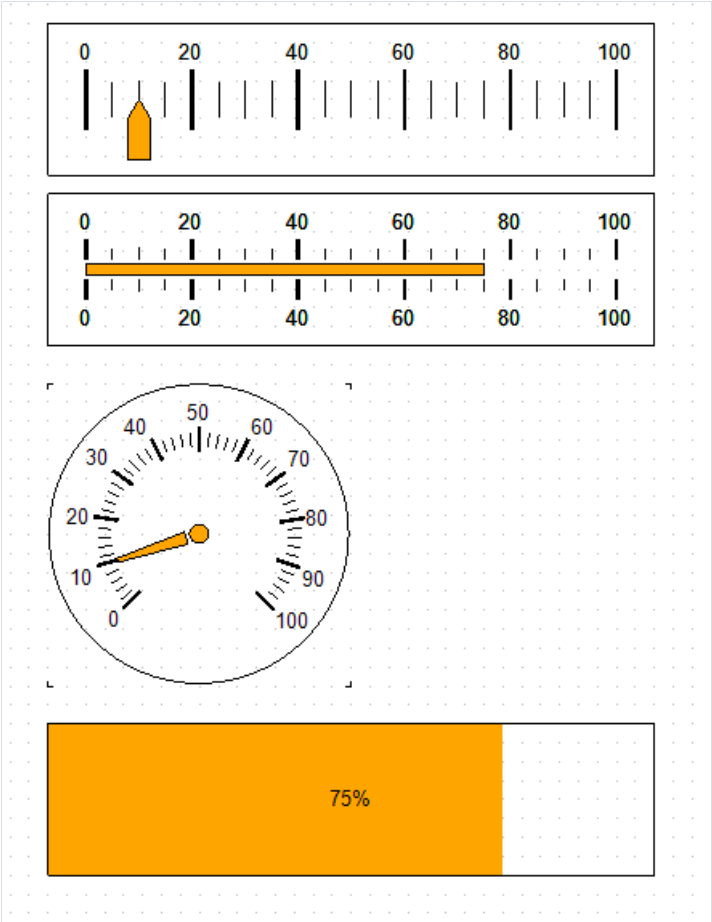
```
[Employees.FirstName]
```

Объект имеет следующие свойства:

Свойство	Описание
"Размеры ячейки" (CellWidth, CellHeight)	Свойства определяют размеры ячейки. Если оба свойства равны 0 (по умолчанию), размеры ячейки вычисляются автоматически, в зависимости от размера шрифта.
"Промежуток между ячейками" (HorzSpacing, VertSpacing)	Свойства определяют горизонтальный и вертикальный промежуток между соседними ячейками.

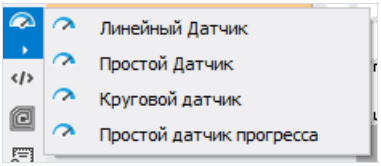
Группа объектов "Датчики"

Эти объекты предназначены для визуального отображения какого-либо значения. Вот как выглядят четыре различных вида датчиков, которые поддерживаются сейчас:

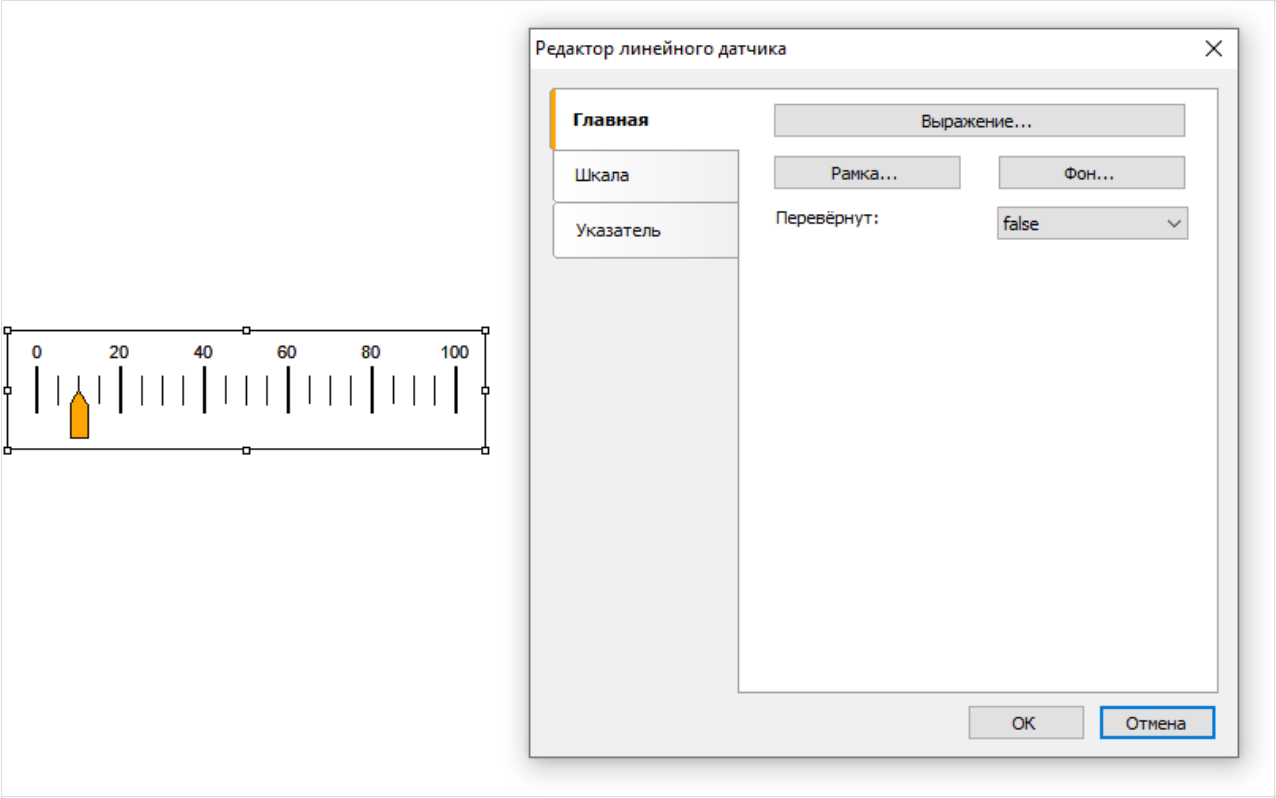


- линейный датчик;
- простой датчик;
- круговой датчик;
- линейный датчик прогресса.

Чтобы добавить объект "Датчик" в ваш отчёт, выберите один из вариантов из подменю, открывающегося при нажатии кнопки :



Можно изменить некоторые аспекты внешнего вида датчика, например, цвет указателя. Чтобы отредактировать параметры датчика, дважды нажмите на нём левой клавишей мыши, или же нажмите кнопку в контекстном меню.



Кнопка "Выражение" откроет текстовый редактор, где можно ввести значение датчика самостоятельно, или составить для него выражение и подключить датчик к данным.

Объект "Цифровая подпись"

Цифровая подпись – шифр, гарантирующий уникальность и неповторимость, позволяющий однозначно установить авторство и защитить от изменения документа. Благодаря надежным алгоритмам шифрования такие подписи ничем не хуже рукописных, а даже лучше, надежнее.

В текущей версии доступно два вида подписи:

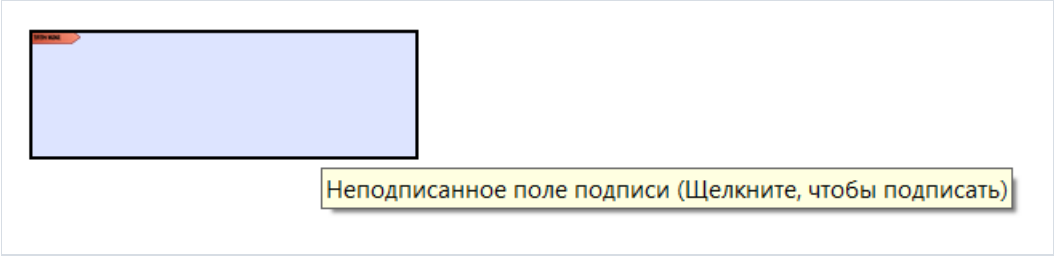
1. **Поле для подписывания** (signature field) - подразумевает наличие специального поля в документе, кликнув по которому, пользователь сможет прикрепить свой сертификат.

Объект, реализующий эту возможность, добавляется нажатием этой клавиши: 

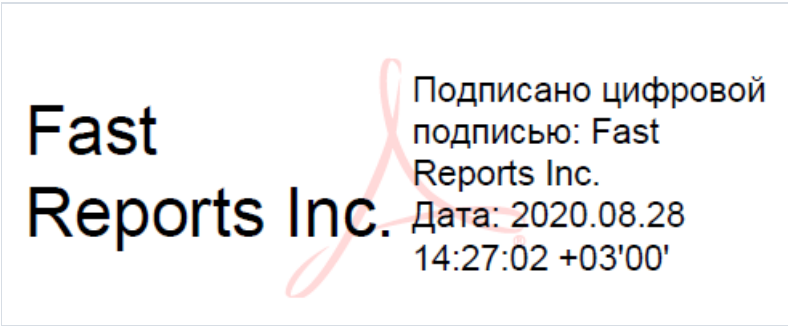
Он называется «Цифровая подпись» (Digital Signature). При размещении этого объекта на странице отчета он выглядит так:

В режиме просмотра подготовленного отчета он невидим. Его функциональность ограничивается исключительно PDF экспортом, то есть вы увидите это поле при просмотре PDF файла в Acrobat Reader или другого PDF-просмотрщика.

После экспорта, поле будет выглядеть так:

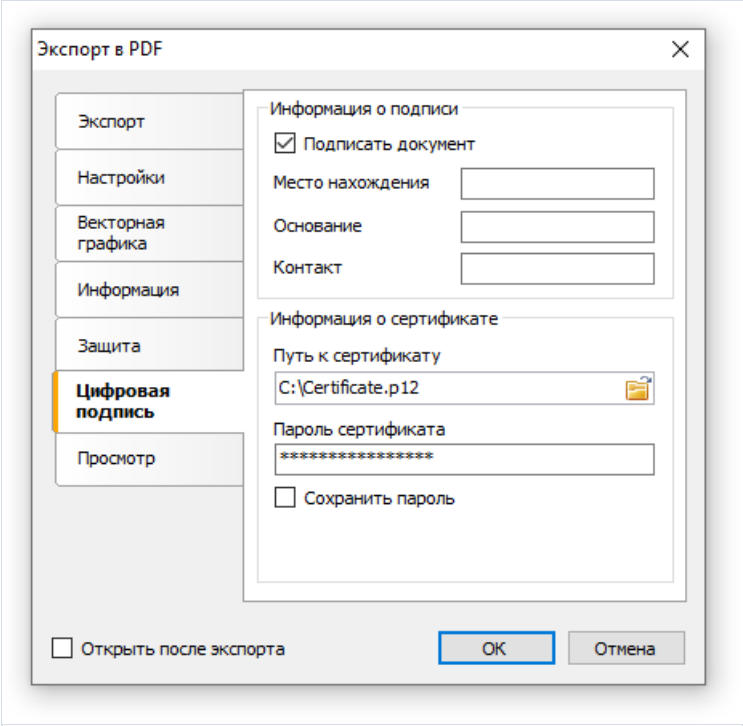


По нажатию откроется диалог выбора сертификата, и после ввода пароля поле изменит свой вид:



2. **Невидимая подпись** (invisible signature) – это подпись сертификатом. Визуально она не видна, но в свойствах документа можно получить информацию о подписанте, достоверности подписи, версии документа на момент подписания и др. информации.

Для того, чтобы подписать экспортируемый PDF-файл невидимой подписью, не нужно добавлять объект "Цифровая подпись" на страницу отчета. Нужно включить опцию подписывания в настройках экспорта, и там же выбрать сертификат:


Экспорт в PDF
Экспорт
Настройки
Векторная графика
Информация
Защита
Цифровая подпись
Просмотр
Информация о подписи
Подписать документ
Место нахождения
Основание
Контакт
Информация о сертификате
Путь к сертификату
C:\Certificate.p12
Пароль сертификата
****
Сохранить пароль
Открыть после экспорта
OK
Отмена

Первый отчет в FastReport

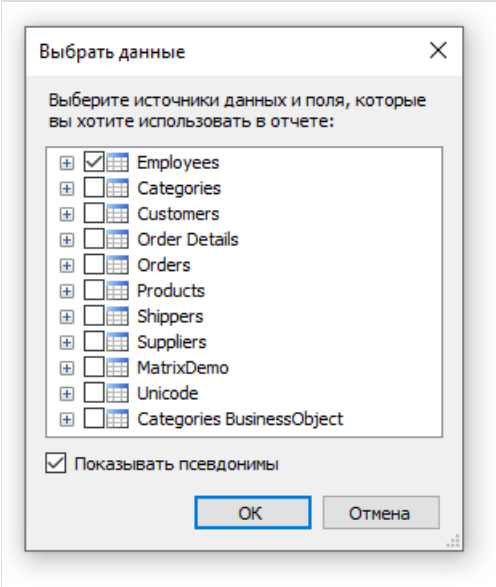
Давайте создадим простой отчет, который печатает список продуктов. В качестве источника данных будет использована таблица Products, которая входит в демонстрационную базу данных.

Предполагается, что вы будете выполнять описанные ниже действия в демонстрационной программе Demo.exe, из которой можно вызвать дизайнер отчетов.

Пример 1. Создание отчета вручную

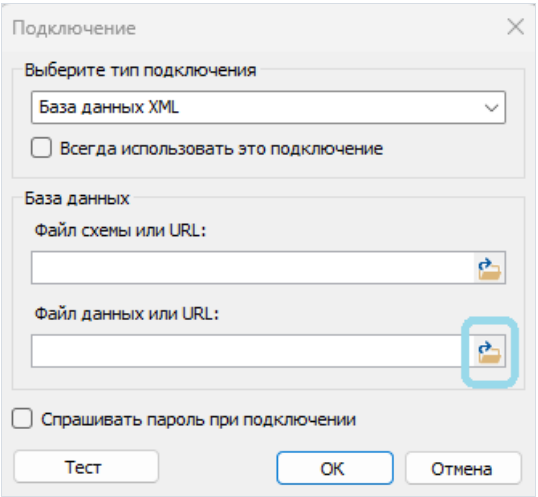
В этом примере мы создадим отчет вручную. Для этого выполните следующие действия:

- нажмите кнопку  на панели инструментов и в окне "Создать" выберите "Пустой отчет";
- в меню "Данные" выберите пункт "Выбрать данные для отчета..." и отметьте флажком источник данных "Products":

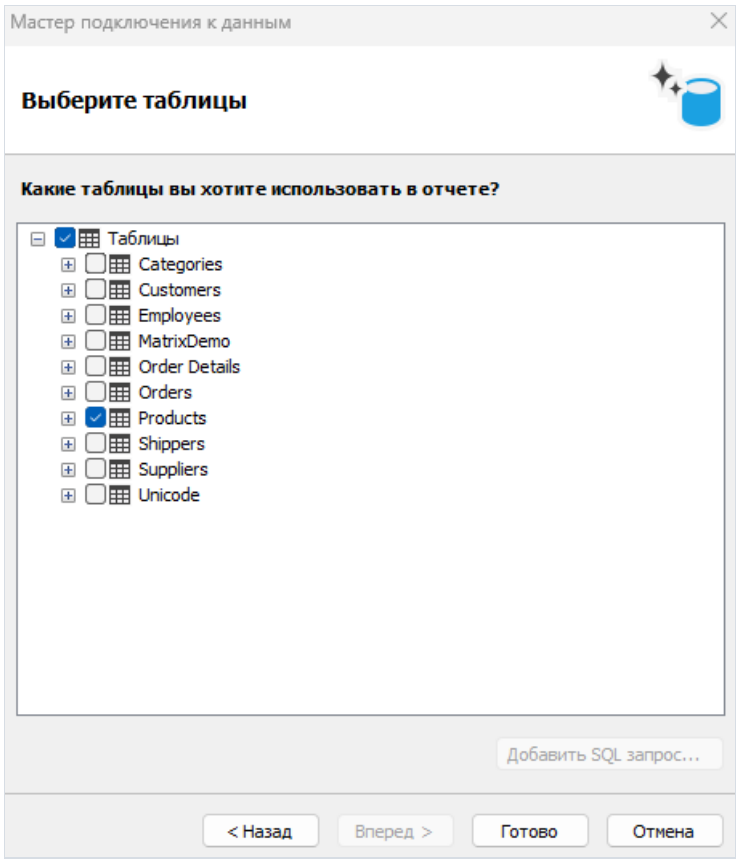


Второй вариант добавления своих данных из файла XML:

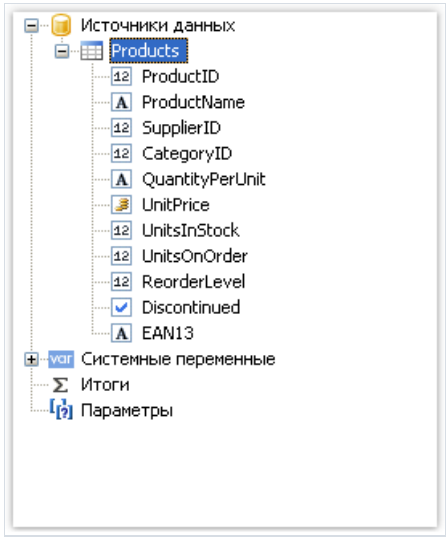
- в меню "Данные" выберите пункт "Новый источник данных". Откроется форма Мастера подключения к данным;
- нажмите на кнопку "Новое подключение";
- выберите тип подключения: база данных XML. Для выбора файла данных нажмите на значок файла данных или URL-адреса;



- в диалоговом окне "Открыть файл" перейдите к расположению файла, например, C:\Program Files (x86)\Fast Reports\2024.2.12\FastReport .NET Trial\Demos\Reports, затем выберите файл nwind.xml;
- для проверки подключения к базе данных нажмите на кнопку "Тест". Если тест подключения успешно завершен, нажмите кнопку "OK" для закрытия окна подключения;
- в Мастере подключения к данным нажмите кнопку "Вперед";
- разверните корневой узел "Таблицы", затем выберите таблицу "Products" и нажмите "Готово":



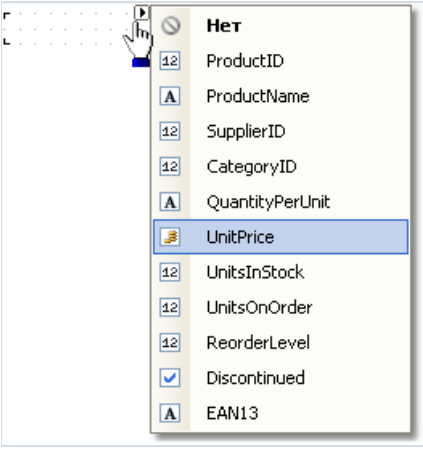
- переключитесь на служебное окно "Данные" (если оно не присутствует на экране, его можно показать, выбрав пункт меню "Данные|Показать окно данных") и раскройте элемент "Источники данных" и внутри него - "Products":



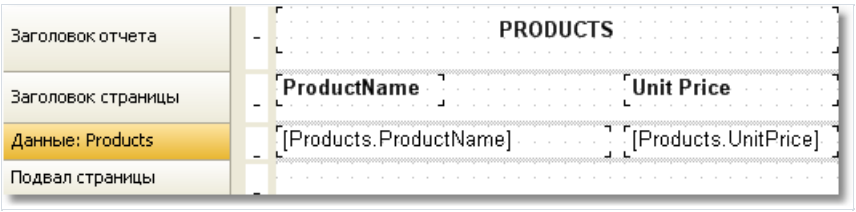
- перетащите поле `ProductName` на бэнд "Данные". FastReport создаст объект "Текст", который подключен к этому полю, и заголовок для него;
- второе поле - `UnitPrice` мы создадим другим способом. Для этого на панели инструментов "Объекты" нажмите кнопку "Текст":

- отпустите кнопку мыши и переместите ее указатель на рабочее поле дизайнера - вы увидите, что FastReport предлагает вставить объект. Выберите нужную позицию и нажмите кнопку мыши для вставки объекта;

- поместите указатель мыши над объектом и нажмите маленькую стрелку в верхнем правом углу объекта. Вы увидите список полей, из которого надо выбрать поле `UnitPrice`:



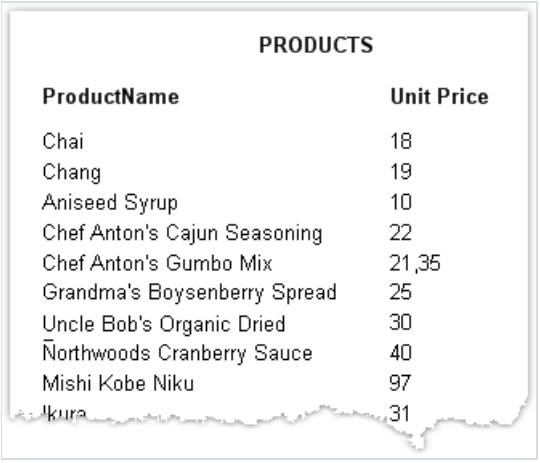
- создайте объект "Текст" - заголовок для поля `UnitPrice`, добавив описанным выше способ новый объект на бэнд "Заголовок страницы". Сделайте на объекте двойной щелчок и впишите текст "Unit Price";
- создайте объект "Текст" - заголовок отчета. Расположите его на бэнде "Заголовок отчета" и впишите текст "PRODUCTS";
- установите стиль шрифта "жирный" для объектов, лежащих на заголовке отчета и заголовке страницы. Для этого выделите три объекта, нажав клавишу `Shift`, и нажмите кнопку  на панели инструментов "Текст". После этого отчет будет выглядеть так:



Способы запуска отчета:

- с помощью кнопки  на панели управления;
- используя сочетание клавиш `Ctrl+P`.

Отчет будет построен и показан в окне предварительного просмотра:

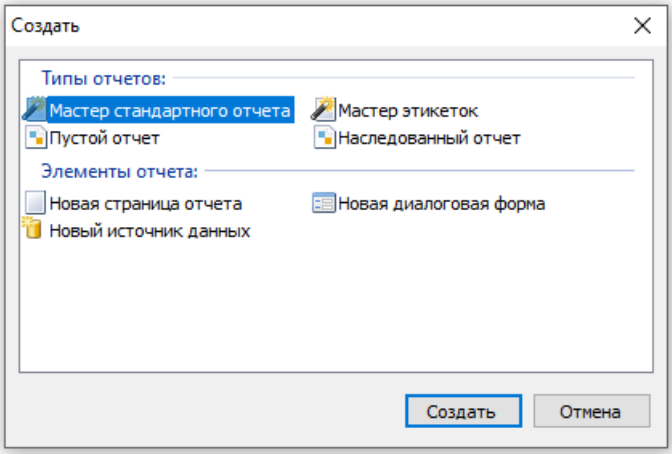

PRODUCTS

ProductName	Unit Price
Chai	18
Chang	19
Aniseed Syrup	10
Chef Anton's Cajun Seasoning	22
Chef Anton's Gumbo Mix	21,35
Grandma's Boysenberry Spread	25
Uncle Bob's Organic Dried	30
Northwoods Cranberry Sauce	40
Mishi Kobe Niku	97
Ikura	31

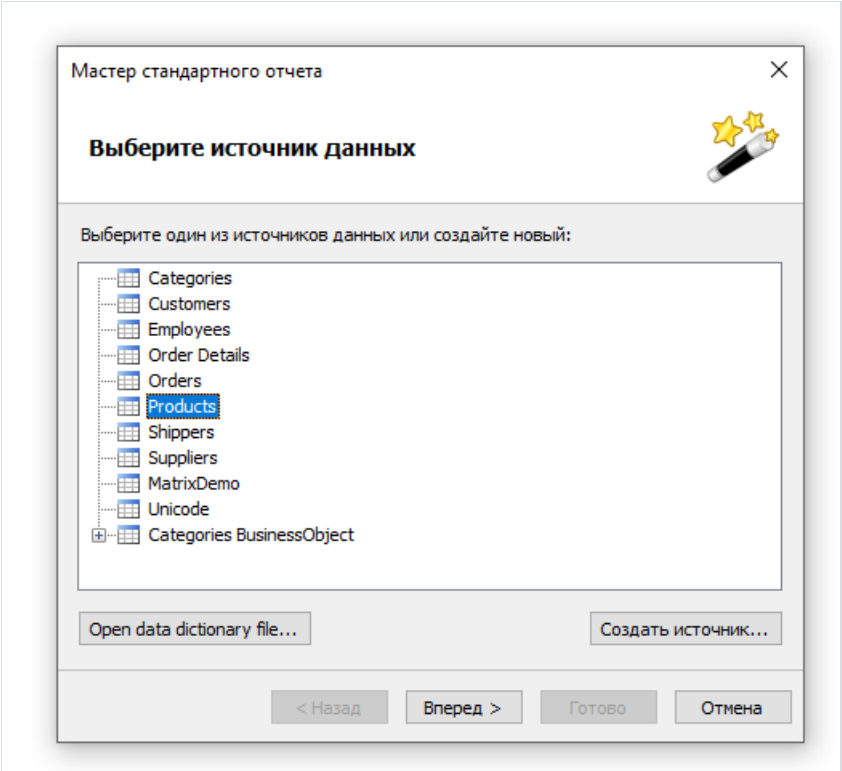
Пример 2. Создание отчета с помощью мастера

В этом примере мы создадим отчет с помощью "Мастера стандартного отчета". Для этого выполните следующие действия:

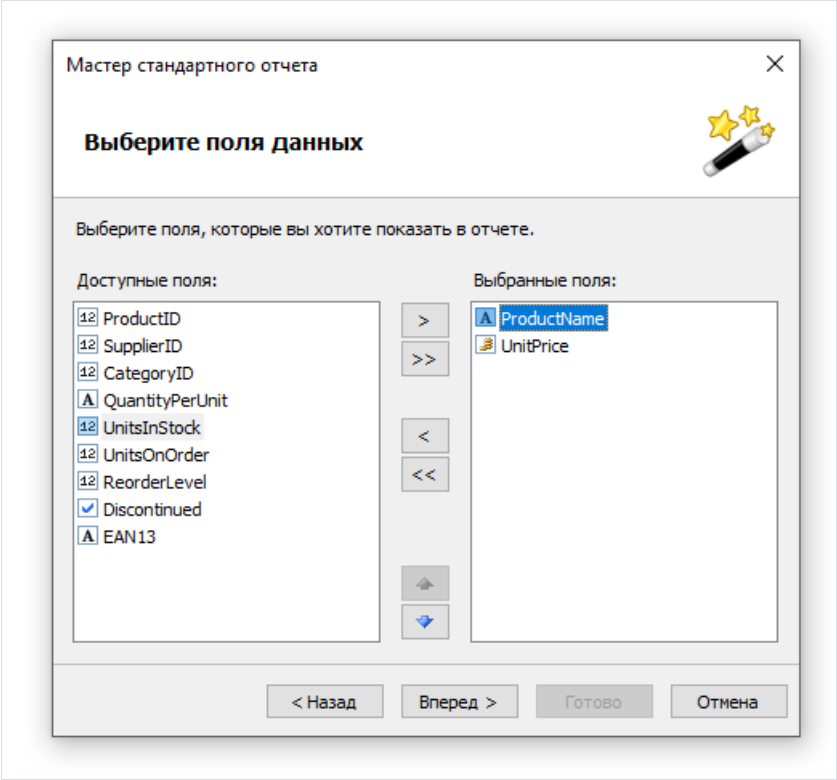
- нажмите кнопку  на панели инструментов и в окне "Создать" выберите "Мастер стандартного отчета":



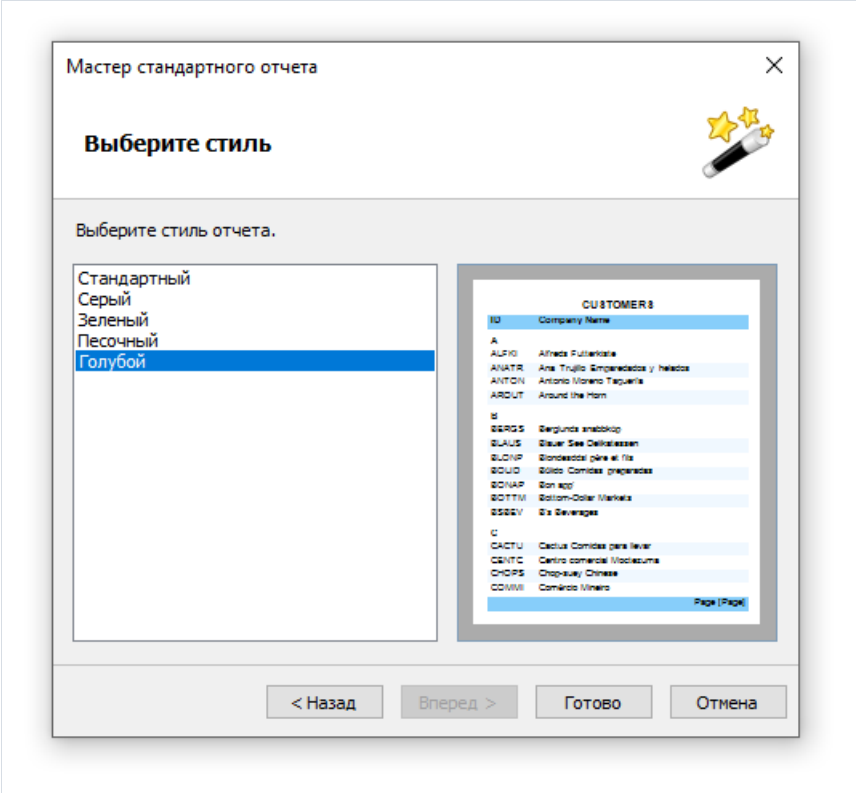
- на первом шаге мастера выберите таблицу "Products" и нажмите кнопку "Вперед":



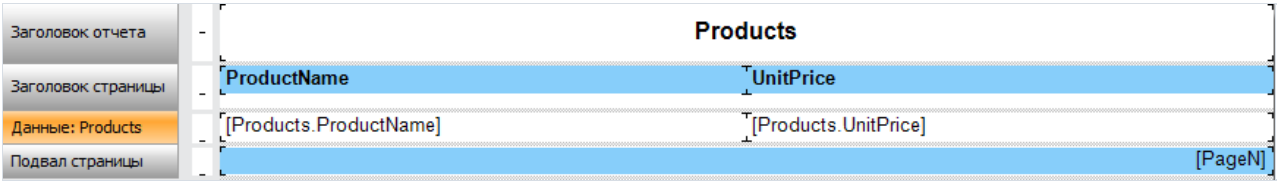
- на втором шаге мастера выберите поля `ProductName` и `UnitPrice` и перенесите их в список выбранных полей:



- остальные шаги мастера можно пропустить, нажимая кнопку "Вперед";
- на последнем шаге мастера выберите стиль Blue и нажмите кнопку "Готово":

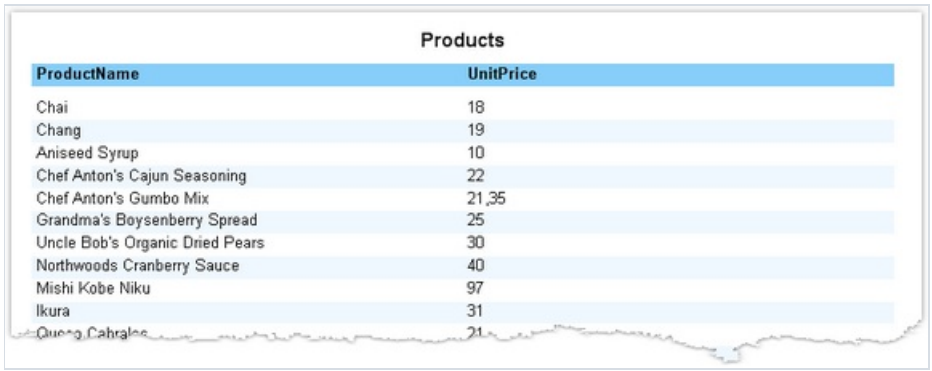


FastReport создаст отчет следующего вида:



Чтобы запустить отчет, нажмите кнопку  на панели управления. Отчет будет построен и показан в окне

предварительного просмотра:



Products

ProductName	UnitPrice
Chai	18
Chang	19
Aniseed Syrup	10
Chef Anton's Cajun Seasoning	22
Chef Anton's Gumbo Mix	21,35
Grandma's Boysenberry Spread	25
Uncle Bob's Organic Dried Pears	30
Northwoods Cranberry Sauce	40
Mishi Kobe Niku	97
Ikura	31
Queso Cabrales	21

Построение отчетов

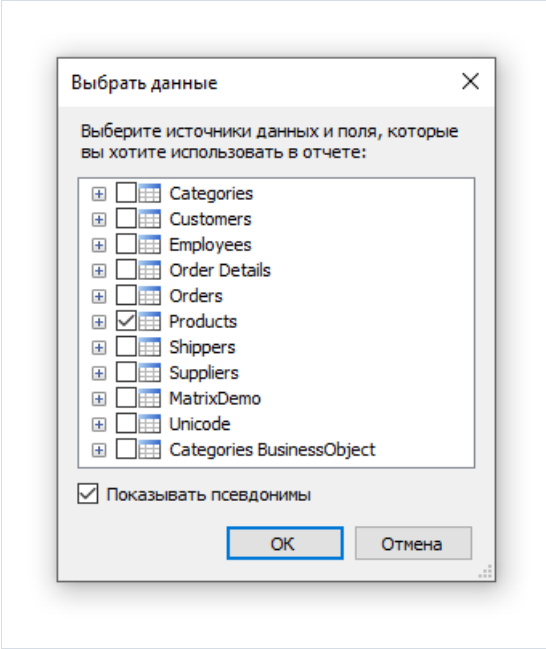
В этой главе будут описаны способы построения основных типов отчетов. Для построения любого отчета, как правило, нужно выполнить следующие действия:

1. Выбрать или создать данные, которые будут использованы в отчете.
2. Создать структуру отчета, добавив в него необходимые бэнды.
3. Подключить бэнды к источникам данных.
4. Разместить на бэндах объекты "Текст" для печати данных.
5. Задать форматирование у объектов.

Выбор данных для отчета

Прежде чем вы начнете построение отчета, вам необходимо выбрать данные, которые будут печататься в отчете. Вы можете сделать это двумя способами:

- вы выбираете один из источников данных, которые были зарегистрированы в отчете программным способом. Это можно сделать в меню "Данные|Выбрать данные для отчета...", отметив галочками нужные источники:



- вы создаете новый источник данных в меню "Данные|Новый источник данных...".

Подробнее об источниках данных можно прочитать в главе "Работа с данными".

Как только вы выбрали источник данных, он появляется в окне "Данные". С этого момента вы можете использовать этот источник в отчете. Как правило, большинство отчетов использует один источник данных. Для отчетов типа "master-detail" нужно выбрать два источника, между которыми установлена связь (о связях можно прочитать в главе "Работа с данными"). Несколько источников также может понадобиться в отчете, который печатает данные из связанных источников (например, из справочников).

Автоматический подбор высоты объектов

Часто приходится печатать текст, размер которого на этапе создания отчета неизвестен. Например, это может быть описание товара. В этом случае приходится решать следующие задачи:

- подобрать высоту объекта так, чтобы он вместил весь текст;
- подобрать высоту бэнда так, чтобы он вместил объекты с переменным количеством текста;
- сдвинуть или изменить высоту других объектов, имеющихся на бэнде, так, чтобы не нарушить общее оформление отчета.

Эти задачи можно решить, используя некоторые свойства объектов и бэндов:

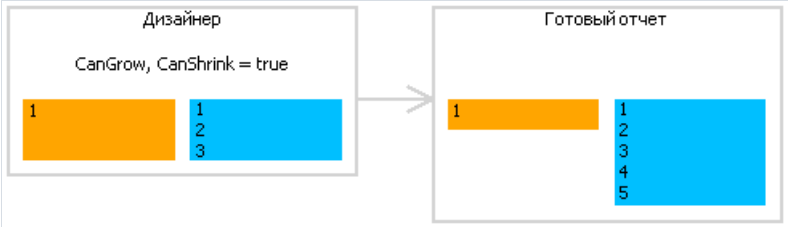
- свойства "Может расти" и "Может сжиматься" позволяют подбирать высоту объекта автоматически;
- свойство "Сдвиг" позволяет сдвигать объекты, расположенные под расширяемыми объектами;
- свойство "Расти вниз" позволяет дотягивать объекты до нижнего края бэнда;
- свойства "Якорь" и "Стыковка" позволяют управлять размером объектов в зависимости от размера бэнда.

Все эти свойства будут рассмотрены ниже.

Свойства "Может расти", "Может сжиматься" (CanGrow, CanShrink)

Эти свойства имеются у бэндов и объектов отчета. Свойства определяют, может ли объект расти или сжиматься в зависимости от размера своего содержимого. Если оба свойства отключены, объект всегда имеет размер, заданный в дизайнере.

Эти свойства очень полезны, если надо напечатать текст, размер которого неизвестен на этапе дизайна. Для того, чтобы объект смог вместить весь текст, у него нужно включить свойства "Может расти"/"Может сжиматься":

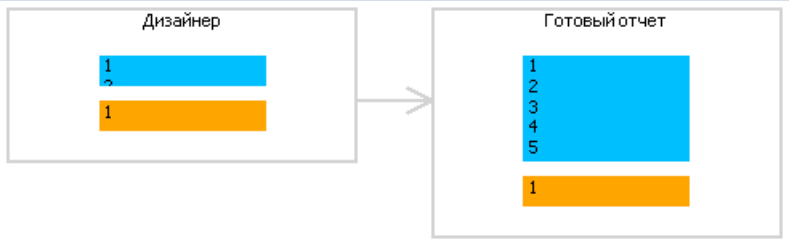


Также эти свойства надо включить у бэнда, чтобы он расширился или сжался в зависимости от размера объектов, которые на нем расположены. Менять размер бэнда могут следующие объекты:

- "Текст";
- "Форматированный текст";
- "Рисунок" (с включенным свойством "Авторазмер");
- "Таблица".

Свойство "Сдвиг" (ShiftMode)

Это свойство имеется у объектов отчета. Его можно изменить в окне "Свойства". Объект с включенным свойством будет сдвинут вниз или вверх, если над ним имеется объект, который расширяется или сжимается:



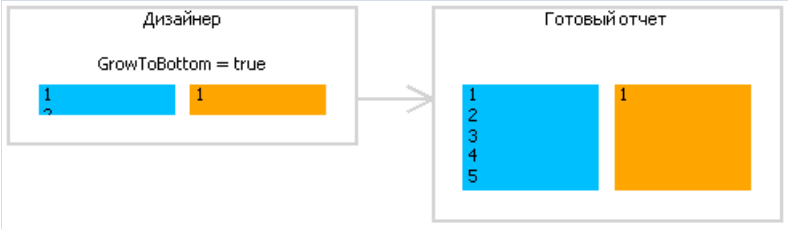
Свойство `ShiftMode` может иметь одно из значений:

- всегда (Always) - объект нужно сдвигать всегда. Это значение по умолчанию.
- никогда (Never) - сдвигать объект не нужно.
- при перекрытии (WhenOverlapped) - сдвигать объект нужно в том случае, если расширяющийся объект находится строго над ним (т.е. оба объекта перекрываются по горизонтали).

Это свойство удобно использовать при печати информации табличного вида, когда несколько ячеек таблицы находятся друг над другом и могут иметь переменное количество текста.

Свойство "Расти вниз" (GrowToBottom)

Это свойство имеется у объектов отчета. Объект с включенным свойством при печати растягивается до нижней границы бэнда, независимо от того, сколько в нем текста:



Это бывает необходимо при печати информации табличного вида. На бэнде может находиться несколько объектов, которые могут растягиваться. Это свойство позволяет "дотягивать" остальные объекты, чтобы не нарушать внешний вид таблицы.

Свойство "Якорь" (Anchor)

Это свойство есть у объектов отчета. Оно определяет, как будет изменяться позиция объекта и/или его размеры при изменении размеров контейнера, на котором он лежит. Используя якорь, можно сделать так, чтобы объект расширялся или сдвигался синхронно с контейнером.

Контейнер, о котором идет речь, в большинстве случаев является бэндом. Но это может быть и объект "Таблица" или "Матрица", которые могут содержать внутри другие объекты.

Свойство может иметь одно из следующих значений, а также любую комбинацию этих значений:

Значение	Описание
Left	Заякорен левый край объекта. При изменении ширины контейнера объект не будет смещаться влево/вправо.
Top	Заякорен верхний край объекта. При изменении высоты контейнера объект не будет смещаться вверх/вниз.
Right	Заякорен правый край объекта. При изменении ширины контейнера расстояние между правыми краями объекта и контейнера будет постоянным. Если при этом заякорен левый край объекта, объект будет расти и сжиматься синхронно с контейнером.
Bottom	Заякорен нижний край объекта. При изменении высоты контейнера расстояние между нижними краями объекта и контейнера будет постоянным. Если при этом заякорен верхний край объекта, объект будет расти и сжиматься синхронно с контейнером.

По умолчанию значение этого свойства равно `Left, Top`. Это значит, что при изменении размеров контейнера объект меняться не будет. В таблице ниже приведены некоторые часто используемые комбинации значений:

Значение	Описание
Left, Top	Значение по умолчанию. Объект не меняется при изменении размеров контейнера.
Left, Bottom	Объект смещается вверх/вниз при изменении высоты контейнера. Положение объекта относительно нижнего края контейнера остается неизменным.
Left, Top, Bottom	При изменении высоты контейнера высота объекта изменяется синхронно с ним.
Left, Top, Right, Bottom	При изменении ширины и высоты контейнера объект растет/сжимается синхронно с ним.

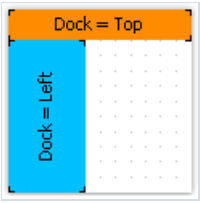
Свойство "Стыковка" (Dock)

Это свойство есть у объектов отчета. Оно определяет, к какой стороне контейнера будет пристыкован объект.

Свойство может иметь одно из следующих значений:

Значение	Описание
None	Значение по умолчанию. Объект не стыкуется.
Left	Объект пристыкован к левому краю контейнера. При этом высота объекта будет равна высоте контейнера*.
Top	Объект пристыкован к верхнему краю контейнера. При этом ширина объекта будет равна ширине контейнера*.
Right	Объект пристыкован к правому краю контейнера. При этом высота объекта будет равна высоте контейнера*.
Bottom	Объект пристыкован к нижнему краю контейнера. При этом ширина объекта будет равна ширине контейнера*.
Fill	Объект занимает все свободное место контейнера.

* - это не совсем так, если одновременно стыкуется несколько объектов. На рисунке ниже показано два объекта, первый пристыкован к верхнему краю контейнера, второй – к левому:



Как видно, высота второго объекта равна высоте свободного места, которое осталось после стыковки первого объекта.

> Стыковка объектов зависит от порядка их создания. Его можно изменить в контекстном меню объекта с помощью пунктов "На передний план" и "На задний план".

Форматирование

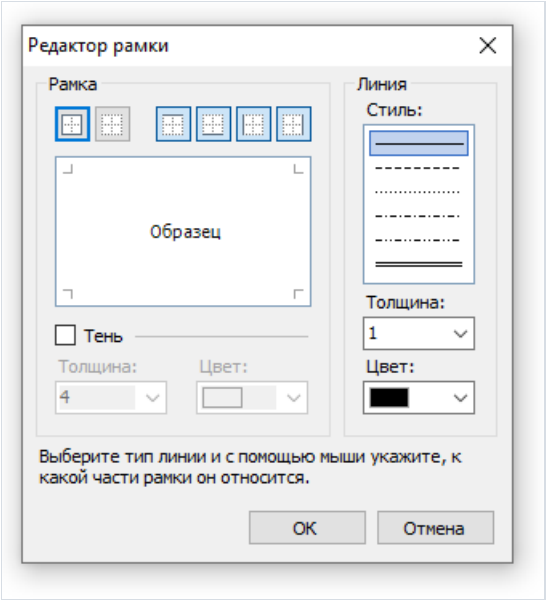
В этом разделе рассматриваются следующие вопросы:

- изменение внешнего вида объекта;
- изменение формата печатаемых значений;
- автоматическое изменение внешнего вида объекта при выполнении какого-либо условия;
- скрытие ненужных значений;
- выделение четных строк данных другим цветом.

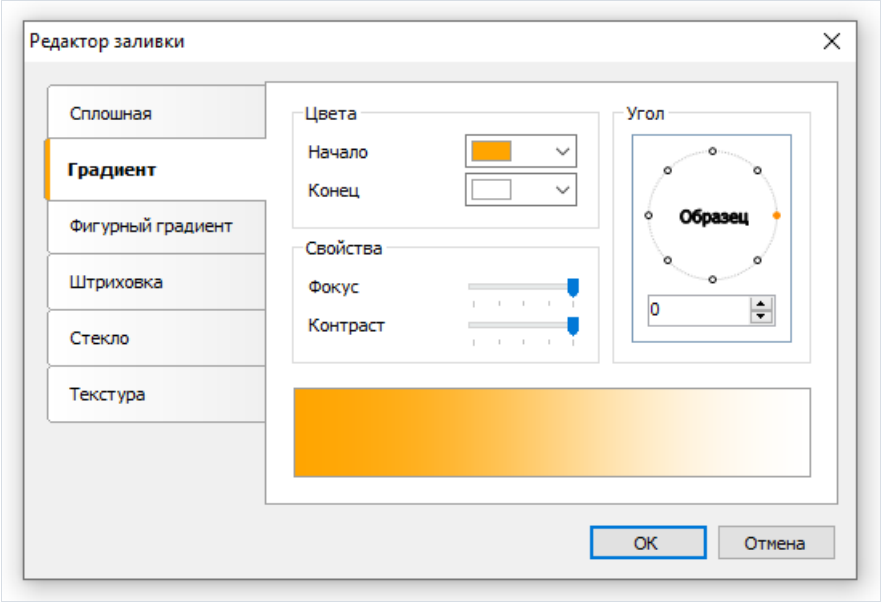
Рамка и заливка

Практически все объекты имеют рамку (свойство `Border`) и заливку (свойство `Fill`). Для работы с ними используйте панель инструментов "Рамка и заливка":

Рамка объекта состоит из четырех линий, каждая из которых может иметь разную толщину, цвет и стиль. Кнопки на панели инструментов меняют настройки для всех линий рамки. Кнопка  вызывает диалоговое окно, в котором можно настроить каждую линию отдельно:

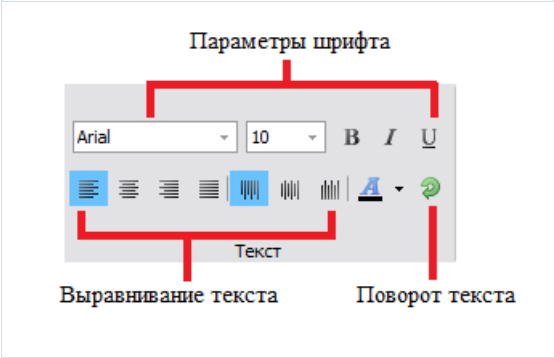


Для работы с заливкой на панели "Рамка и заливка" есть две кнопки. Кнопка  позволяет сделать сплошную заливку выбранным цветом. Кнопка  вызывает диалоговое окно, в котором можно выбрать тип заливки и настроить все ее параметры:


Редактор заливки
Сплошная
Градиент
Фигурный градиент
Штриховка
Стекло
Текстура
Цвета
Начало
Конец
Свойства
Фокус
Контраст
Угол
Образец
0
OK
Отмена

Формат текста

Для изменения внешнего вида объекта "Текст" используйте панель инструментов "Текст":

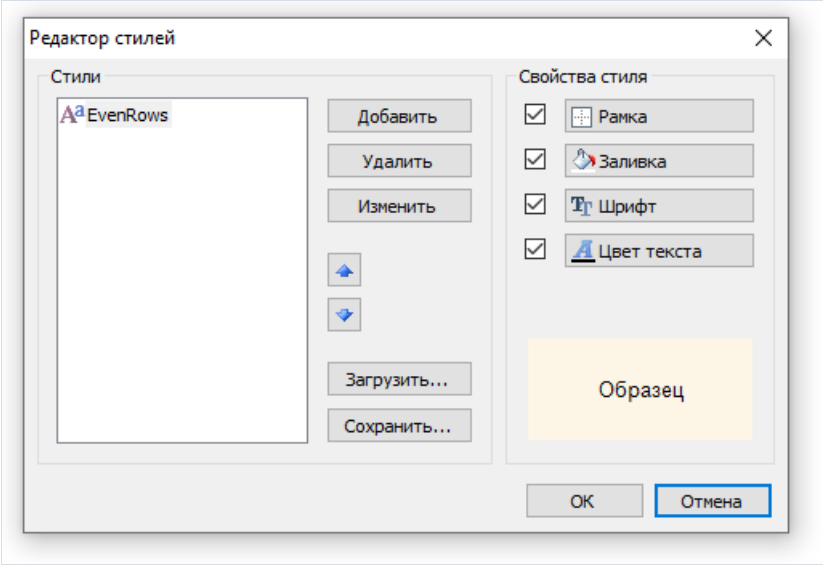


Стили

Для оформления объектов можно использовать стили. Стиль – это набор следующих свойств:

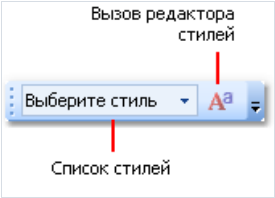
- рамка;
- заливка;
- шрифт;
- цвет текста.

Список стилей содержится в отчете. Вы можете управлять им из меню "Отчет|Стили..." или с помощью кнопки  на панели инструментов "Стиль":



Выбрать стиль для объекта можно двумя способами:

- установить значение свойства `Style` в окне "Свойства";
- использовать панель инструментов "Стили":



> Если панель отсутствует на экране, ее можно показать в меню "Вид|Панели инструментов|Стиль".

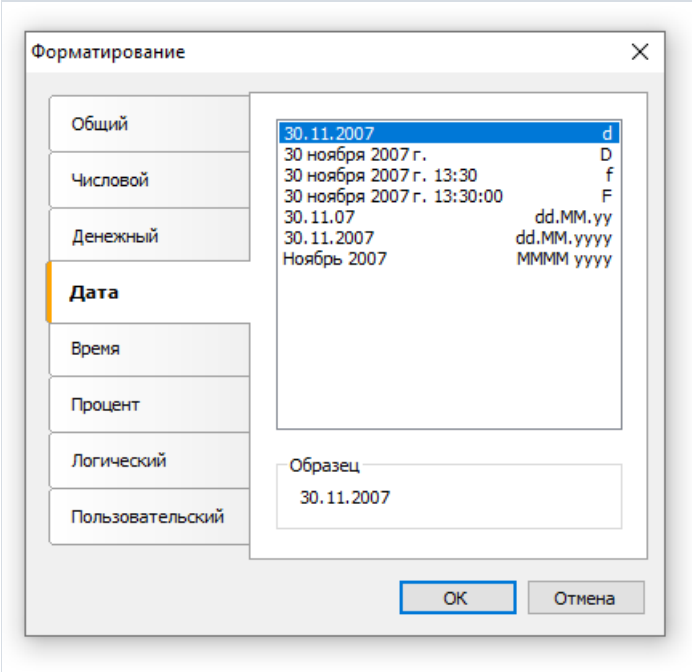
При этом объект станет выглядеть так, как указано в стиле. Если параметры стиля поменяются, внешний вид объекта изменится автоматически.

Формат данных

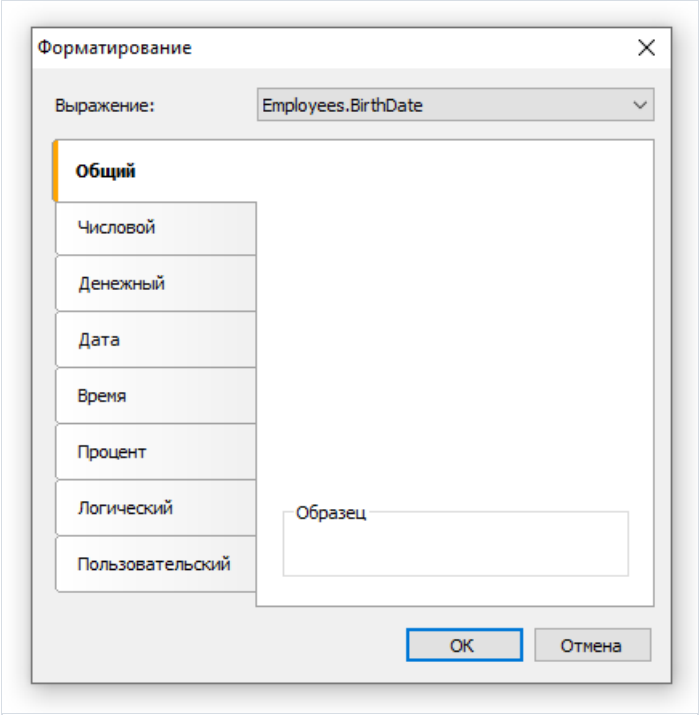
Для печати текстовой информации в отчете используется объект "Текст". По умолчанию он печатает данные в том виде, в котором они представлены в источнике данных. Например, поле данных типа "Дата" помимо даты содержит и время (это особенность типа `DateTime` в .NET). При печати такого поля в отчете вы увидите примерно следующее:

```
02.04.2024 18:04:52
```

Если вам нужно напечатать только дату, используйте форматирование данных. Для этого выделите объект "Текст" и нажмите правую кнопку мыши, чтобы вызвать контекстное меню. В меню выберите пункт "Формат данных...". Вы увидите следующий диалог:



В диалоге можно выбрать один из доступных форматов, а также задать собственную строку форматирования, выбрав формат "Пользовательский". Если в тексте объекта есть несколько выражений, вы можете задать форматирование для каждого из них. Для этого в верхней части окна выберите нужное выражение и задайте формат:

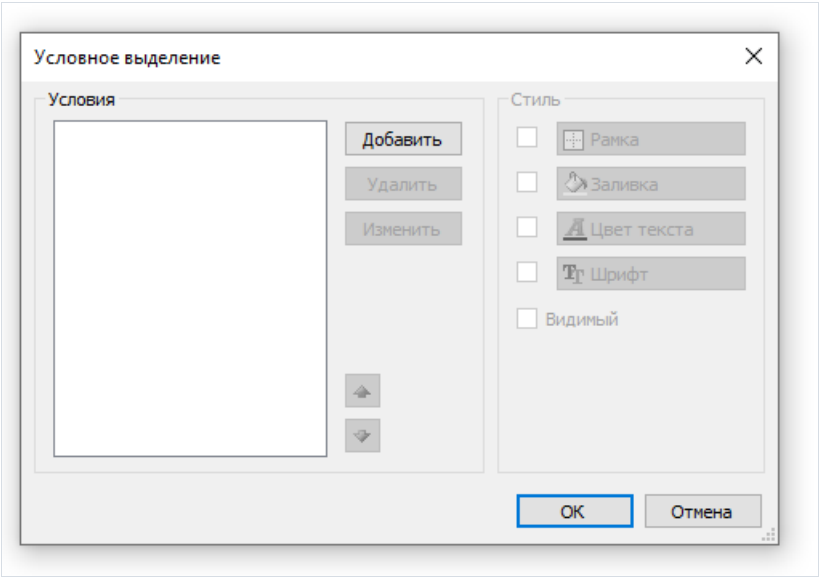


Вы также можете форматировать данные, используя системную функцию `String.Format`. Справку по этой функции можно найти в справочной системе MSDN.

```
Сегодня [String.Format("{0:d}", [Date])], страница [Page]
```

Условное выделение

В объекте "Текст" предусмотрена возможность смены внешнего вида объекта в зависимости от заданных условий. Например, объект можно выделить красным цветом, если он содержит отрицательное значение. Эта возможность называется "условное выделение". Для ее настройки выберите объект и нажмите кнопку  на панели инструментов "Текст". Вы увидите следующее диалоговое окно:

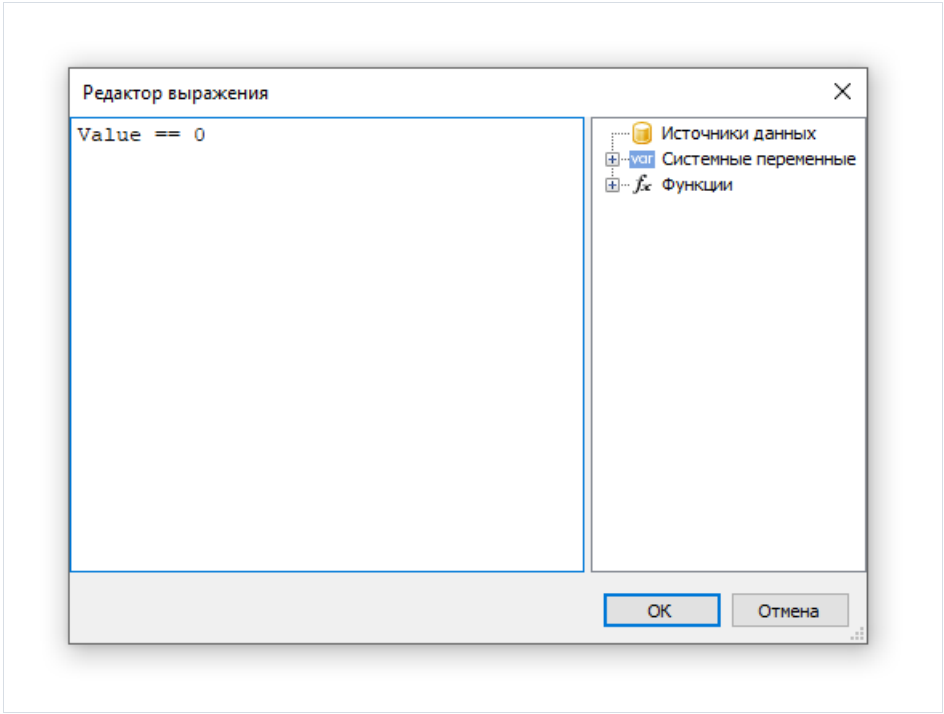


Здесь можно определить одно или несколько условий и задать стиль для каждого условия. Стиль может содержать одно или несколько свойств:

- заливка;
- цвет текста;
- параметры шрифта;
- видимость объекта.

Можно указать какие свойства необходимо менять при срабатывании условия. Для этого используйте флажки в правой части окна. По умолчанию новый стиль имеет одну настройку – цвет текста.

Для того чтобы создать новое условие, нажмите кнопку "Добавить":



Здесь можно написать любое выражение, которое возвращает логический результат (да/нет). В большинстве случаев в выражении участвует текущее печатаемое значение, которое доступно через переменную `Value`.

Рассмотрим следующий пример: есть объект "Текст", в котором печатается остаток товара на складе:

```
[Products.UnitsInStock]
```

Необходимо подсветить объект красным цветом, если количество товара = 0. Для этого создадим следующее условие:

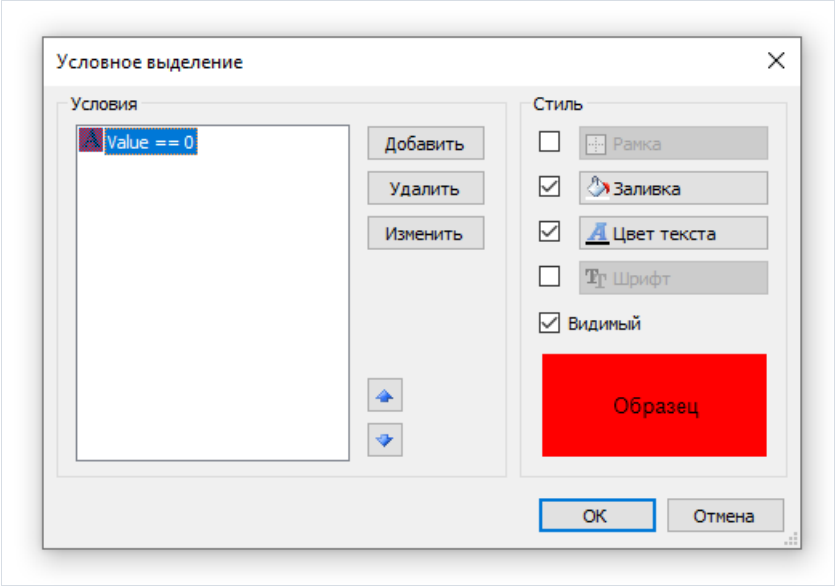
```
Value == 0
```

В данном случае мы использовали переменную `Value`, которая содержит печатаемое значение. Если в объекте есть несколько выражений, эта переменная будет содержать значение последнего выражения. Вместо `Value` можно использовать поле из источника данных, например:

```
[Products.UnitsInStock] == 0
```

> Операция сравнения написана в стиле C#. Это верно, если в качестве языка отчета выбран C#. Для языка VisualBasic.NET операция сравнения – одиночный знак `=`. Язык отчета можно поменять в меню "Отчет|Свойства...".

Настроим стиль для данного условия так, чтобы использовалась только заливка, и укажем цвет заливки – красный:



При печати объекты, содержащие нулевое значение, будут красными. Теперь усложним наш пример, добавив еще одно условие. Если остаток товара меньше 10, он должен быть напечатан желтым цветом. Для этого откроем редактор условий и нажмем кнопку "Добавить". Второе условие будет выглядеть так:

```
Value < 10
```

В случае, когда указано несколько условий, FastReport проверяет все условия, начиная с первого. Если какое-то условие выполняется, FastReport применяет его стиль к объекту, и процесс завершается. Здесь важно расставить условия в правильном порядке. Так, порядок, который мы рассмотрели в этом примере, правильный:

```
1. Value == 0
2. Value < 10
```

Если условия поменять местами, то выделение будет работать неправильно.

```
1. Value < 10
2. Value == 0
```

В данном случае условие `Value == 0` выполняться не будет, потому что при нулевом значении сработает первое условие. Для того чтобы поменять порядок условий, используйте кнопки и в редакторе условий.

Скрытие нулевых значений

Объект "Текст" имеет свойство `HideZeros`, которое позволяет скрывать нулевые значения. Рассмотрим объект со следующим содержимым:

```
Всего элементов: [CountOfElements]
```

Если значение переменной `CountOfElements` равно 0, и свойство `HideZeros` равно `true`, то объект будет напечатан так:

```
Всего элементов:
```

Объект "Текст" также имеет свойство `HideValue` строкового типа, которое позволяет скрывать значения выражений, равные заданному значению. Например, если значение свойства равно 0, то будут скрыты все нулевые поля. Это свойство также можно использовать для скрытия "нулевых" дат. Как правило, это даты 1.1.0001 или 1.1.1900. В этом случае значение свойства `HideValue` должно быть таким:

```
1.1.1900 0:00:00
```

Как можно увидеть, кроме даты надо написать и время. Это необходимо, потому что значение даты в .NET содержит и время.

> Важное замечание: рассматриваемый механизм зависит от региональных установок, заданных в Панели управления. Это происходит потому, что FastReport сравнивает строки, используя метод `ToString()` у значения выражения. Этот метод преобразует значение в строку, используя текущие региональные установки. В связи с этим, будьте осторожны при разработке отчетов, которые могут быть запущены на компьютере с другими региональными установками.

Наконец, свойство `NullValue` объекта "Текст" позволяет выводить какой-либо текст вместо пустых (`null`) значений. Часто это применяется для того, чтобы напечатать прочерк вместо пустого значения. Рассмотрим объект со следующим содержимым:

```
Всего элементов: [CountOfElements]
```

Если значение переменной `CountOfElements` равно `null`, и свойство `NullValue` равно `--`, то объект будет напечатан так:

```
Всего элементов: --
```

Скрытие повторяющихся значений

У объекта "Текст" есть свойство `Duplicates`, позволяющее скрывать повторяющиеся значения. Это свойство работает только для объектов "Текст", которые лежат на бэнде "Данные". Повторяющимися считаются одинаковые значения, напечатанные в соседних строках данных.

Свойство может иметь одно из следующих значений:

- показывать (Show) - показывать повторяющиеся значения (значение по умолчанию);
- прятать (Hide) - прятать объект с повторяющимся значением;
- очищать (Clear) - очищать текст в объекте, но показывать сам объект;
- объединять (Merge) - объединять объекты с одинаковыми значениями.

На рисунке продемонстрировано отличие между режимами:

Выделение строк данных через одну

Для улучшения внешнего вида отчета можно выделить четные строки данных другим цветом. Сделать это можно, используя свойство `EvenStyle` у бэнда или объектов, лежащих на бэнде. Этому свойству присваивается имя стиля, который будет использован для четных строк бэнда.

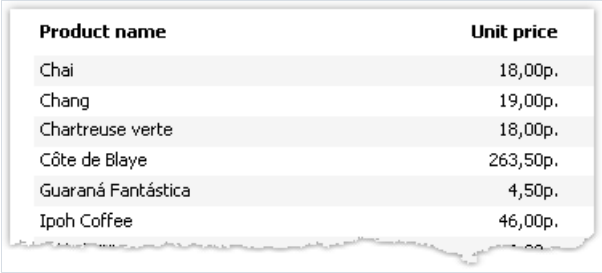
> Предпочтительнее использовать свойство `EvenStyle` у объектов, лежащих на бэнде. Это позволит избежать возможных проблем при экспорте отчета.

Чтобы настроить выделение строк, сделайте следующее:

- определите стиль, который будет использован для выделения строк. Это можно сделать в меню "Отчет|Стили...". Как правило, для выделения используется бледно-серый цвет заливки.
- имя нового стиля укажите в свойстве `EvenStyle` у бэнда или его объектов.

По умолчанию, объекты используют только заливку стиля, заданного в свойстве `EvenStyle`. Такое поведение определяется в свойстве `EvenStylePriority` - по умолчанию оно равно `UseFill`. Если вам нужны остальные параметры стиля, установите это свойство в `UseAll`.

Готовый отчет, который использует эту технику, может выглядеть так:



Product name	Unit price
Chai	18,00р.
Chang	19,00р.
Chartreuse verte	18,00р.
Côte de Blaye	263,50р.
Guaraná Fantástica	4,50р.
Ipoh Coffee	46,00р.

Отчет с одним бэндом "Данные"

Отчет этого типа является наиболее часто востребованным. Он позволяет напечатать список строк из источника данных. Например, это может быть список клиентов.

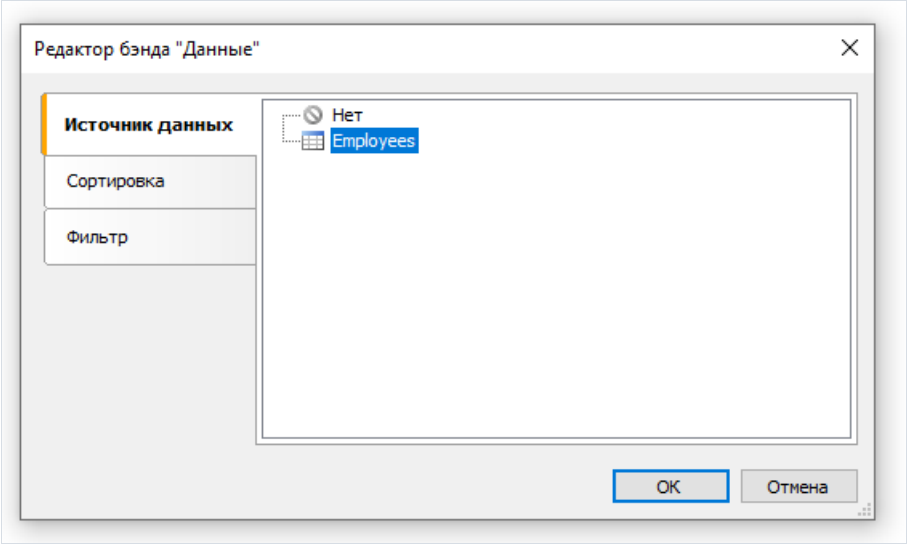
Подключение бэнда к данным

Для печати данных в отчете в виде списка вам понадобится бэнд "Данные", который надо подключить к источнику данных. Бэнд будет напечатан столько раз, сколько строк в источнике.

> Если бэнд "Данные" не подключен к источнику, он печатается один раз.

Когда вы создаете новый отчет, он уже содержит несколько пустых бэндов, в том числе и бэнд "Данные". Этот бэнд также можно добавить в окне "Настройка бэндов", выбрав пункт меню "Отчет|Настроить бэнды...".

Для того чтобы подключить бэнд к данным, сделайте на нем двойной щелчок мышью. В окне редактора выберите источник данных и нажмите кнопку "OK":



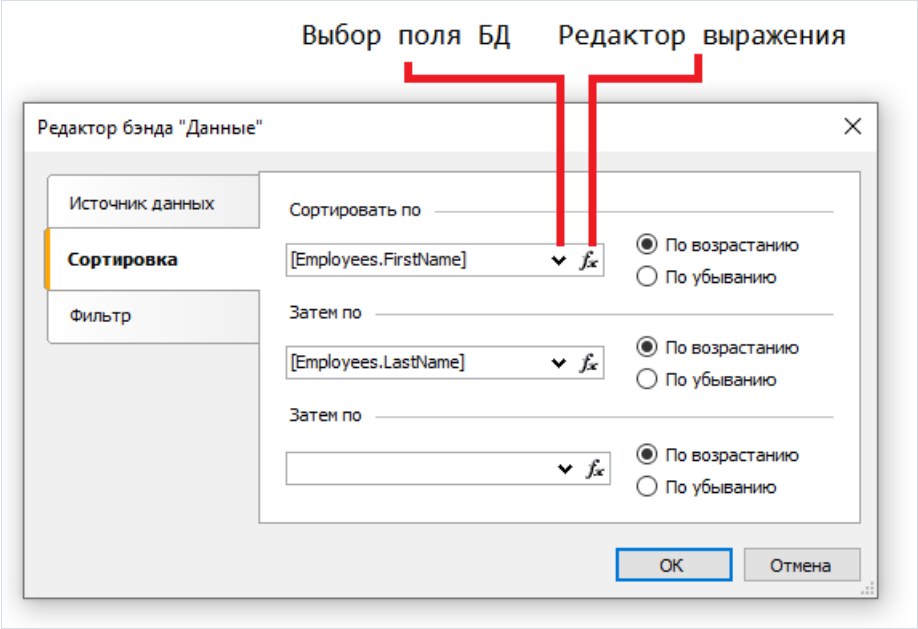
Печать текстовых данных

После того, как бэнд подключен к данным, на нем можно разместить объекты "Текст", которые будут показывать информацию из полей данных. Самый быстрый способ сделать это - перетащить (drag&drop) поля из окна "Данные" на бэнд. Подробнее о всех возможностях объекта "Текст" читайте в главе "Объект "Текст".

Сортировка данных

Бэнд "Данные" печатает данные в том порядке, в каком они расположены в источнике данных. Часто бывает необходимо отсортировать данные перед печатью. Например, список клиентов удобно представить в отсортированном по алфавиту виде.

Вы можете управлять сортировкой в редакторе бэнда "Данные". Чтобы вызвать редактор, сделайте двойной щелчок на свободном месте бэнда:

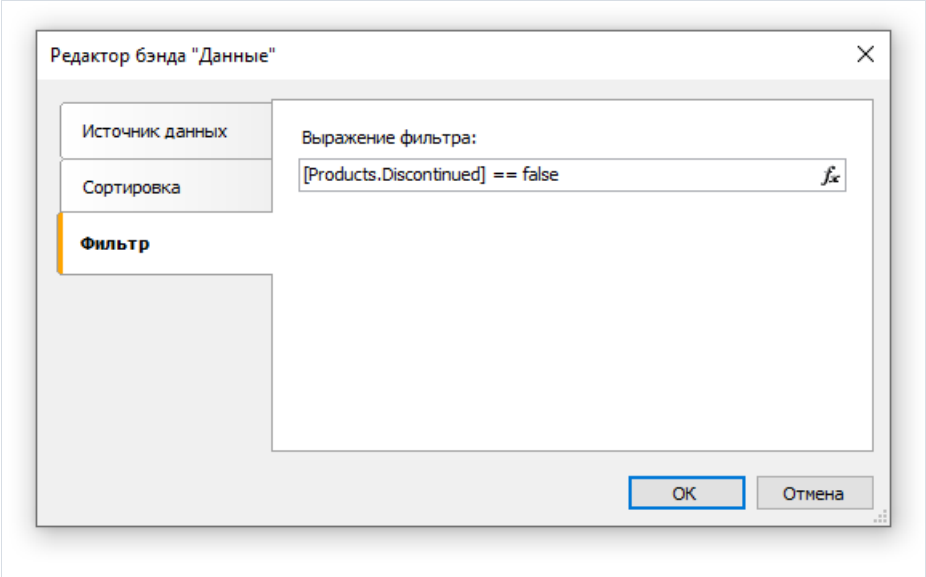


В качестве критерия сортировки можно использовать поле данных или выражение. Вы можете указать несколько (не более трех) условий для сортировки. Это может понадобиться, к примеру, если вы хотите сортировать список клиентов по городам, затем по названию клиента. Для каждого условия можно выбрать порядок сортировки - по возрастанию или по убыванию.

Другой способ указать сортировку - использовать в качестве источника данных запрос на языке SQL, в котором указать нужный критерий сортировки. Запрос будет выполнен на сервере данных и вернет отсортированные строки.

Фильтрация данных

Чтобы отфильтровать строки, которые печатаются в бэнде "Данные", вызовите его редактор и переключитесь на закладку "Фильтр":



В качестве выражения фильтра можно указать любое корректное с точки зрения FastReport выражение. Подробнее о выражениях читайте в главе "Выражения".

В примере выше стоит фильтр

```
[Products.Discontinued] == false
```

Это значит, что будут выбраны все строки данных, у которых флаг `Discontinued` равен `false`.

Другой пример фильтра со сложным выражением:

```
[Products.Discontinued] == false && [Products.UnitPrice] < 10
```

Это значит, что будут выбраны все строки данных, у которых флаг `Discontinued` равен `false` и цена меньше 10.

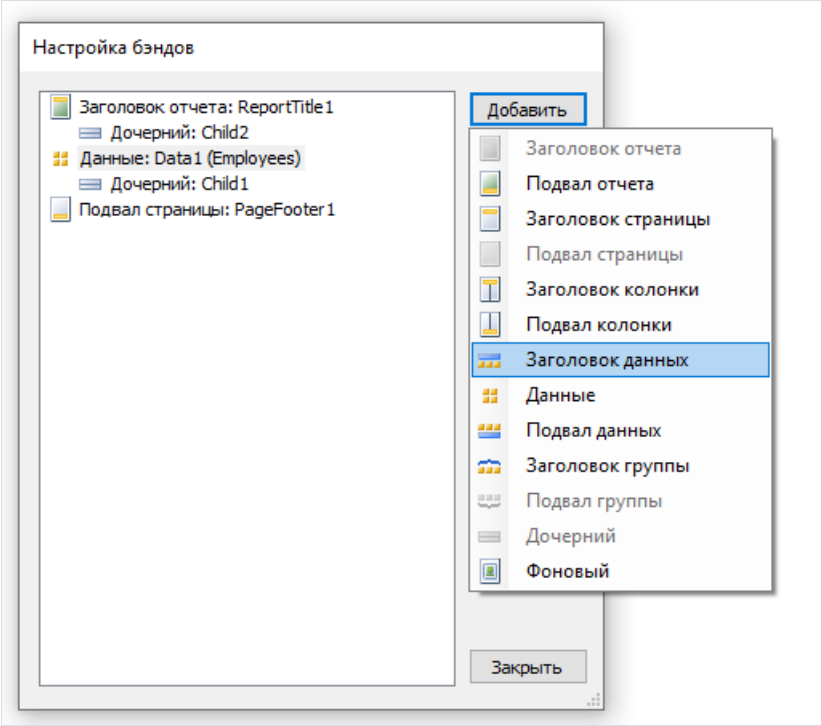
Этот способ фильтрации предполагает, что источник данных содержит все строки, часть из которых будет отфильтрована. Если источник содержит большое количество строк, это может серьезно замедлить работу отчета. В этом случае вы можете использовать в качестве источника данных запрос на языке SQL, в котором указать нужное условие фильтрации. Запрос будет выполнен на сервере данных и вернет только те строки, которые нужны в отчете.

> Для фильтрации данных также можно использовать диалоговые формы в отчете. Подробнее см. в главе "Диалоговые формы".

Заголовок и подвал данных

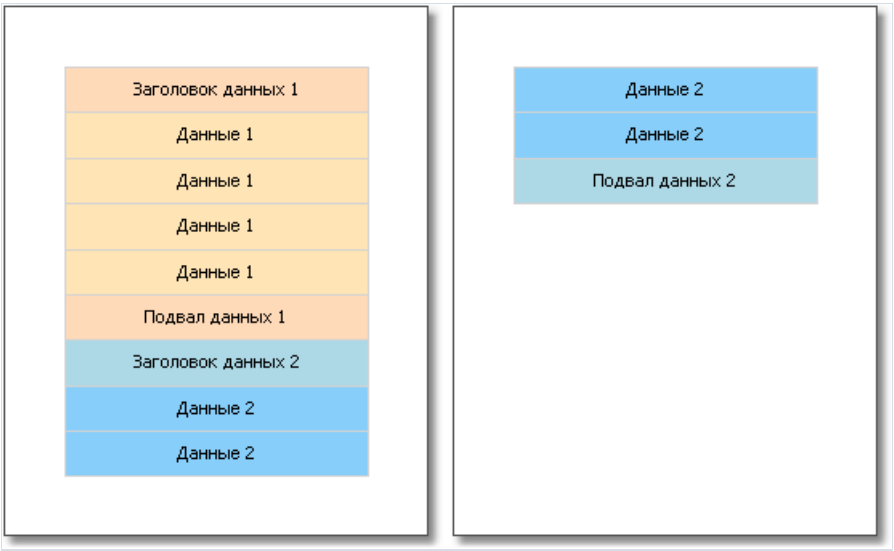
Бэнд "Данные" может иметь заголовок и подвал. Заголовок будет напечатан один раз перед данными; подвал будет напечатан после вывода всех данных.

Чтобы добавить к бэнду "Данные" заголовок и подвал, выберите пункт меню "Отчет|Настроить бэнды". В окне выделите нужный бэнд "Данные" и нажмите правую кнопку мыши. В открывшемся списке выберите пункты "Заголовок данных" и "Подвал данных":



В отчетах типа "Простой список" с одним бэндом "Данные" заголовок и подвал данных использовать не обязательно - их функции дублируются бэндами "Заголовок отчета" и "Подвал отчета". Эти бэнды могут быть полезными в следующих случаях:

- при печати нескольких списков на одной странице (см. далее). Каждый из бэндов в этом случае может иметь собственный заголовок и подвал:



- при печати одного списка, если список не помещается на одной странице готового отчета. Используя

свойство "Повторять на каждой странице" у заголовка/подвала данных, можно напечатать эти бэнды на каждой странице отчета:

Разрыв и удерживание данных

В этом разделе будут рассмотрены два режима печати бэндов - "Разрыв" и "Удерживание".

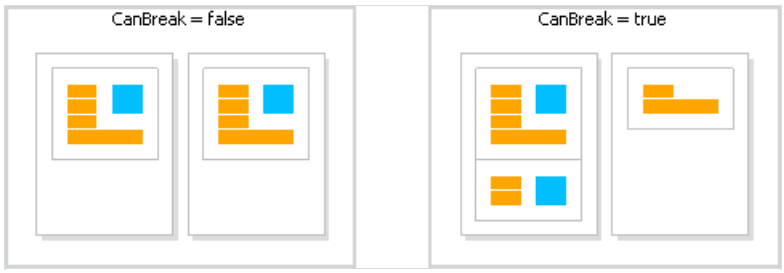
В обычном режиме печати бэнда FastReport проверяет, достаточно ли места на текущей странице. Если места не достаточно, бэнд печатается целиком на следующей странице. Если включить свойство "Может разрываться" у бэнда, FastReport будет пытаться напечатать часть содержимого бэнда на имеющемся месте, т.е. "разорвать" его.

Попытка разрыва бэнда может быть успешной, а может, и нет. Это зависит от того, какие объекты расположены на бэнде, и их настроек. Разрываться могут следующие объекты:

- "Текст";
- "Форматированный текст";
- "Таблица".

Эти объекты также имеют свойство "Может разрываться". Если оно включено, объект может быть разорван. Неразрываемый объект всегда выводится целиком там, где ему хватает места.

На рисунке ниже показан пример того, как может быть разорван бэнд.



> Алгоритм разрыва не всегда работает корректно. В случае, когда одновременно разрывается несколько объектов с разным размером шрифта, либо сложным обрамлением, могут возникать артефакты.

Цель разрыва данных - максимально использовать место на листе, чтобы в дальнейшем сэкономить бумагу при печати отчета. У удерживания данных цель противоположная: вывести определенную группу бэндов целиком на одном листе. В этом случае на листах отчета остается довольно много неиспользованного пространства, но зато данные печатаются в виде, более удобном для восприятия.

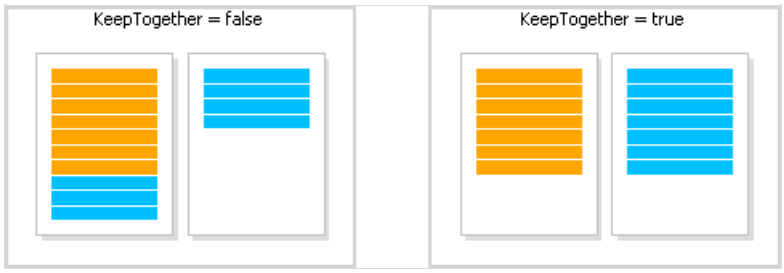
Механизм удерживания данных (`KeepTogether`) позволяет удерживать на одной странице (или колонке, если отчет с колонками) определенную группу бэндов. Если при печати удерживаемых данных достигается конец страницы, FastReport переносит все ранее напечатанные данные на новую страницу.

Вы можете использовать удерживание данных в следующих случаях:

- печать всех строк бэнда "Данные" вместе;
- печать всех элементов группы (заголовок, данные, подвал) вместе;
- печать строки главного источника данных вместе со всеми строками из подчиненного источника в отчетах типа master-detail;
- печать заголовка отчета или заголовка данных вместе с хотя бы одной строкой данных;
- печать подвала отчета или подвала данных вместе с хотя бы одной строкой данных;
- печать вместе основного и дочернего бэнда.

Рассмотрим применение удерживания данных.

Для удерживания вместе всех строк данных или группы (заголовок, данные, подвал) включите свойство "Держать вместе" (`KeepTogether`). Это свойство используется в бэндах "Данные" и "Заголовок группы". На рисунке показано, как печатаются данные без удерживания и с удерживанием:

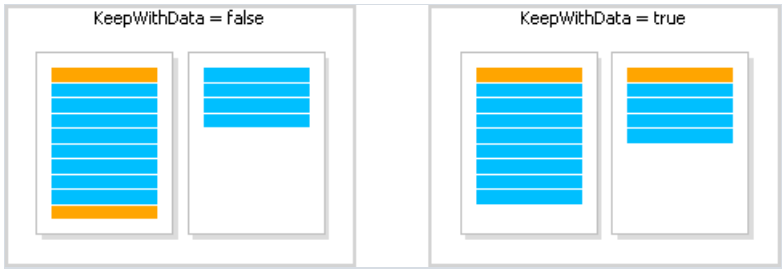


Для удерживания строки данных главного источника вместе со строками подчиненного источника включите свойство "Держать детали вместе" (`KeepDetail`) у бэнда "Данные". Это свойство используется в отчете типа master-detail:

Для запрета "висячих" заголовков и подвалов используйте свойство "Не отрывать от данных" (`KeepWithData`). Это свойство имеется у следующих бэндов:

- заголовок отчета;
- подвал отчета;
- заголовок данных;
- подвал данных;
- заголовок группы;
- подвал группы.

Оно позволяет не отрывать заголовок/подвал от строки данных. То есть, вместе с заголовком на странице должна помещаться хотя бы одна строка данных:

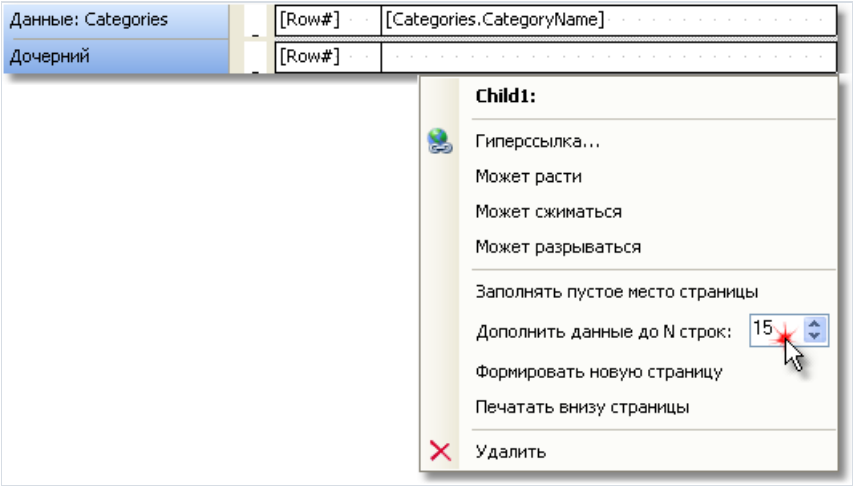


Для того чтобы держать вместе бэнд и его дочерний бэнд, включите свойство "Держать дочерний бэнд вместе" (`KeepChild`).

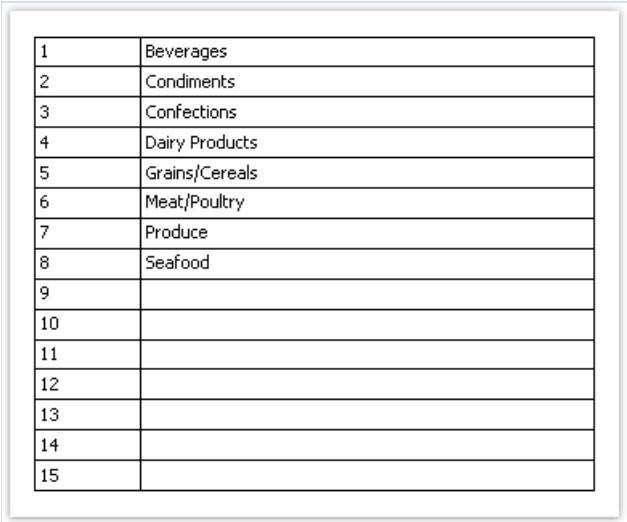
Печать пустых строк данных

Довольно часто при печати на бланках требуется напечатать определенное количество строк данных. Если реальных данных меньше, чем требуемое количество строк, надо напечатать пустые строки. Это можно сделать с помощью бэнда "Дочерний", прикрепив его к бэнду "Данные".

У бэнда "Дочерний" имеется свойство "Дополнить данные до N строк" (`CompleteToNRows`). Если в этом свойстве указать значение > 0, бэнд будет использован для дополнения строк данных до указанного количества. Допустим, требуется напечатать 15 строк данных, тогда как в источнике данных имеется лишь 8. В этом случае бэнд "Дочерний" будет напечатан 7 раз.



Готовый отчет будет выглядеть так:



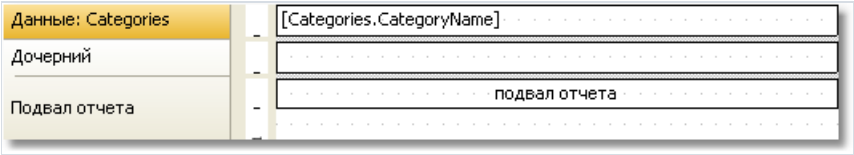
1	Beverages
2	Condiments
3	Confections
4	Dairy Products
5	Grains/Cereals
6	Meat/Poultry
7	Produce
8	Seafood
9	
10	
11	
12	
13	
14	
15	

> Если данных в источнике больше, чем указано в свойстве "Дополнить данные до N строк", пустые строки печататься не будут.

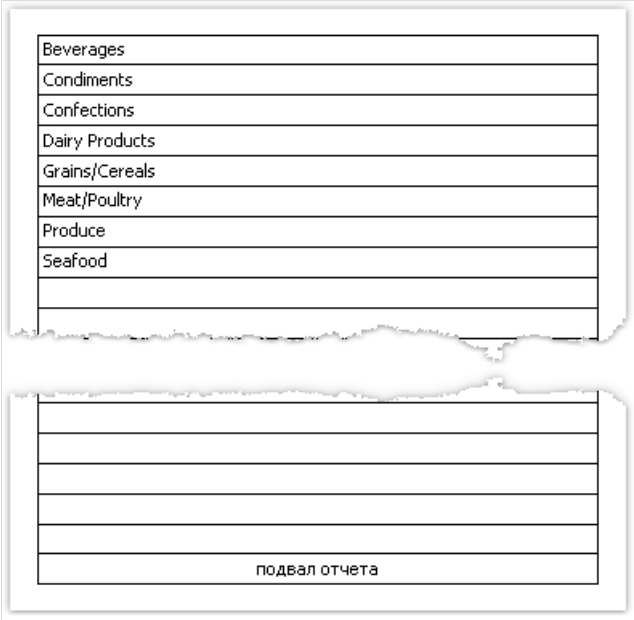
Второй способ печати пустых строк позволяет заполнить пустыми строками свободное место страницы. В этом случае бэнд "Дочерний" прикрепляется к бэнду типа "Подвал данных" или "Подвал группы" и заполняет свободное место страницы таким образом, чтобы на ней поместился бэнд-подвал.

Для того чтобы напечатать пустые строки указанным способом, прикрепите бэнд "Дочерний" к бэнду-подвалу и включите его свойство "Заполнять пустое место страницы". В дизайнере дочерний бэнд, у которого включено это свойство, будет отображаться выше бэнда, к которому он подключен. На рисунке

ниже бэнд "Дочерний" прикреплен к бэнду "Подвал отчета":



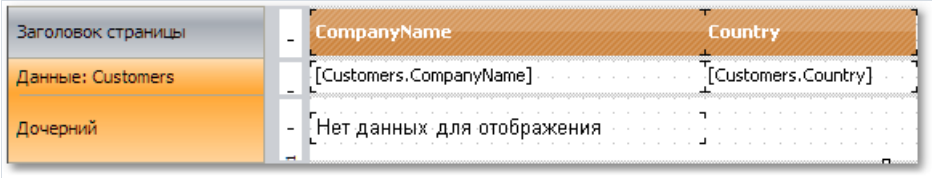
Если запустить такой отчет, мы увидим следующее:



Печать текста "Нет данных"

Если бэнд "Данные" подключен к пустому источнику данных, то он напечатан не будет. В таком случае иногда требуется печатать надпись типа "Нет данных". Для этого сделайте следующее:

- добавьте к бэнду "Данные" дочерний бэнд;
- у дочернего бэнда включите свойство `PrintIfDatabandEmpty` (это можно сделать в окне "Свойства");
- на дочерний бэнд положите объект "Текст" с текстом "Нет данных для отображения".



Отчет будет построен следующим образом:

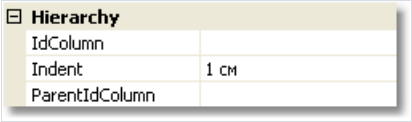
- если в источнике есть данные, будет напечатан бэнд "Данные" со всеми сопутствующими бэндами (заголовок данных, подвал);
- если источник данных пустой, будет напечатан только дочерний бэнд с текстом "Нет данных для отображения".

Печать иерархии

Бэнд "Данные" позволяет печатать иерархический список. Для этого используется один бэнд и один источник данных. Иерархия должна быть определена в источнике данных с помощью двух полей:

1. ключевое поле - это идентификатор строки данных;
2. поле, которое содержит ключ родительской строки данных.

Чтобы напечатать такой источник в иерархическом виде, вам нужно настроить следующие свойства бэнда "Данные". Это можно сделать в окне "Свойства":

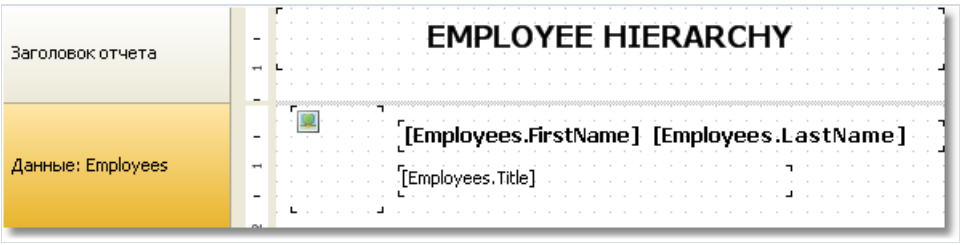


- ключевое поле указываем в свойстве `IdColumn`;
- поле, содержащее значение родителя, указываем в свойстве `ParentIdColumn`;
- в свойстве `Indent` задаем величину отступа.

Покажем на примере, как распечатать иерархию сотрудников из таблицы Employees. Таблица имеет два необходимых нам поля:

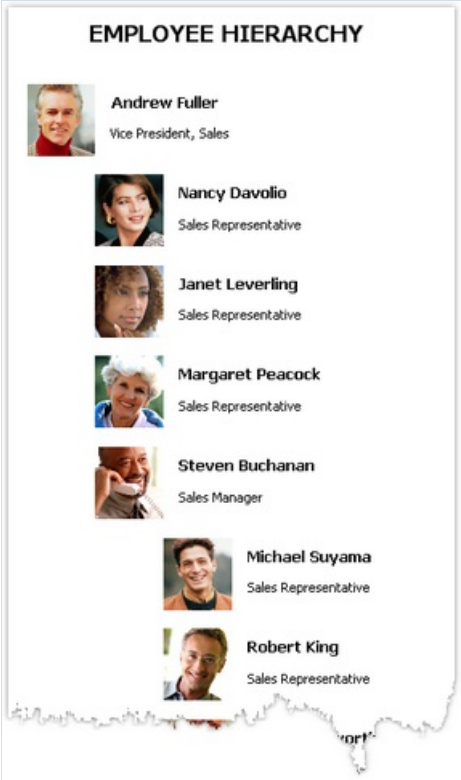
- поле `EmployeeID` является ключевым и содержит номер сотрудника;
- поле `ReportsTo` содержит номер сотрудника, которому подчиняется данный сотрудник.

Создадим отчет следующего вида:



Настроим свойства бэнда "Данные", которые отвечают за иерархию, следующим образом:

Запустим отчет на выполнение. В результате получится следующее:

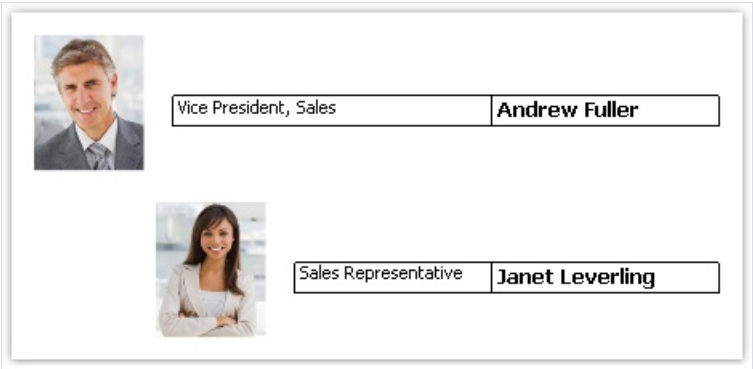


При печати иерархии FastReport сдвигает бэнд вправо (на величину, указанную в свойстве `Indent`), а также уменьшает ширину бэнда на это же значение. При этом вы можете использовать свойство `Anchor` у объектов, лежащих на бэнде.

Возможные комбинации значения `Anchor`:

- `Left, Top` (по умолчанию) - объект сдвигается вместе с бэндом;
- `Right, Top` - объект остается на месте;
- `Left, Right, Top` - фиксируется правый край объекта, левый край сдвигается вместе с бэндом.

Это позволяет получать разнообразные эффекты, как показано на рисунке:



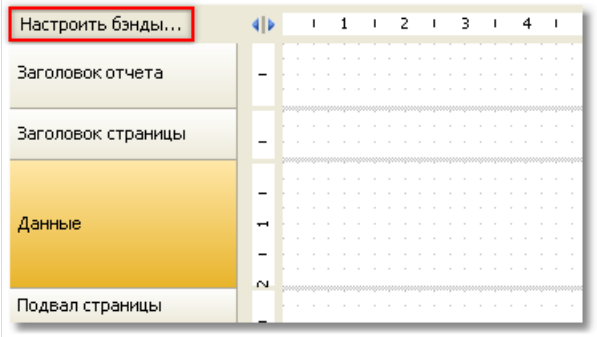
В данном примере у картинки `Anchor = Left, Top`; у объекта с названием должности `Anchor = Left, Right, Top`; у объекта с именем - `Anchor = Right, Top`.

Отчет "Главный-подчиненный" (Master-detail)

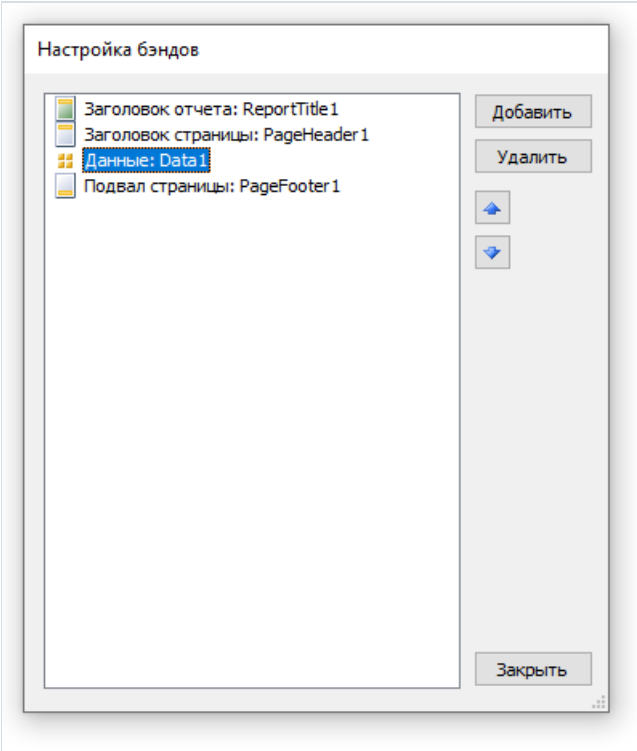
Используя два бэнда "Данные", легко построить отчет типа "Главный-подчиненный" (master-detail). В этом отчете используются два источника данных, между которыми установлена связь. Одной строке главного источника может соответствовать несколько строк подчиненного источника. Подробнее о связях читайте в главе "Данные".

Бэнды необходимо расположить в отчете так, чтобы главный бэнд содержал внутри себя подчиненный. Это делается в окне "Настройка бэндов", которое можно вызвать в меню "Отчет|Настроить бэнды...".

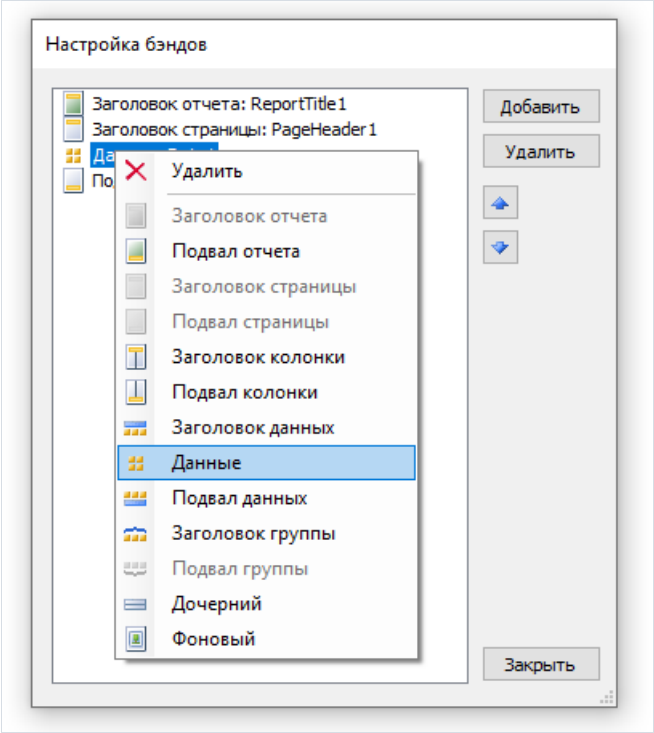
Рассмотрим создание master-detail отчета "с нуля". Для этого запустим дизайнер отчета и создадим новый пустой отчет. Он уже содержит один бэнд "Данные":



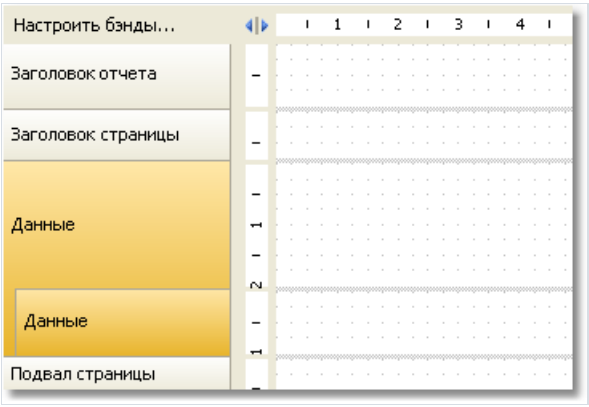
Чтобы добавить подчиненный бэнд "Данные", вызовите окно "Настройка бэндов". Это можно сделать, нажав кнопку "Настроить бэнды...", показанную на рисунке, или выбрав пункт меню "Отчет|Настроить бэнды...". В окне настройки отображается текущая структура бэндов:



Выберите бэнд "Данные", как показано на рисунке, и нажмите правую кнопку мыши, чтобы показать контекстное меню (либо нажмите кнопку "Добавить" в нижней части окна). В открывшемся меню выберите бэнд "Данные":

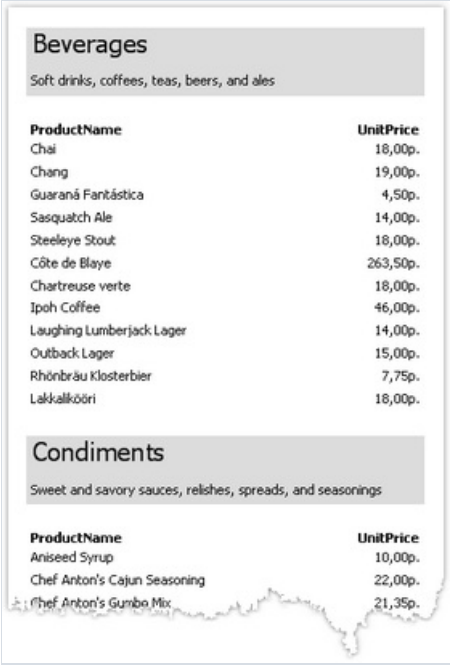


После этого к выбранному бэнду добавится вложенный бэнд "Данные". Закройте окно кнопкой "Закрыть". Вы увидите, что шаблон отчета изменился следующим образом:



Вложенность бэндов данных хорошо видна на структуре бэндов в левой части окна. После этого нужно привязать бэнды к соответствующим источникам данных и расположить поля данных на бэндах. Мы будем использовать связанные источники данных Categories (Категории) и Products (Продукты) из демонстрационной базы данных, которая поставляется в комплекте FastReport:

При запуске отчета мы увидим следующее:

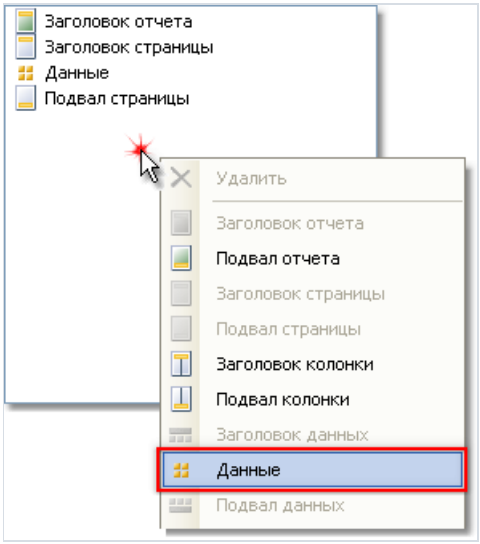


Таким образом вы можете создавать отчеты типа master-detail с неограниченной вложенностью данных, например, master-detail-subdetail. Другой способ, который применяется для создания отчетов типа master-detail, связан с использованием вложенных отчетов. Вложенные отчеты будут рассмотрены далее.

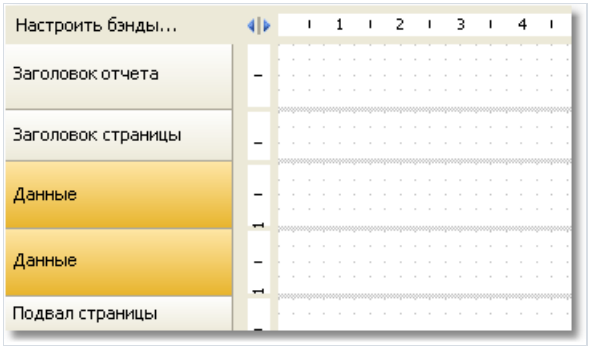
Отчет типа master-master

На странице отчета можно напечатать несколько простых списков. Это можно сделать, разместив на странице два или несколько бэндов "Данные". В отличие от отчета master-detail, где бэнды являются вложенными друг в друга и печатают данные из связанных источников, в отчете этого типа (назовем его master-master) и бэнды, и источники данных независимы друг от друга.

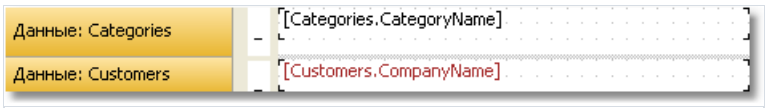
Покажем на примере, как создать отчет, который печатает на странице два списка - список категорий (таблица Categories) и клиентов (таблица Customers). Создайте новый отчет и добавьте в него нужные источники данных. Чтобы добавить второй бэнд "Данные", вызовите окно "Настройка бэндов":



Щелкните правой кнопкой мышки на пустом месте списка, как показано на рисунке, и выберите в контекстном меню бэнд "Данные". Это создаст второй независимый бэнд "Данные". Шаблон отчета будет выглядеть так:



Теперь подключим бэнды к источникам данных и разместим на них несколько полей данных:

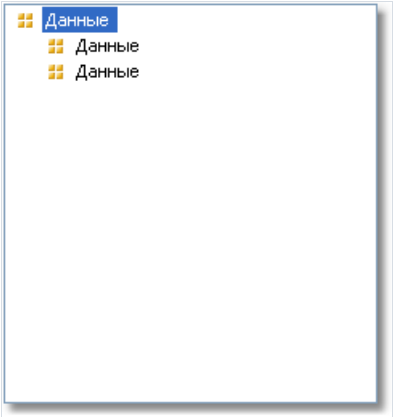


Если запустить отчет, мы увидим следующее:

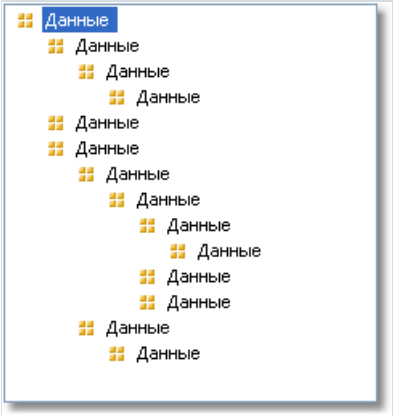

Beverages
Condiments
Confections
Dairy Products
Grains/Cereals
Meat/Poultry
Produce
Seafood
Alfreds Futterkiste
Ana Trujillo Emparedados y helados
Antonio Moreno Taquería
Around the Horn

Отчет типа master-detail-detail

Бэнд "Данные" может содержать один или несколько вложенных бэндов "Данные". Это позволяет строить отчеты типа master-detail-detail. Настроить структуру бэндов для такого отчета можно в окне "Настройка бэндов". Для этого щелкните правой кнопкой мыши на главном бэнде "Данные" и добавьте к нему подчиненный бэнд "Данные". Повторите эту процедуру для добавления второго подчиненного бэнда:



Таким образом, можно добавлять неограниченное количество подчиненных бэндов в главный бэнд "Данные". Например, структура отчета может выглядеть так (это вымышленная структура, она лишь дает представление о возможностях FastReport):



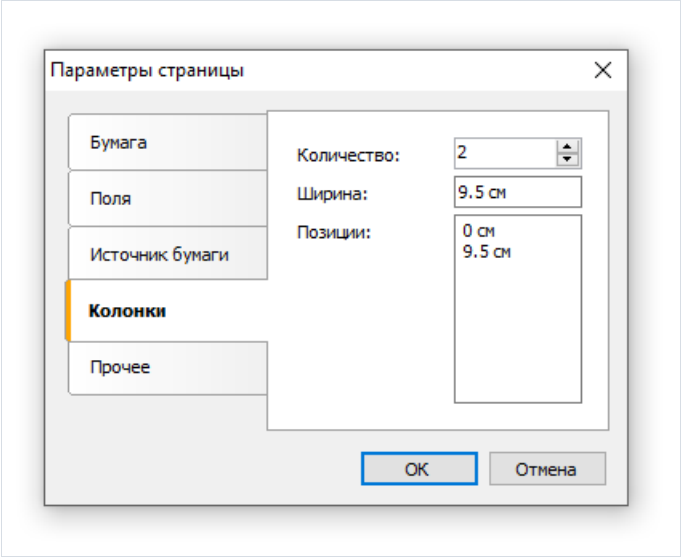
Многоколоночный отчет

Обычный отчет печатает данные до тех пор, пока не достигнут конец страницы. После этого формируется новая страница, и печать продолжается на ней. Отчет с колонками печатает данные в несколько колонок. Когда достигнут конец страницы, печать продолжается с новой колонки на этой же странице. В этом смысле обычный отчет можно рассматривать как отчет с одной колонкой.

В FastReport есть два способа печати колонок.

Колонки страницы

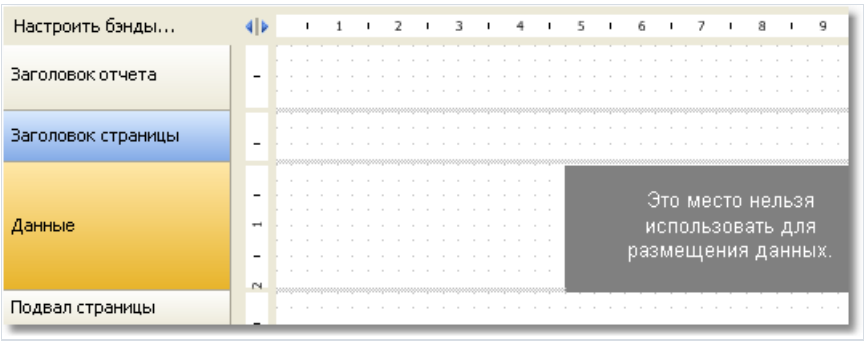
Первый способ заключается в указании количества колонок у страницы отчета. Это делается в настройках страницы на закладке "Колонки":



Как видно, вы можете указать следующие параметры колонок:

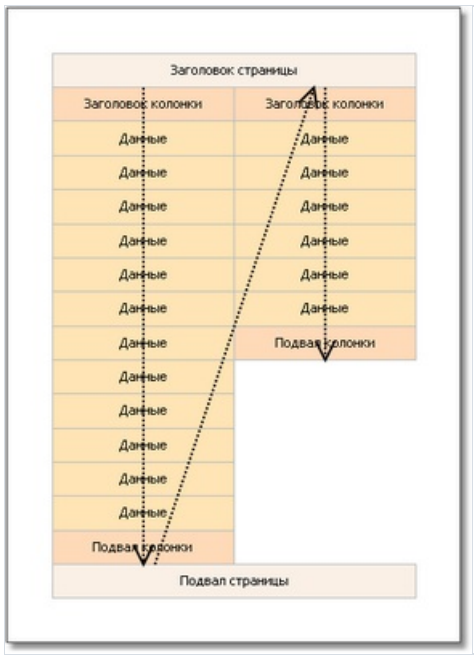
- количество;
- ширина колонки;
- позиция каждой колонки.

Для того чтобы превратить обычный отчет в отчет с колонками, вам нужно указать только количество колонок на странице. Все остальные параметры FastReport подберет самостоятельно. Когда вы включите колонки, вид бэндов в дизайнере изменится:



Такие бэнды, как "Данные", "Заголовок группы" и другие, связанные с выводом данных, используют для печати только область, соответствующую ширине колонки. Эта область показана, как обычно, белым цветом. Область, показанную серым, для размещения данных использовать нельзя: здесь будут печататься данные в соседних колонках.

Для работы с колонками используются бэнды - "Заголовок колонки" и "Подвал колонки". Как видно из их названий, они печатаются вверху и внизу каждой колонки соответственно. Схема печати отчета с колонками представлена на рисунке:


Заголовок страницы
Заголовок колонки
Заголовок колонки
Данные
Данные
Данные
Данные
Данные
Данные
Данные
Данные
Данные
Данные
Данные
Данные
Данные
Данные
Данные
Данные
Данные
Данные
Данные
Подвал колонки
Подвал колонки
Подвал страницы

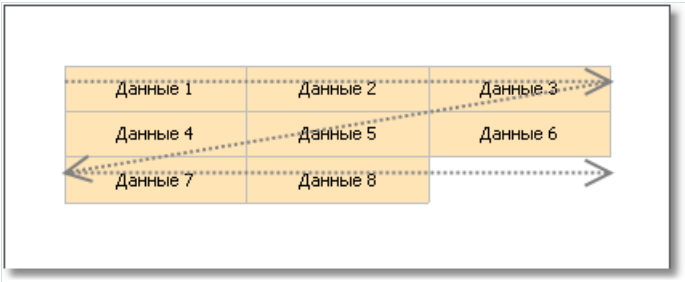
Колонки бэнда "Данные"

Второй способ многоколоночной печати заключается в использовании колонок бэнда "Данные". При этом остальные бэнды продолжают печататься в одну колонку.

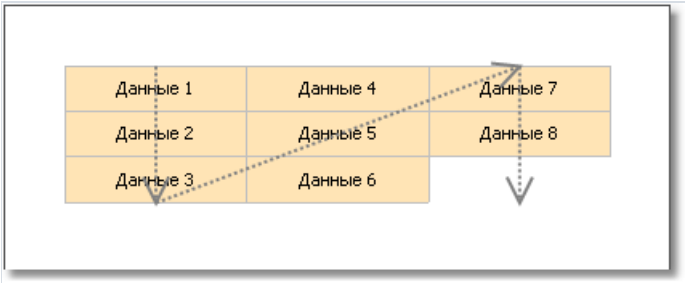
Параметры колонок настраиваются с помощью свойства `Columns`, которое можно изменить в окне "Свойства". Вы можете задать следующие параметры:

- количество колонок;
- ширина колонки;
- режим печати колонок. Вы можете выбрать один из двух режимов - "слева направо, сверху вниз" и "сверху вниз, слева направо";
- минимальное количество строк в одной колонке, если выбран режим печати "сверху вниз, слева направо".

Колонки бэнда могут печататься в одном из двух режимов. В режиме "слева направо, сверху вниз" (режим по умолчанию) колонки печатаются следующим образом:



В режиме "сверху вниз, слева направо" печать колонок происходит следующим образом:



В этом режиме FastReport подбирает количество строк данных в колонке таким образом, чтобы колонки заполнялись равномерно. Вы также можете указать минимальное количество строк в колонке с помощью свойства `Columns.MinRowCount`.

> Бэнд, у которого есть колонки, не должен иметь подчиненных бэндов. Это касается отчетов типа master-detail. Не устанавливайте колонки у бэнда master.

Многостраничный отчет типа "Буклет"

При печати отчета в виде буклета вы, как правило, столкнетесь со следующими требованиями:

- отдельные страницы отчета - титульная страница (cover), оглавление (table of contents), содержимое отчета, последняя страница (back cover);
- разные поля страницы для четных и нечетных страниц;
- разное оформление заголовков и подвалов на четных и нечетных страницах.

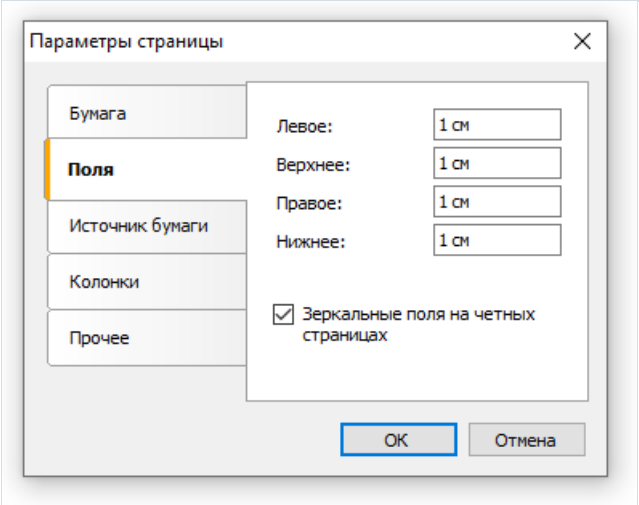
Добавление страницы в отчет

Вы можете добавлять в шаблон отчета любое количество страниц. На каждой странице можно размещать отдельный отчет. Для добавления новой страницы нажмите кнопку  на панели инструментов. Страницу можно также добавить, нажав кнопку "Создать..." и выбрав в окне пункт "Страница отчета".

> Для печати раздела "Оглавление" вы можете воспользоваться приемом, описанным в главе "Интерактивные отчеты".

Настройка страниц

В настройках страницы на закладке "Поля" можно указать, что для четных страниц левое и правое поля поменяются местами:



Если необходимо, чтобы страница начиналась с нечетного номера, установите свойство страницы `StartOnOddPage` в `true`. При необходимости FastReport вставит пустую страницу перед началом печати указанной страницы.

Печать на четных и нечетных страницах

Используя свойство "Печатать на..." (`PrintOn`) объектов отчета, можно печатать разные объекты на четных и нечетных страницах.

> Свойство `PrintOn` можно поменять в окне "Свойства", предварительно выделив объект.

Это свойство определяет, на каких страницах может быть напечатан объект. Доступны следующие значения, а также любая их комбинация:

- первая страница (`FirstPage`);
- последняя страница (`LastPage`) - должен быть включен двойной проход у отчета;
- нечетные страницы (`OddPages`);
- четные страницы (`EvenPages`);
- повторение печати бэнда (`RepeatedBand`). Возникает при печати бэнда, если у него установлен флаг "Повторять на каждой странице";
- единственная страница (`SinglePage`) - должен быть включен двойной проход у отчета.

По умолчанию значение этого свойства равно `FirstPage, LastPage, OddPages, EvenPages, RepeatedBand, SinglePage`. Это означает, что объект будет напечатан на всех страницах отчета. В случае, если отчет состоит из единственной страницы, печать объекта определяется только наличием флага `SinglePage`.

Приведем несколько типичных примеров использования этого свойства:

Значение свойства	Где будет напечатан объект
FirstPage	Только на первой странице.
LastPage, OddPages, EvenPages, RepeatedBand	На всех страницах, кроме первой.
FirstPage, OddPages, EvenPages, RepeatedBand	На всех страницах, кроме последней.
RepeatedBand	Только на бэндах, которые повторяются на каждой странице.
FirstPage, LastPage, OddPages, EvenPages	На всех бэндах, кроме тех, что повторяются на каждой странице.
FirstPage, LastPage, OddPages, RepeatedBand	Только на нечетных страницах.
FirstPage, LastPage, EvenPages, RepeatedBand	Только на четных страницах.

Примечание: Четность и нечетность страниц определяется номером страницы в подготовленном отчете, отсчет страниц в котором начинается с 0. Например, вторая страница, отображаемая в предварительном просмотре, будет нечетной так как ее номер в подготовленном отчете равен 1.

Например, чтобы напечатать разный текст на четных и нечетных страницах, положите на бэнд два объекта "Текст" и настройте их следующим образом:

- первый объект будет печататься на нечетных страницах. Установите его свойство `PrintOn = FirstPage, LastPage, OddPages` (т.е. все значения, кроме `EvenPages`).
- второй объект будет печататься на четных страницах. Установите его свойство

`PrintOn = FirstPage, LastPage, EvenPages` (т.е. все значения, кроме `OddPages`).

Объекты можно расположить друг над другом или иным образом. Они никогда не будут печататься одновременно.

Аналогичное свойство есть и у бэндов. Чтобы создать два экземпляра бэнда, которые печатаются на разных страницах, используйте бэнд "Дочерний". Его можно прикрепить к любому бэнду в окне "Настройка бэндов". Настройте главный и дочерний бэнды следующим образом:

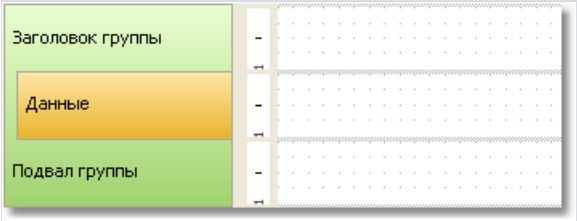
- главный бэнд будет печататься на нечетных страницах. Установите его свойство `PrintOn = FirstPage, LastPage, OddPages` (т.е. все значения, кроме `EvenPages`).
- дочерний бэнд будет печататься на четных страницах. Установите его свойство `PrintOn = FirstPage, LastPage, EvenPages` (т.е. все значения, кроме `OddPages`).

Бэнды могут иметь разную высоту и разное содержимое. Рассмотрим пример, печатающий разные заголовки страницы:

Группировка, итоги

Ранее мы рассматривали отчет типа "Главный-подчиненный", который печатал данные из двух связанных источников. FastReport позволяет создать отчет, который выглядит аналогичным образом, но использует только один источник данных. Для этого применяются группы.

Группа представляет собой набор из трех бэндов: "Заголовок группы", "Данные" и "Подвал группы". В дизайнере это выглядит следующим образом:



> Группа всегда имеет заголовок и данные. Подвал группы является необязательным, вы можете его удалить.

Для заголовка группы указывается условие группировки. Это может быть любое выражение, но, как правило, это одно из полей источника данных. Сам источник данных подключается к бэнду Данные". Печать группы выполняется следующим образом:

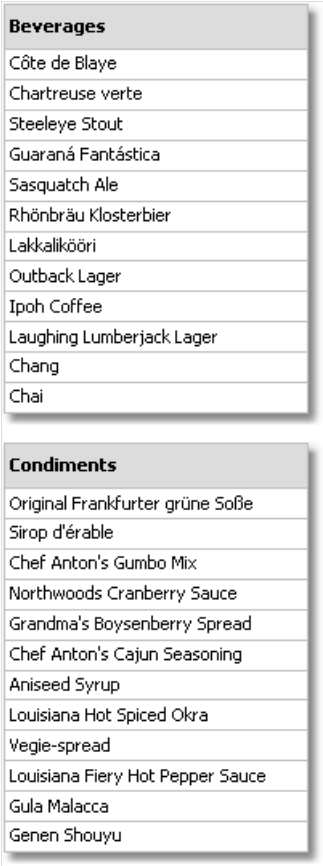
1. печатается заголовок группы;
2. печатается строка данных;
3. проверяется, не изменилось ли условие группировки;
4. если условие не изменилось, печатается очередная строка данных (п.2);
5. если условие изменилось, печатается подвал группы, и начинается печать новой группы (п.1).

Допустим, у нас есть таблица Products со следующими данными:

CategoryName	ProductName
Beverages	Côte de Blaye
Beverages	Chartreuse verte
Beverages	Steeleye Stout
Beverages	Guaraná Fantástica
Beverages	Sasquatch Ale
Beverages	Rhönbräu Klosterbier
Beverages	Lakkalikööri
Beverages	Outback Lager
Beverages	Ipoh Coffee
Beverages	Laughing Lumberjack Lager
Beverages	Chang
Beverages	Chai
Condiments	Original Frankfurter grüne Soße
Condiments	Sirop d'érable
Condiments	Chef Anton's Gumbo Mix
Condiments	Northwoods Cranberry Sauce
Condiments	Grandma's Boysenberry Spread
Condiments	Chef Anton's Cajun Seasoning
Condiments	Aniseed Syrup
Condiments	Louisiana Hot Spiced Okra
Condiments	Vegie-spread
Condiments	Louisiana Fiery Hot Pepper Sauce
Condiments	Gula Malacca
Condiments	Genen Shouyu

Данные можно сгруппировать по полю `CategoryName`. Это поле будет печататься в заголовке группы. Сами данные представлены полем `ProductName`. Отчет может выглядеть следующим образом:

Если запустить отчет, получится следующее:



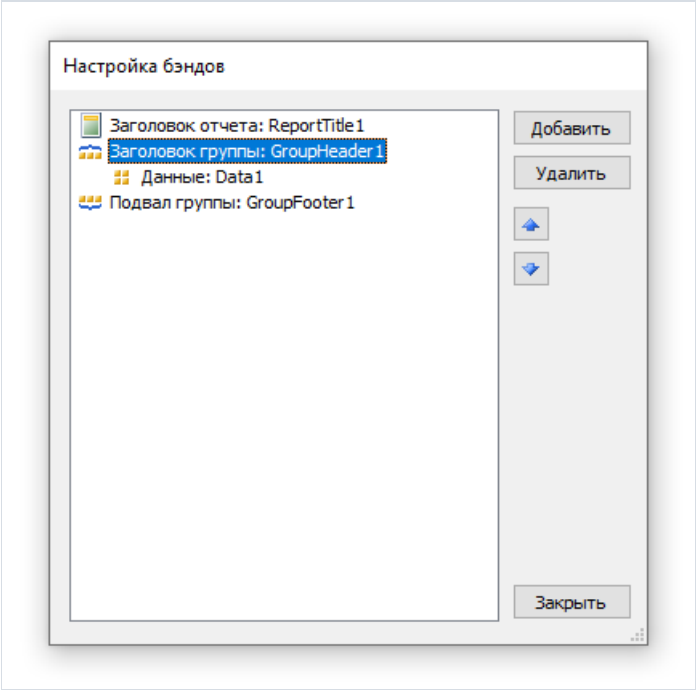
Beverages
Côte de Blaye
Chartreuse verte
Steeleye Stout
Guaraná Fantástica
Sasquatch Ale
Rhönbräu Klosterbier
Lakkalikööri
Outback Lager
Ipoh Coffee
Laughing Lumberjack Lager
Chang
Chai

Condiments
Original Frankfurter grüne Soße
Sirop d'érable
Chef Anton's Gumbo Mix
Northwoods Cranberry Sauce
Grandma's Boysenberry Spread
Chef Anton's Cajun Seasoning
Aniseed Syrup
Louisiana Hot Spiced Okra
Vegie-spread
Louisiana Fiery Hot Pepper Sauce
Gula Malacca
Genen Shouyu

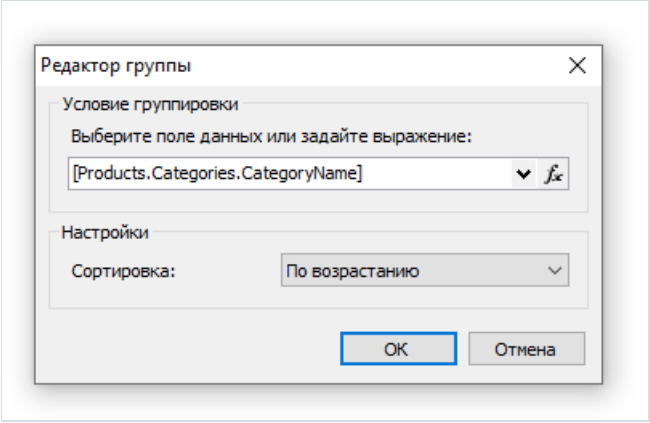
Создание группы

Добавить группу в отчет можно двумя способами.

Первый способ: вы добавляете бэнд "Заголовок группы" в окне "Настройка бэндов". Для этого в окне нажмите кнопку "Добавить" и выберите бэнд "Заголовок группы". FastReport добавит группу к имеющемуся бэнду "Данные" или создаст группу целиком, если такого бэнда в отчете нет:

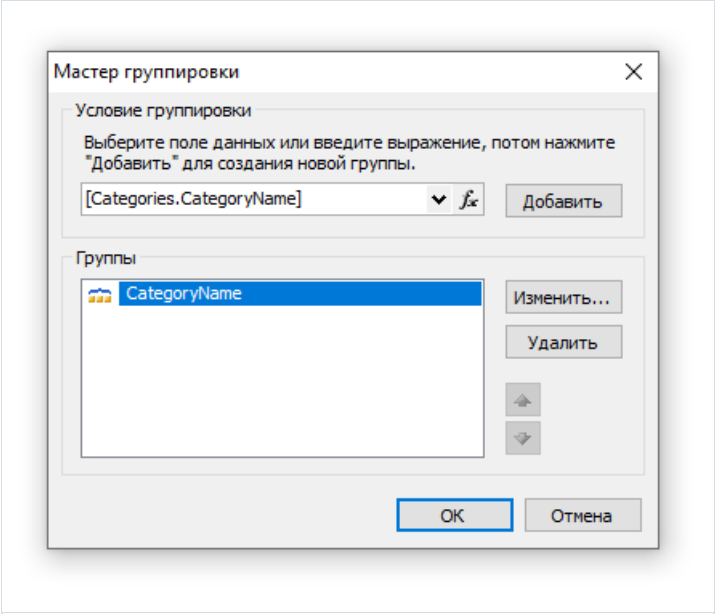


Чтобы настроить группу, сделайте двойной щелчок на бэнде "Заголовок группы". Вы увидите окно редактора группы:

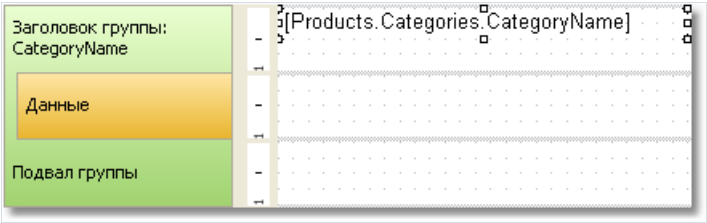


Вам необходимо указать условие группировки: это может быть любое выражение и поле источника данных. Также укажите сортировку. По умолчанию данные сортируются по возрастанию.

Второй способ: вы используете мастер, который можно вызвать из меню "Отчет", выбрав пункт "Мастер группировки...". Чтобы создать группу, введите условие группировки и нажмите кнопку "Добавить":



Мастер добавит в отчет все элементы группы и объект "Текст" на заголовке группы, в котором печатается условие группировки:

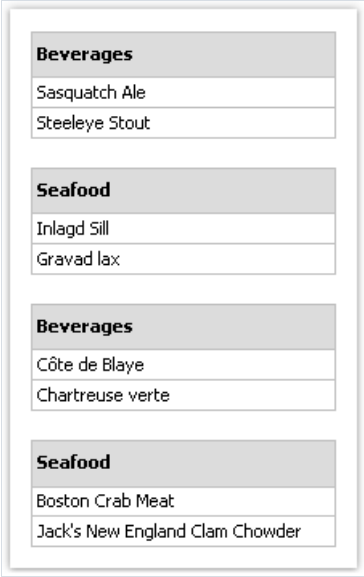


Сортировка данных

Для правильной работы группы необходимо выполнение следующего условия:

> Источник данных должен быть отсортирован по тому полю, которое используется в условии группировки.

Если это условие не выполняется, вы увидите множество одинаковых групп, содержащих по 1-2 строки данных:



К счастью, в FastReport есть возможность отсортировать источник данных двумя способами.

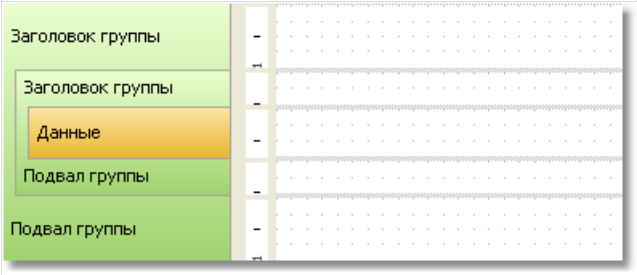
1. вы указываете порядок сортировки данных в редакторе группы. Источник данных автоматически сортируется по условию группировки;
2. вы указываете сортировку в редакторе бэнда "Данные", который входит в группу.

Оба способа равнозначны, однако удобнее использовать первый способ - при создании группы вы указываете группировку и сортировку данных в одном диалоге.

В некоторых случаях использовать первый способ нельзя. Допустим, мы задаем группировку по первой букве названия продукта. При этом продукты будут отсортированы только по первой букве, что нежелательно. В данном случае нужно воспользоваться вторым способом и указать сортировку по полному названию продукта.

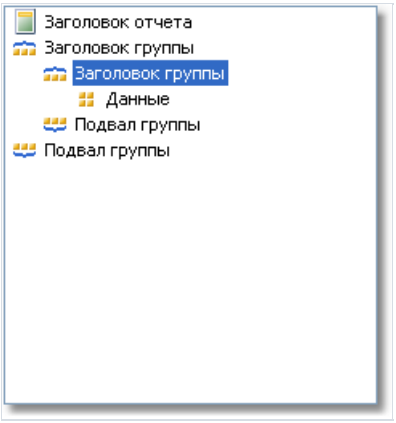
Вложенные группы

Вложенная группа имеет несколько бэндов "Заголовок группы". Последний бэнд содержит в себе бэнд "Данные":



Каждый заголовок группы имеет свое собственное условие группировки.

Создать вложенную группу можно теми же способами, что и обычную. В первом случае вы создаете простую группу, а затем добавляете вложенную группу в окне "Настройка бэндов". Для этого выделите существующий заголовок группы, нажмите кнопку "Добавить" и добавьте бэнд "Заголовок группы". Он будет добавлен к имеющейся группе:



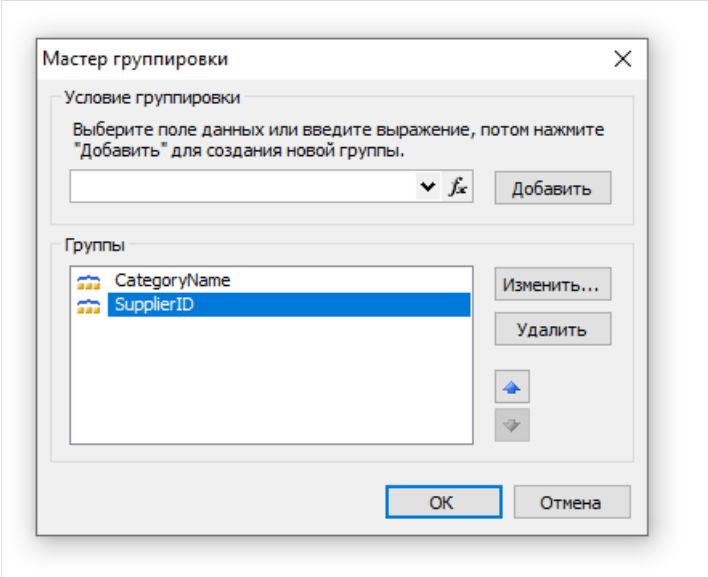
После этого вызовите редактор у добавленной группы и укажите условие группировки.

Во втором случае вы используете уже рассмотренный нами мастер группировки. Укажите условие группировки и нажмите кнопку "Добавить". Мастер добавит новую группу к уже имеющейся.

Печать вложенной группы мало чем отличается от печати обычной. При печати данных проверяются условия группировки у всех уровней группы. Если условие изменилось, соответствующая группа закрывается и начинается печать новой группы.

Управление группами

Для управления группами используйте мастер группировки, который можно вызвать из меню "Отчет|Мастер группировки...":



С помощью мастера вы можете добавить или удалить группу, а также поменять порядок группировки, используя кнопки и . С помощью кнопки "Изменить..." можно изменить условие группировки у выбранной группы.

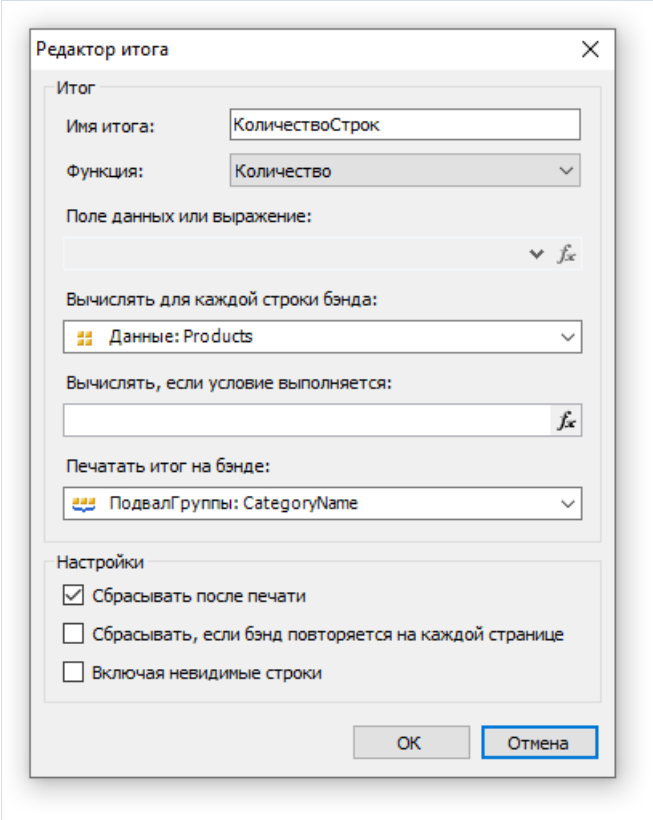
Печать итоговых значений

Группировка часто используется для печати каких-либо итоговых значений в каждой группе. Например, это может быть количество строк в группе или сумма по одному из полей данных. Для печати таких значений используются итоги. Использование итогов подробно описано в главе "Данные".

Чтобы напечатать итоговое значение в группе, вам необходимо сделать следующее:

- создайте итог, выбрав в окне "Данные" пункт "Действия|Новый итог";
- в качестве строки данных укажите бэнд "Данные", который входит в группу;
- в поле "Печатать на бэнде" укажите бэнд - подвал группы;
- положите объект "Текст", который печатает значение итога, на подвал группы.

Например, для печати количества строк в каждой группе настройка итога будет выглядеть примерно так:



Чтобы вывести значение итога, перетащите его на подвал группы:

Готовый отчет будет выглядеть следующим образом:

Meat/Poultry
Alice Mutton
Perth Pasties
Thüringer Rostbratwurst
Pâté chinois
Tourtière
Mishi Kobe Niku

6

Produce
Rössle Sauerkraut
Uncle Bob's Organic Dried Pears
Manjimup Dried Apples
Longlife Tofu
Tofu

5

Повторение заголовков и подвалов

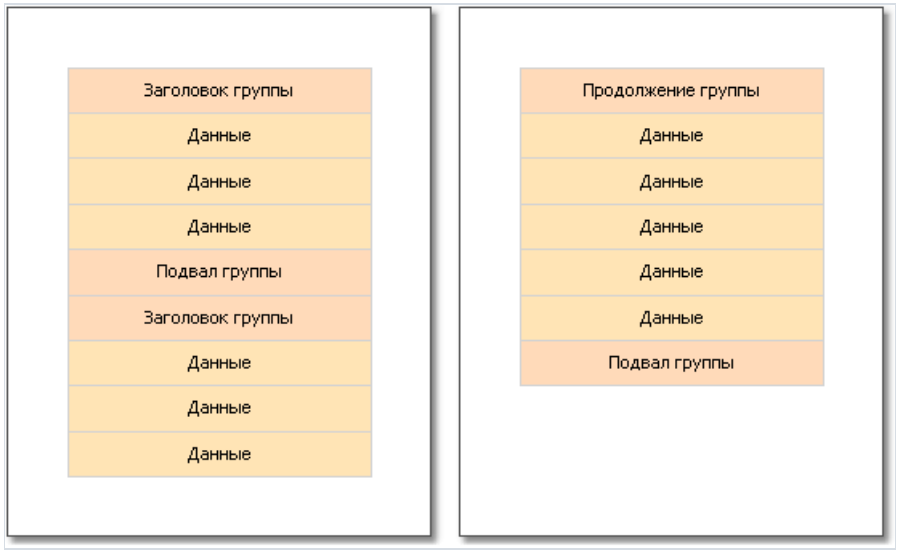
У заголовка и подвала группы имеется свойство "Повторять на каждой странице" (`RepeatOnEveryPage`). Оно может быть полезным, если в группе много строк, и они не помещаются на одной странице готового отчета. Используя это свойство, можно напечатать заголовок/подвал группы на каждой странице, которую занимает группа.

При работе свойства "Повторять..." печатается заголовок, у которого установлен флаг "Повторение" (`Repeated`). Это можно использовать для печати разных объектов на обычном заголовке группы и на повторении, например, печатать строку "Продолжение" на новой странице. Для этого используется свойство `PrintOn` объекта "Текст" (подробно оно рассмотрено в разделе "Многостраничный отчет типа "Буклет").

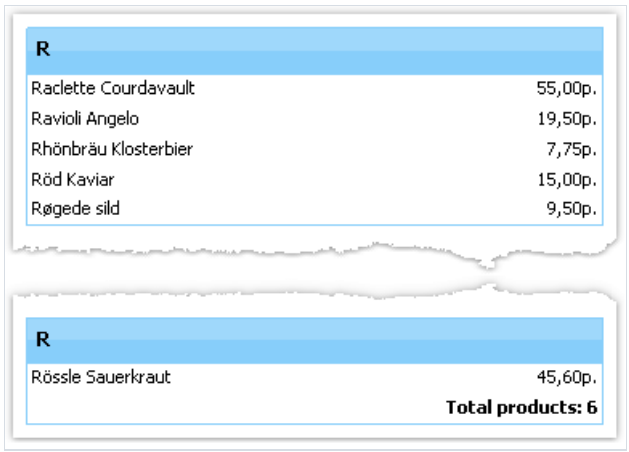
Чтобы напечатать разный текст, положите на заголовок группы два объекта, один над другим:

- первый объект будет печататься на обычных заголовках. У него надо установить значение свойства `PrintOn = FirstPage, LastPage, OddPages, EvenPages` (т.е. все доступные значения, кроме `RepeatedBand`);
- второй объект будет печататься только на повторяющихся заголовках. У него надо установить свойство `PrintOn = RepeatedBand` и добавить текст "продолжение".

В результате будет напечатан отчет следующего вида:



Подвал группы также может повторяться на каждой странице. Его можно использовать, например, для поддержания внешнего вида группы, если оформление сделано с помощью вертикальных линий:



Подвал группы содержит два объекта, расположенных друг над другом:


Total products: [TotalProducts]
PrintOn = FirstPage, LastPage, OddPages, EvenPages
PrintOn = RepeatedBand

Свойства группы

Рассмотрим свойства группы, которые имеются у бэнда "Заголовок группы".

Свойство "Формировать новую страницу" (`StartNewPage`) позволяет сформировать новую страницу перед печатью группы. Таким образом, каждая группа будет расположена на новой странице.

> Для самой первой группы новая страница сформирована не будет. Это сделано для того, чтобы избежать пустой первой страницы.

Свойство "Сбрасывать номер страницы" (`ResetPageNumber`) позволяет при печати группы сбрасывать номер страницы. Как правило, оно используется совместно со свойством "Формировать новую страницу". Таким образом, если включить оба этих свойства, каждая группа будет напечатана на новой странице, и будет иметь свою собственную нумерацию страниц.

Вложенные отчеты

Иногда в определенном месте основного отчета требуется вывести дополнительные данные, которые могут представлять собой отдельный отчет с довольно сложной структурой. Можно попробовать решить эту задачу, используя богатый набор бэндов FastReport. Однако в некоторых случаях предпочтительнее использовать объект "Вложенный отчет".

Объект "Вложенный отчет" представляет собой обычный объект отчета, который можно положить на один из бэндов. При этом FastReport добавит в отчет дополнительную страницу и свяжет ее с объектом вложенного отчета. На этой странице можно создать дополнительный отчет, имеющий любую структуру.

При печати отчета, в котором имеется объект "Вложенный отчет", будет сделано следующее:

1. будет печататься главный отчет, пока не встретится объект "Вложенный отчет";
2. будут печататься бэнды вложенного отчета;
3. будет продолжена печать основного отчета.

> Поскольку вложенный отчет формируется на листе основного отчета, он не может содержать следующих бэндов: "Заголовок/Подвал отчета", "Заголовок/Подвал/Фон страницы", "Заголовок/Подвал колонки".

Режимы печати

Вложенный отчет может печататься в двух режимах.

Первый режим печати - основной. В этом режиме бэнды и объекты вложенного отчета печатаются на странице основного отчета. Имеются ограничения на размещение объектов на бэнде, который содержит вложенный отчет:

- объект "Вложенный отчет" должен находиться у нижней границы бэнда;
- нельзя размещать другие объекты ниже объекта "Вложенный отчет". При работе отчета такие объекты будут перекрыты объектами вложенного отчета:

Для размещения других объектов ниже вложенного отчета используйте бэнд "Дочерний". Его можно прикрепить к основному бэнду и разместить объекты следующим образом:

Второй режим печати отличается тем, что вложенный отчет печатается на бэнде, который содержит объект "Вложенный отчет". Включить этот режим можно из контекстного меню объекта "Вложенный отчет", выбрав пункт "Печатать на родителе". Этот режим не накладывает ограничений на размещение объектов. Кроме того, в этом режиме родительский бэнд может расти и сжиматься в зависимости от того, сколько данных напечатано во вложенном отчете.

Единственная неприятность, связанная с этим режимом работы - данных во вложенном отчете может быть много. При его печати получится большая высота у родительского бэнда. Чтобы корректно напечатать такой бэнд, придется применить разрыв его содержимого (свойство "Может разрываться"). Алгоритм разрыва не обеспечивает 100% качества и в некоторых случаях может привести к смещению объектов.

Печать нескольких отчетов рядом (side-by-side)

Расположив рядом два объекта "Вложенный отчет", можно напечатать два независимых списка данных, которые будут располагаться рядом. При печати такого отчета FastReport действует следующим образом:

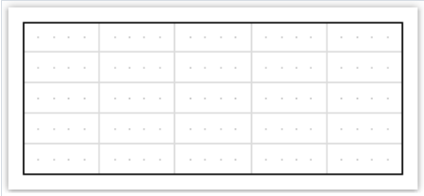
- печатается основной отчет до тех пор, пока не встретится объект "Вложенный отчет";
- печатается первый вложенный отчет;
- происходит переход на страницу, где начиналась печать вложенного отчета, и печатается следующий вложенный отчет;
- после того как напечатаны все вложенные отчеты, продолжается печать основного отчета с того места, где закончилась печать самого длинного вложенного отчета.

Несколько уровней вложенности

На страницу вложенного отчета можно поместить объект "Вложенный отчет", увеличив тем самым уровень вложенности. Количество уровней вложенности формально ничем не ограничивается, однако не следует этим увлекаться. Многократно вложенные отчеты довольно сложны для понимания. Если у вас есть возможность, используйте для печати вложенных данных бэнды FastReport. Напоминаем, что бэнд "Данные" может иметь один или несколько вложенных бэндов "Данные", и так далее. Таким образом, для печати отчетов типа master-detail или master-detail-subdetail нет необходимости использовать вложенные отчеты.

Табличные отчеты

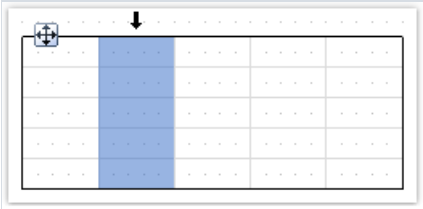
Объект "Таблица" состоит из строк, колонок и ячеек и представляет собой упрощенный аналог таблицы Microsoft Excel. Он выглядит следующим образом:



Настройка колонок

Вы можете удалить или вставить колонки с помощью контекстного меню. Для этого:

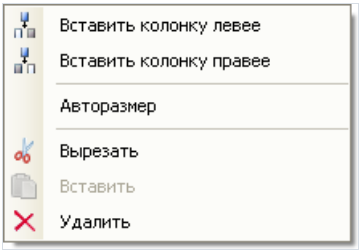
- выделите таблицу или любой ее элемент и поместите указатель мыши над нужной колонкой. Форма указателя поменяется на маленькую черную стрелку:



- нажмите левую кнопку мыши, чтобы выделить колонку;
- нажмите правую кнопку мыши, чтобы показать контекстное меню колонки;
- если вам нужно выделить несколько соседних колонок, нажмите левую кнопку и, не отпуская ее, двигайте мышь влево или вправо, чтобы выделить соседние колонки.

Контекстное меню колонки можно также вызвать в окне "Дерево отчета". Откройте окно, выделите нужную колонку и нажмите правую кнопку мыши.

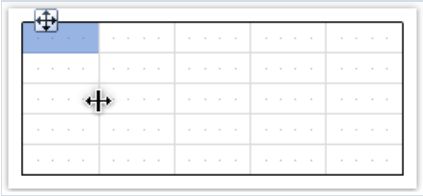
В контекстном меню колонки вы можете выполнить следующие действия:



Управление размером колонок

Вы можете указать ширину колонки одним из следующих способов:

- выделите таблицу или любой ее элемент и поместите указатель мыши на границе между двумя колонками. Форма указателя поменяется на горизонтальный разделитель:



Нажмите левую кнопку мыши и потяните мышь, чтобы изменить размеры колонки;

- выделите колонку и укажите нужную ширину в свойстве "Ширина" (`Width`). Это свойство доступно в окне "Свойства".

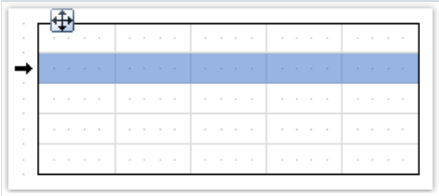
Вы также можете включить свойство колонки "Авторазмер" (`AutoSize`). При запуске отчета ширина колонки будет подобрана автоматически. Для того чтобы ограничить ширину колонки, можно указать свойства "Минимальная ширина" (`MinWidth`) и "Максимальная ширина" (`MaxWidth`).

> Ширина колонки не должна быть больше, чем ширина страницы.

Настройка строк

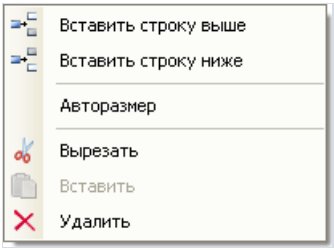
Строки настраиваются аналогичным образом. Чтобы выделить строку, сделайте следующее:

- выделите таблицу или любой ее элемент и поместите указатель мыши слева от нужной строки. Форма указателя поменяется на маленькую черную стрелку:



- нажмите левую кнопку мыши, чтобы выделить строку;
- нажмите правую кнопку мыши, чтобы показать контекстное меню строки;
- если вам нужно выделить несколько соседних колонок, нажмите левую кнопку и, не отпуская ее, двигайте мышь влево или вправо, чтобы выделить соседние колонки.

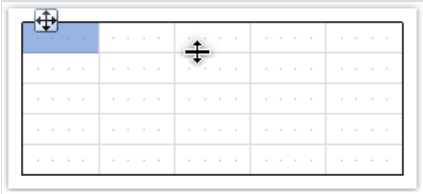
В контекстном меню строки вы можете выполнить следующие действия:



Управление размером строк

Вы можете указать высоту строки одним из следующих способов:

- выделите таблицу или любой ее элемент и поместите указатель мыши на границе между двумя строками. Форма указателя поменяется на вертикальный разделитель:



Нажмите левую кнопку мыши и потяните мышь, чтобы изменить размеры колонки;

- выделите строку и укажите нужную высоту в свойстве "Высота" (`Height`). Это свойство доступно в окне "Свойства".

Вы также можете включить свойство строки "Авторазмер" (`AutoSize`). При запуске отчета высота строки будет подобрана автоматически. Для того чтобы ограничить высоту строки, можно указать свойства "Минимальная высота" (`MinHeight`) и "Максимальная высота" (`MaxHeight`).

> Высота строки не должна быть больше, чем высота страницы.

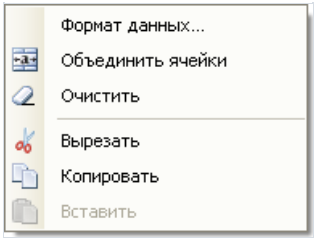
Настройка ячеек

Ячейка представляет собой текстовый объект. По сути, класс ячейки наследуется от объекта "Текст". Все, сказанное выше про объект "Текст", относится и к ячейке таблицы.

Редактировать текст ячейки можно так же, как объект "Текст". Кроме того, вы можете перетащить (drag&drop) элемент из окна "Данные" внутрь ячейки.

Рамка и заливка ячеек могут быть настроены с помощью панели инструментов "Рамка и заливка".

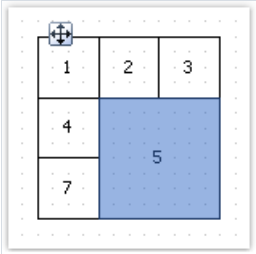
Чтобы вызвать контекстное меню ячейки, щелкните на ней правой кнопкой мыши. В контекстном меню вы можете выполнить следующие действия:



Объединение ячеек

Вы можете объединять соседние ячейки таблицы. При этом получается одна большая ячейка. Для того чтобы это сделать:

- выделите начальную ячейку с помощью мыши;
- нажмите левую кнопку мыши и, не отпуская ее, двигайте мышь, чтобы выделить группу ячеек;
- на выделенной области нажмите правую кнопку мыши, чтобы показать контекстное меню ячейки. В контекстном меню ячейки выберите пункт "Объединить ячейки".



Для того чтобы разбить ячейку, вызовите ее контекстное меню и выберите пункт "Разбить ячейку".

Объекты в ячейках

В ячейку можно добавлять другие объекты отчета, например, рисунки. Следующие объекты добавлять в ячейку нельзя:

- "Таблица";
- "Матрица";
- "Вложенный отчет".

Для того чтобы добавить объект в ячейку, просто перетащите его внутрь ячейки. Вы можете свободно перемещать объект между ячейками, а также вынести его обратно за пределы таблицы.

Ячейка служит контейнером для помещенных в нее объектов. Это значит, что вы можете использовать свойства "Стыковка" (`Dock`) и "Якорь" (`Anchor`) у объектов внутри ячейки. Это позволит изменять размеры объектов при изменении размеров ячейки.

Печать таблицы

Таблица может быть напечатана в двух режимах.

Первый режим – обычный. Таблица печатается на своем бэнде и выглядит так же, как и в дизайнере. В этом режиме ширина таблицы не может быть больше, чем ширина страницы отчета. Это режим печати таблицы по умолчанию.

Второй режим – динамический. В этом режиме таблица строится с помощью скрипта. При этом таблица-результат может отличаться от исходной таблицы так же, как готовый отчет FastReport отличается от шаблона отчета. В динамическом режиме таблица может расти как в ширину, так и в высоту, автоматически разбиваясь на страницы.

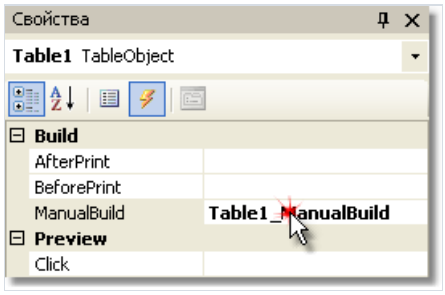
В данном режиме таблица не печатается на бэнде, на который она была положена. Вместо этого она сама генерирует набор бэндов, которые содержат части таблицы-результата. Этот режим работы накладывает следующие ограничения:

- не размещайте другие объекты под таблицей или рядом с ней. Вместо этого используйте бэнд "Дочерний";
- не размещайте два объекта "Таблица" на одном бэнде.

Рассмотрим динамический режим работы более подробно.

> Этот режим работы связан с программированием и требует повышенной квалификации от разработчика отчета.

Формирование таблицы выполняется с помощью скрипта. Чтобы создать скрипт, выделите объект "Таблица", в окне "Свойства" нажмите кнопку "События" и сделайте двойной щелчок на событии `ManualBuild`:



При этом в код отчета добавится пустой обработчик события:

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
}
```

В этом режиме исходная таблица используется как шаблон. В коде обработчика вы можете печатать строки и колонки из исходной таблицы столько раз, сколько необходимо. При этом будет формироваться таблица-результат, которая может иметь неограниченное число колонок и строк. Такая таблица будет автоматически разбиваться на страницы.

Для печати таблицы используются следующие методы объекта "Таблица":

Метод	Параметры	Описание
PrintRow	int index	Печатает строку таблицы с указанным номером. Нумерация строк начинается с 0.
PrintColumn	int index	Печатает колонку таблицы с указанным номером. Нумерация колонок начинается с 0.
PrintRows	int[] indices	Печатает несколько строк таблицы.
PrintRows	-	Печатает все строки таблицы.
PrintColumns	int[] indices	Печатает несколько колонок таблицы.
PrintColumns	-	Печатает все колонки таблицы.
PageBreak	-	Вставляет разрыв страницы перед печатью очередной строки или колонки.

Печать таблицы должна выполняться одним из двух способов.

Способ 1: печать сверху вниз, затем слева направо. Этот способ лучше подходит для печати таблицы с переменным количеством строк. Вы должны вызывать методы печати в следующей последовательности:

- `PrintRow` (номер строки);
- один или несколько вызовов метода `PrintColumn` (номер колонки) или `PrintColumns` (номера колонок) для печати указанных колонок;
- либо один вызов метода `PrintColumns()` для печати всех колонок;
- повторяйте эту последовательность для печати всех нужных строк таблицы.

> Каждая строка таблицы должна содержать одинаковое количество колонок. Имейте это в виду, когда пользуетесь методами `PrintColumn(int index)` и `PrintColumns(int[] indices)`.

Способ 2: печать слева направо, затем сверху вниз. Этот способ лучше подходит для печати таблицы с переменным количеством колонок. Вы должны вызывать методы печати в следующей последовательности:

- `PrintColumn` (номер колонки);
- один или несколько вызовов метода `PrintRow` (номер строки) или `PrintRows` (номера строк) для печати указанных строк;
- либо один вызов метода `PrintRows()` для печати всех строк;
- повторяйте эту последовательность для печати нужных колонок таблицы.

> Каждая колонка таблицы должна содержать одинаковое количество строк. Имейте это в виду, когда пользуетесь методами `PrintRow(int index)` и `PrintRows(int[] indices)`.

Нарушение порядка вызова методов печати приведет к ошибкам при выполнении отчета. Так, одна из ошибок – попытка напечатать таблицу с помощью следующего кода:

```
Table1.PrintRows();
Table1.PrintColumns();
```

Такая последовательность методов неправильная. Вы должны начинать печать таблицы с метода `PrintRow` или `PrintColumn`.

Печать сложных заголовков

Здесь имеются в виду заголовки, которые содержат объединенные ячейки. Когда печатается строка или колонка таблицы, которая содержит объединенную ячейку, ячейка увеличивается в размере автоматически. Покажем это на следующем примере:

Создадим обработчик события `ManualBuild`, который будет печатать первую колонку 3 раза и вторую колонку 1 раз:

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
  // печатаем строку 0 и колонки 0, 0, 0, 1
  Table1.PrintRow(0);
  Table1.PrintColumn(0);
  Table1.PrintColumn(0);
  Table1.PrintColumn(0);
  Table1.PrintColumn(1);
  // печатаем строку 1 и колонки 0, 0, 0, 1
  Table1.PrintRow(1);
  Table1.PrintColumn(0);
  Table1.PrintColumn(0);
  Table1.PrintColumn(0);
  Table1.PrintColumn(1);
}
```

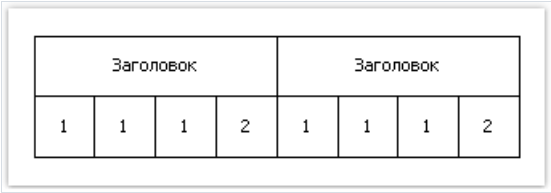
> Обратите внимание, что в каждой строке мы печатаем одинаковое количество колонок. Если нарушить это правило, мы получим непредсказуемый результат.

В результате выполнения этого кода мы получим следующее:

Как видно, заголовок растянулся автоматически. Немного усложним код, чтобы распечатать две группы колонок:

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
  // печатаем строку 0 и две группы колонок 0, 0, 0, 1
  Table1.PrintRow(0);
  // группа 1
  Table1.PrintColumn(0);
  Table1.PrintColumn(0);
  Table1.PrintColumn(0);
  Table1.PrintColumn(1);
  // группа 2
  Table1.PrintColumn(0);
  Table1.PrintColumn(0);
  Table1.PrintColumn(0);
  Table1.PrintColumn(1);

  // печатаем строку 1 и две группы колонок 0, 0, 0, 1
  Table1.PrintRow(1);
  // группа 1
  Table1.PrintColumn(0);
  Table1.PrintColumn(0);
  Table1.PrintColumn(0);
  Table1.PrintColumn(1);
  // группа 2
  Table1.PrintColumn(0);
  Table1.PrintColumn(0);
  Table1.PrintColumn(0);
  Table1.PrintColumn(1);
}
```

Запустим отчет на выполнение и увидим следующий результат:



При печати второй колонки

```
  Table1.PrintColumn(1);
```

заголовок закрывается, и последующая печать первой колонки открывает новый заголовок:

```
  // группа 2
  Table1.PrintColumn(0);
```

Печать итогов

В динамическом режиме объект "Таблица" поддерживает следующие итоговые функции:

Функция	Параметры	Описание
Sum	TableCell cell	Возвращает сумму значений, содержащихся в ячейке cell.
Min	TableCell cell	Возвращает минимальное из значений, содержащихся в ячейке cell.
Max	TableCell cell	Возвращает максимальное из значений, содержащихся в ячейке cell.
Avg	TableCell cell	Возвращает среднее из значений, содержащихся в ячейке cell.
Count	TableCell cell	Возвращает количество строк, содержащих ячейку cell.

> В обычном режиме печати (не динамическом) итоговые функции не работают.

Чтобы использовать итоговую функцию, поместите ее в ячейку таблицы. Например, следующая функция считает сумму значений в ячейке с именем Cell1:

```
[Sum(Cell1)]
```

> При этом анализируются все ячейки, которые находятся выше и левее текущей (в которой мы печатаем итог).

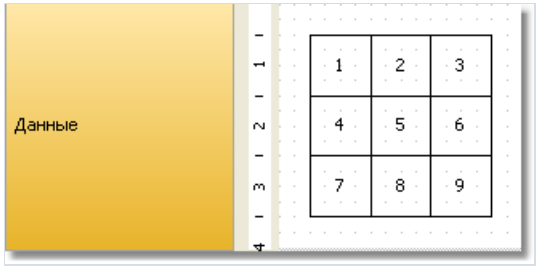
Развертка таблицы

Таблица, которая формируется в динамическом режиме, может автоматически разбиваться на страницы (как по ширине, так и по высоте). Вы можете выбрать один из трех алгоритмов разбивки, за это отвечает свойство `Layout` таблицы:

Значение	Описание
AcrossThenDown	Развертка сначала вбок, затем вниз.
DownThenAcross	Развертка сначала вниз, затем вбок.
Wrapped	Широкая таблица выводится на одной странице.

Таблица

Рассмотрим печать таблицы на примерах. В качестве шаблона будем использовать следующий отчет:

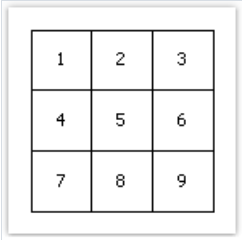


Выделите таблицу и создайте обработчик события `ManualBuild`.

Пример 1. Печать всей таблицы сверху вниз

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
  // печатаем строку 0 и все ее колонки
  Table1.PrintRow(0);
  Table1.PrintColumns();
  // печатаем строку 1 и все ее колонки
  Table1.PrintRow(1);
  Table1.PrintColumns();
  // печатаем строку 2 и все ее колонки
  Table1.PrintRow(2);
  Table1.PrintColumns();
}
```

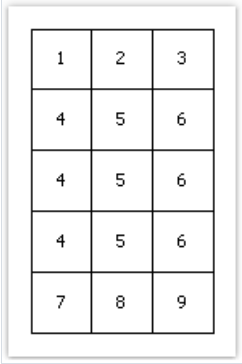
В результате будет напечатана следующая таблица, которая не отличается от шаблона:



Пример 2. Печать таблицы сверху вниз, с повторением строк

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
  // печатаем строку 0 и все ее колонки
  Table1.PrintRow(0);
  Table1.PrintColumns();
  // печатаем 3 экземпляра строки 1 и все ее колонки
  for (int i = 0; i < 3; i++)
  {
    Table1.PrintRow(1);
    Table1.PrintColumns();
  }
  // печатаем строку 2 и все ее колонки
  Table1.PrintRow(2);
  Table1.PrintColumns();
}
```

В этом примере средняя строка таблицы печатается 3 раза. В результате получается следующее:

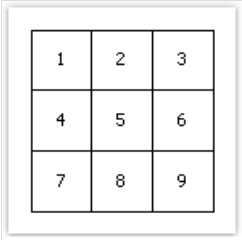


1	2	3
4	5	6
4	5	6
4	5	6
7	8	9

Пример 3. Печать всей таблицы слева направо

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
  // печатаем колонку 0 и все ее строки
  Table1.PrintColumn(0);
  Table1.PrintRows();
  // печатаем колонку 1 и все ее строки
  Table1.PrintColumn(1);
  Table1.PrintRows();
  // печатаем колонку 2 и все ее строки
  Table1.PrintColumn(2);
  Table1.PrintRows();
}
```

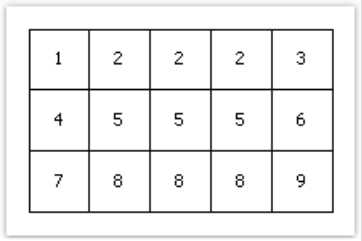
В результате будет напечатана следующая таблица, которая не отличается от шаблона:



Пример 4. Печать таблицы слева направо, с повторением колонок

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
  // печатаем колонку 0 и все ее строки
  Table1.PrintColumn(0);
  Table1.PrintRows();
  // печатаем 3 экземпляра колонки 1 и все ее строки
  for (int i = 0; i < 3; i++)
  {
    Table1.PrintColumn(1);
    Table1.PrintRows();
  }
  // печатаем колонку 2 и все ее строки
  Table1.PrintColumn(2);
  Table1.PrintRows();
}
```

В этом примере средняя колонка таблицы печатается 3 раза. В результате получается следующее:



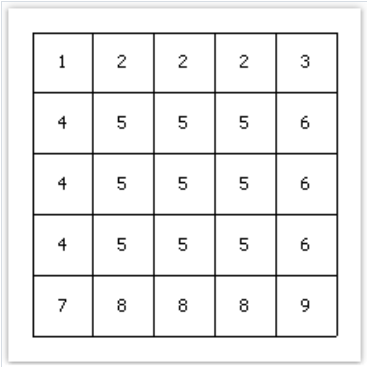
1	2	2	2	3
4	5	5	5	6
7	8	8	8	9

Пример 5. Печать таблицы с повторением строк и колонок

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
  // --------- печатаем строку 0
  Table1.PrintRow(0);
  // печатаем колонку 0
  Table1.PrintColumn(0);
  // печатаем 3 экземпляра колонки 1
  for (int i = 0; i < 3; i++)
    Table1.PrintColumn(1);
  // печатаем колонку 2
  Table1.PrintColumn(2);

  // --------- печатаем 3 экземпляра строки 1
  for (int j = 0; j < 3; j++)
  {
    Table1.PrintRow(1);
    // печатаем колонку 0
    Table1.PrintColumn(0);
    // печатаем 3 экземпляра колонки 1
    for (int i = 0; i < 3; i++)
      Table1.PrintColumn(1);
    // печатаем колонку 2
    Table1.PrintColumn(2);
  }
  // --------- печатаем строку 2
  Table1.PrintRow(2);
  // печатаем колонку 0
  Table1.PrintColumn(0);
  // печатаем 3 экземпляра колонки 1
  for (int i = 0; i < 3; i++)
    Table1.PrintColumn(1);
  // печатаем колонку 2
  Table1.PrintColumn(2);
}
```

> Обратите внимание, что в каждой строке мы печатаем одинаковое количество колонок. Если нарушить это правило, мы получим непредсказуемый результат.

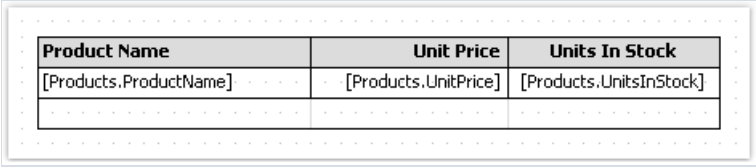
В этом примере средняя строка и средняя колонка таблицы печатается 3 раза. В результате получается следующее:



1	2	2	2	3
4	5	5	5	6
4	5	5	5	6
4	5	5	5	6
7	8	8	8	9

Пример 6. Привязка к источникам данных

Во всех рассмотренных примерах мы печатали таблицу, которая содержит обычный текст. В этом примере мы покажем, как сформировать таблицу, используя источник данных. Для этого создадим таблицу следующего вида:



Product Name	Unit Price	Units In Stock
[Products.ProductName]	[Products.UnitPrice]	[Products.UnitsInStock]

Создадим обработчик события `ManualBuild`, который будет делать следующее:

- получит ссылку на источник данных, определенный в отчете;
- инициализирует его (заполнит данными);
- распечатает строки таблицы столько раз, сколько записей в источнике данных.

Вот код обработчика:

```
private void Table1_ManualBuild(object sender, EventArgs e)
{
  // получаем источник данных по его имени
  DataSourceBase rowData = Report.GetDataSource("Products");
  // инициализируем его
  rowData.Init();

  // печатаем заголовок таблицы
  Table1.PrintRow(0);
  Table1.PrintColumns();

  // выполняем цикл, пока в источнике данных есть записи
  while (rowData.HasMoreRows)
  {
    // печатаем строку таблицы
    Table1.PrintRow(1);
    Table1.PrintColumns();

    // переходим на следующую запись источника
    rowData.Next();
  }

  // печатаем подвал таблицы
  Table1.PrintRow(2);
  Table1.PrintColumns();
}
```

Примерно такие же действия выполняет FastReport при печати бэнда "Данные". Если запустить отчет, мы получим следующее:

Product Name	Unit Price	Units In Stock
Chai	18,00р.	39
Chang	19,00р.	17
Aniseed Syrup	10,00р.	13
Chef Anton's Cajun Seasoning	22,00р.	53
Chef Anton's Gumbo Mix	21,35р.	0
Grandma's Boysenberry Spread	25,00р.	120
Uncle Bob's Organic Dried Pears	30,00р.	15
Northwoods Cranberry Sauce	40,00р.	6

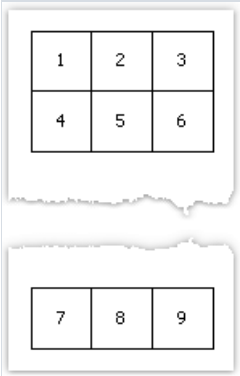
Пример 7. Вставка разрыва страницы

Для вставки разрыва страницы используется метод `PageBreak` объекта "Таблица". Вызывайте его перед тем, как напечатать очередную строку или колонку таблицы.

Рассмотрим использование метода `PageBreak` на примере 1. Сделаем так, чтобы третья строка печаталась на новой странице.

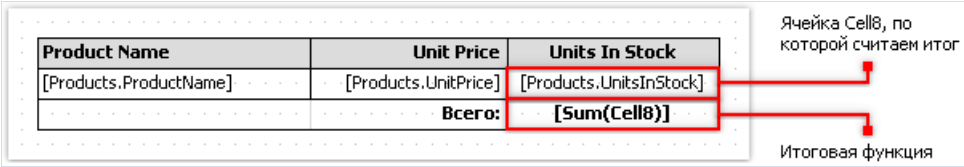
```
private void Table1_ManualBuild(object sender, EventArgs e)
{
  // печатаем строку 0 и все ее колонки
  Table1.PrintRow(0);
  Table1.PrintColumns();
  // печатаем строку 1 и все ее колонки
  Table1.PrintRow(1);
  Table1.PrintColumns();
  // делаем разрыв страницы
  Table1.PageBreak();
  // печатаем строку 2 и все ее колонки
  Table1.PrintRow(2);
  Table1.PrintColumns();
}
```

В результате будет напечатана такая таблица:

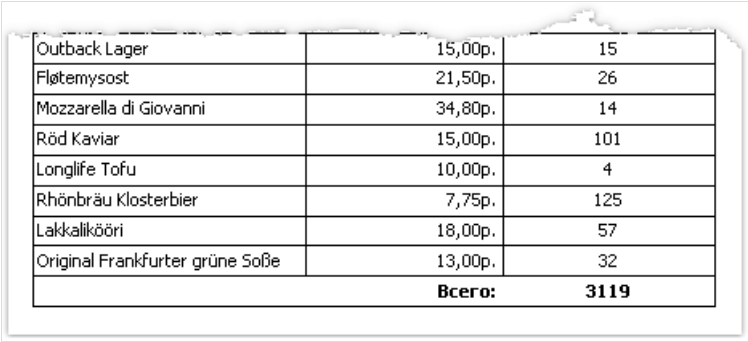


Пример 8. Печать итогов

Рассмотрим использование итоговых функций на Примере 6. Модифицируем его следующим образом:



Если запустить отчет, мы получим следующее:

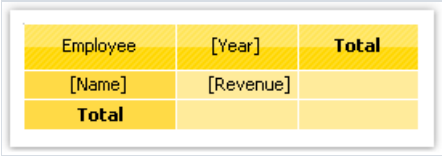


Outback Lager	15,00р.	15
Fløtemysost	21,50р.	26
Mozzarella di Giovanni	34,80р.	14
Röd Kaviar	15,00р.	101
Longlife Tofu	10,00р.	4
Rhönbräu Klosterbier	7,75р.	125
Lakkalikööri	18,00р.	57
Original Frankfurter grüne Soße	13,00р.	32
	Всего:	**3119**

Матричные (сводные) отчеты

Объект "Матрица" является разновидностью таблицы и, как и объект "Таблица", состоит из строк, колонок и ячеек. Причем заранее неизвестно, сколько строк и столбцов будет в матрице - это зависит от данных, к которым она подключена.

Объект выглядит следующим образом:



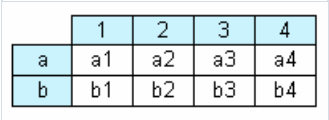
Employee	[Year]	**Total**
[Name]	[Revenue]	
Total		

При печати матрица заполняется значениями и растет как вниз, так и вбок. Результат может выглядеть следующим образом:

Employee	1999	2000	2001	2002	**Total**
Andrew Fuller	3 900,00р.	2 100,00р.		1 800,00р.	7 800,00р.
Janet Leverling	6 100,00р.	3 200,00р.			9 300,00р.
Nancy Davolio	3 300,00р.	2 700,00р.	3 100,00р.	1 700,00р.	10 800,00р.
Steven Buchanan		3 999,00р.	8 100,00р.		12 099,00р.
Total	13 300,00р.	11 999,00р.	11 200,00р.	3 500,00р.	**39 999,00р.**

Немного теории

Рассмотрим элементы матрицы:



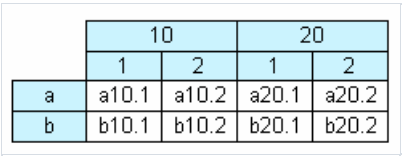
	1	2	3	4
a	a1	a2	a3	a4
b	b1	b2	b3	b4

На рисунке мы видим матрицу с двумя строками и четырьмя столбцами. Здесь a и b – заголовки строк, 1, 2, 3, 4 – заголовки столбцов, a1..a4, b1..b4 – ячейки. Чтобы построить такой отчет, понадобится всего один набор данных (запрос или таблица), который имеет три поля и содержит следующие данные:

```
a  1  a1

a  2  a2

a  3  a3

a  4  a4

b  1  b1

b  2  b2

b  3  b3

b  4  b4
```

Как видно, первое поле содержит номер строки, второе – номер столбца, третье – содержимое ячейки на пересечении строки и столбца с указанным номером. При построении отчета FastReport создает в памяти матрицу и заполняет ее данными. При этом матрица динамически расширяется, если строки или столбца с заданным номером еще не существует.

Заголовки могут иметь более одного уровня. Рассмотрим следующий пример:



	10		20	
	1	2	1	2
a	a10.1	a10.2	a20.1	a20.2
b	b10.1	b10.2	b20.1	b20.2

В этом примере номер, или индекс, столбца – составной, т.е. состоит из двух значений. Этот отчет требует следующих данных:

```
a  10  1  a10.1

a  10  2  a10.2

a  20  1  a20.1

a  20  2  a20.2

b  10  1  b10.1

b  10  2  b10.2

b  20  1  b20.1

b  20  2  b20.2
```

Здесь первое поле, как и прежде, содержит индекс строки, второе и третье поля – индекс колонки. Последнее поле содержит значение ячейки.

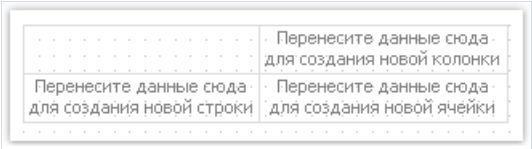
Следующий элемент матрицы – промежуточные итоги и итоги, демонстрирует следующий рисунок:

	10			20			Total
	1	2	Total	1	2	Total	
a	a10.1	a10.2	a10.1+a10.2	a20.1	a20.2	a20.1+a20.2	sum(a)
b	b10.1	b10.2	b10.1+b10.2	b20.1	b20.2	b20.1+b20.2	sum(b)
Total	a10.1+ b10.1	a10.2+ b10.2	a10.1+b10.1+ a10.2+b10.2	a20.1+ b20.1	a20.2+ b20.2	a20.1+b20.1+ a20.2+b20.2	sum(a)+ sum(b)

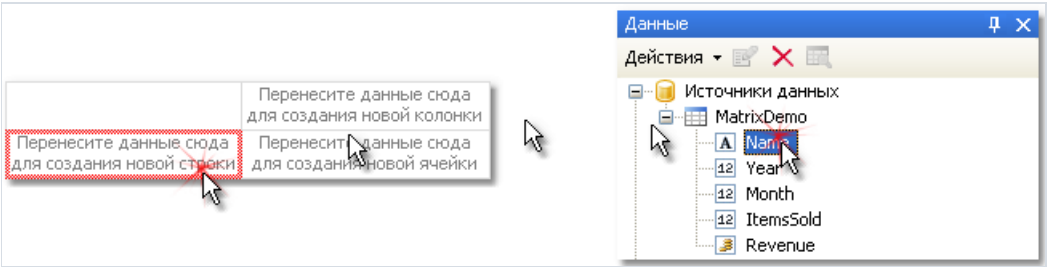
Этот отчет строится на тех же данных, что и предыдущий. Столбцы, показанные серым на рисунке, вычисляются автоматически.

Настройка структуры

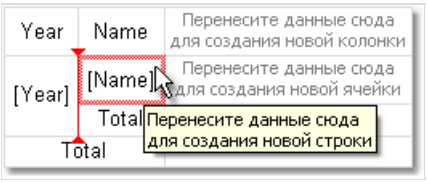
После того, как вы положили на лист отчета новый объект "Матрица", он выглядит следующим образом:



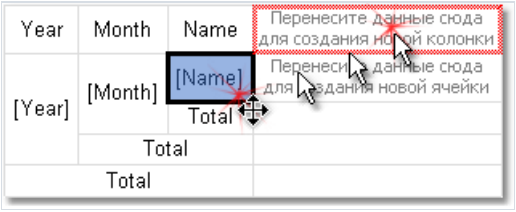
Настройка структуры матрицы выполняется с помощью мыши. Для этого перетаскивайте (drag&drop) поля источников данных из окна "Данные" на матрицу, чтобы создать строки, колонки и ячейки. Матрица подсвечивает место, куда будут вставлены новые данные, красной рамкой:



Если в матрице уже есть элементы (заголовки, ячейки), то при вставке нового элемента показывается индикатор. В данном случае новое поле будет вставлено между полями `Year` и `Name`:



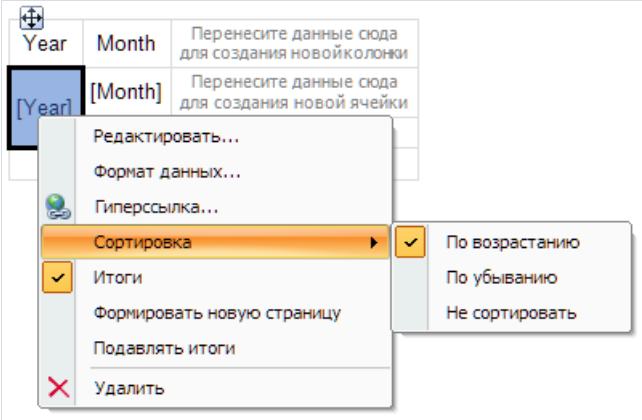
Вы также можете изменить порядок существующих элементов. Для этого щелкните мышкой на элементе и перетащите его на новое место, взяв его за рамку:



Для удаления элемента выберите его мышкой и нажмите клавишу `Delete`.

Настройка заголовка

Для настройки элемента заголовка выделите его и нажмите правую кнопку мыши, чтобы показать контекстное меню:



По умолчанию данные в заголовке матрицы сортируются по возрастанию. Вы можете поменять порядок сортировки, выбрав пункт "Сортировка" в меню.

К каждому полю, которое вы помещаете в заголовок матрицы, FastReport добавляет итог (это ячейка с текстом "Итого"). Вы можете удалить итог, выделив его и нажав клавишу `Delete`. Чтобы включить итог снова, выделите элемент, к которому он относится, и в контекстном меню выберите пункт "Итоги".

Пункт меню "Формировать новую страницу" позволяет добавить разрыв страницы при печати матрицы. Если включить разрыв у ячейки `Year`, как показано на рисунке выше, то при печати матрицы каждый год будет расположен на отдельной странице.

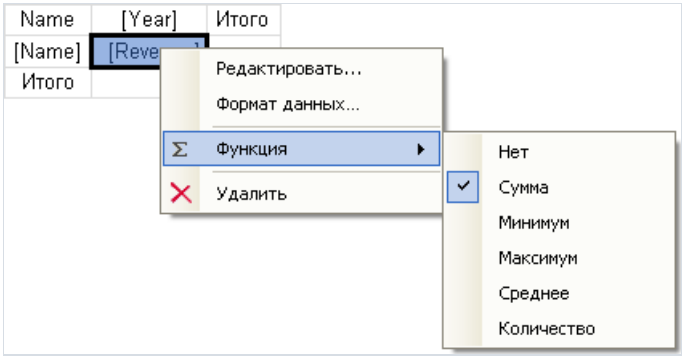
Пункт меню "Подавлять итоги" позволяет подавлять строку итогов в случае, если в группе, по которой считается итог, есть только одно значение.

Настройка ячеек

Для ячеек матрицы можно выбрать функцию, которая будет использована при расчете итогов. Ее значение будет выведено в строках и столбцах итогов. Ниже приведен список функций, которые можно использовать:

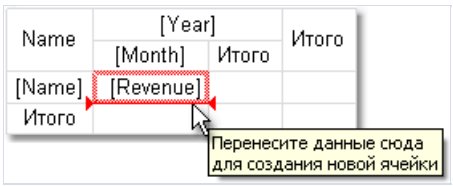
Функция	Описание
Нет	Значения ячеек не обрабатываются.
Сумма	Возвращает сумму значений в ячейках матрицы.
Минимум	Возвращает минимальное из значений.
Максимум	Возвращает максимальное из значений.
Среднее	Возвращает среднее значение.
Количество	Возвращает количество непустых значений.
Количество разных	Возвращает количество разных (уникальных) значений.

Для ячеек, добавляемых в матрицу, по умолчанию используется функция суммирования. Вы можете выбрать другую функцию, выделив ячейку и выбрав в ее контекстном меню пункт "Функция":



Выберите функцию "Нет", если вы не собираетесь печатать итоги по данной ячейке.

В матрице может быть одна или несколько ячеек данных. Если ячеек несколько, они могут располагаться рядом или друг под другом. За это отвечает свойство матрицы "Ячейки одной строкой", которое можно поменять из контекстного меню. Вы также можете выбрать порядок расположения ячеек при добавлении второй ячейки в матрицу. При этом ориентируйтесь на красные индикаторы, которые показывают, куда будет добавляться второе значение - вверх-вниз или вправо-влево:



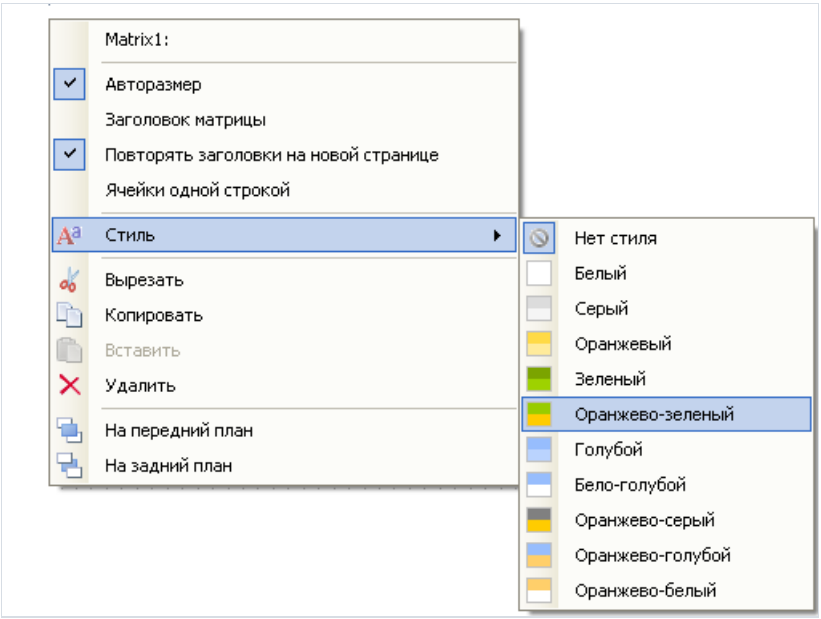
После того как вы добавили второе значение, все последующие значения будут добавляться в выбранном направлении.

Настройка внешнего вида

Чтобы изменить внешний вид ячеек матрицы, щелкните на нужной ячейке. С помощью панели инструментов можно изменить настройки шрифта, рамки и заливки. Ячейка матрицы представляет собой наследник объекта "Текст", и к ней применимо все, сказанное ранее про этот объект.

Чтобы изменить оформление сразу нескольких ячеек, выделите группу ячеек. Для этого выделите левую верхнюю ячейку, и, не отпуская кнопки мыши, тяните мышь, чтобы выделить группу:

Вы также можете выбрать один из стилей для изменения внешнего вида всей матрицы. Для этого щелкните на объекте правой кнопкой мыши и в контекстном меню выберите стиль:



Управление размерами строк и колонок

Так как матрица является разновидностью объекта "Таблица", она позволяет задавать размеры строк и колонок аналогичным образом.

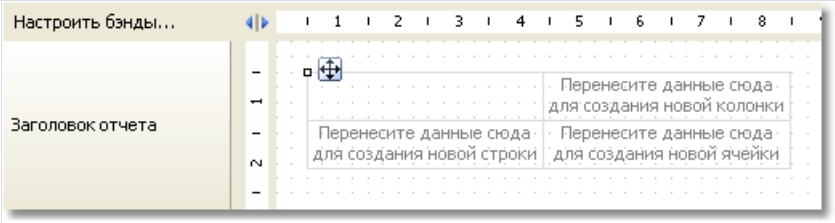
По умолчанию у матрицы включен режим "Авторазмер". В этом режиме объект сам подбирает ширину колонок и высоту строк в зависимости от содержащихся в них данных. Вы также можете управлять размером объекта вручную. Для этого выключите свойство "Авторазмер" у матрицы. У строк и колонок есть аналогичное свойство; его можно использовать, если авторазмер у матрицы отключен.

Чтобы ограничить минимальную и максимальную высоту строки, выберите строку и в окне "Свойства" укажите значения свойств `MinHeight` и `MaxHeight`.

Чтобы ограничить минимальную и максимальную ширину колонки, выберите колонку и в окне "Свойства" укажите значения свойств `MinWidth` и `MaxWidth`.

Матрица

Рассмотрим примеры использования объекта "Матрица". Для начала, создайте новый отчет и положите объект "Матрица" на бэнд "Заголовок отчета" (или бэнд "Данные", в этом случае подключать бэнд к источнику данных не надо). В данном случае неважно, на какой из этих двух бэндов вы положите матрицу, т.к. оба бэнда будут печататься один раз при запуске отчета. Отчет будет выглядеть следующим образом:



> Не кладите объект на бэнды, которые будут печататься на каждой новой странице - "Заголовок страницы", "Подвал страницы" и т.п. Матрица в этом случае будет строиться каждый раз при печати бэнда, что приведет к переполнению стека и аварийному завершению работы программы.

Большинство из примеров будут использовать таблицу MatrixDemo, которая поставляется в комплекте FastReport. Эта таблица содержит следующие данные:

Name	Year	Month	ItemsSold	Revenue
Nancy Davolio	1999	2	1	1000
Nancy Davolio	1999	11	1	1100
Nancy Davolio	1999	12	1	1200
Nancy Davolio	2000	1	1	1300
Nancy Davolio	2000	2	2	1400
Nancy Davolio	2001	2	2	1500
Nancy Davolio	2001	3	2	1600
Nancy Davolio	2002	1	2	1700
Andrew Fuller	2002	1	2	1800
Andrew Fuller	1999	10	2	1900
Andrew Fuller	1999	11	2	2000
Andrew Fuller	2000	2	2	2100
Janet Leverling	1999	10	3	3000

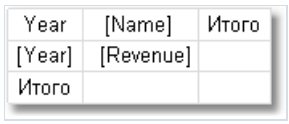
Name	Year	Month	ItemsSold	Revenue
Janet Leverling	1999	11	3	3100
Janet Leverling	2000	3	3	3200
Steven Buchanan	2001	1	3	4000
Steven Buchanan	2001	2	4	4100
Steven Buchanan	2000	1	4	3999

Пример 1. Простая матрица

Матрица будет содержать одно значение в строке и колонке и одну ячейку данных. Для того чтобы построить матрицу, добавьте поля источника данных MatrixDemo следующим образом:

- поле `Year` добавьте в заголовок строки;
- поле `Name` добавьте в заголовок колонки;
- поле `Revenue` добавьте в ячейку матрицы.

После этого матрица будет выглядеть следующим образом:

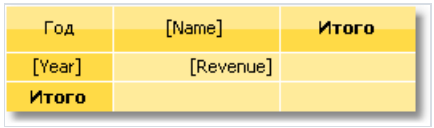


Year	[Name]	Итого
[Year]	[Revenue]	
Итого		

Давайте улучшим внешний вид матрицы следующим образом:

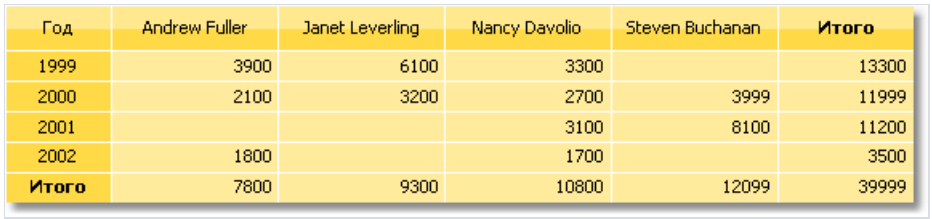
- в верхний левый угол автоматически добавилось имя поля, которое содержится в строке матрицы. Вместо "Year" напишем слово "Год";
- выберем для матрицы стиль "Оранжевый";
- выберем шрифт "Tahoma, 8" для всех ячеек матрицы;
- ячейки со словом "Итого" выделим жирным шрифтом;
- для ячеек в верхней строке таблицы выберем заливку типа "Стекло";
- отключим авторазмер у матрицы и увеличим размеры строк и колонок.

После этого матрица будет выглядеть так:



Год	[Name]	**Итого**
[Year]	[Revenue]	
Итого		

Запустите отчет на построение, и вы увидите следующий результат:



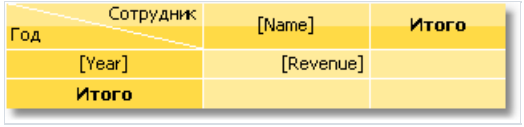
Год	Andrew Fuller	Janet Leverling	Nancy Davolio	Steven Buchanan	**Итого**
1999	3900	6100	3300		13300
2000	2100	3200	2700	3999	11999
2001			3100	8100	11200
2002	1800		1700		3500
Итого	7800	9300	10800	12099	39999

В матрице не хватает следующих вещей:

- нет названия для поля `Name`;
- суммы печатаются не в денежном формате.

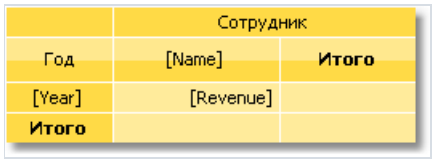
Добавить название для поля `Name` можно следующими способами:

- в левый верхний угол матрицы можно поместить текст "Год/Сотрудник";
- туда же можно поместить диагональную линию и второй объект "Текст", как показано на рисунке ниже:



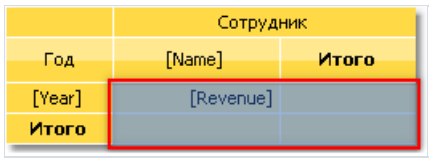
Сотрудник / Год	[Name]	**Итого**
[Year]	[Revenue]	
Итого		

- можно включить верхнюю строку заголовка матрицы. Для этого в контекстном меню объекта "Матрица" выберите пункт "Заголовок матрицы". В заголовок можно помещать любой текст:



	Сотрудник	
Год	[Name]	**Итого**
[Year]	[Revenue]	
Итого		

Для того чтобы задать денежный формат для ячеек матрицы, выделите всю область ячеек, как показано на рисунке, и установите формат, выбрав пункт "Формат данных..." в контекстном меню:



	Сотрудник	
Год	[Name]	**Итого**
[Year]	[Revenue]	
Итого		

После этого готовый отчет будет выглядеть так:

	Сотрудник				
Год	Andrew Fuller	Janet Leverling	Nancy Davolio	Steven Buchanan	**Итого**
1999	3 900,00р.	6 100,00р.	3 300,00р.		13 300,00р.
2000	2 100,00р.	3 200,00р.	2 700,00р.	3 999,00р.	11 999,00р.
2001			3 100,00р.	8 100,00р.	11 200,00р.
2002	1 800,00р.		1 700,00р.		3 500,00р.
Итого	7 800,00р.	9 300,00р.	10 800,00р.	12 099,00р.	39 999,00р.

Пример 2. Многоуровневые заголовки

Матрица будет содержать одно значение в строке, два значения в колонке и одну ячейку данных. Возьмем за основу предыдущий пример и добавим в него поле:

- поле `Month` добавьте в заголовок строки справа от поля `Year`.

После добавления нового поля поправьте внешний вид матрицы. Также необходимо вновь указать форматирование для ячеек. После этого матрица будет выглядеть следующим образом:

		Сотрудник	
Год	Месяц	[Name]	**Итого**
[Year]	[Month]	[Revenue]	
	Итого		
Итого			

Запустите отчет на выполнение, и вы увидите следующий результат:

		Сотрудник				
Год	Месяц	Andrew Fuller	Janet Leverling	Nancy Davolio	Steven Buchanan	**Итого**
1999	2			1 000,00р.		1 000,00р.
	10	1 900,00р.	3 000,00р.			4 900,00р.
	11	2 000,00р.	3 100,00р.	1 100,00р.		6 200,00р.
	12			1 200,00р.		1 200,00р.
	Итого	3 900,00р.	6 100,00р.	3 300,00р.	0,00р.	13 300,00р.
2000	1			1 300,00р.	3 999,00р.	5 299,00р.
	2	2 100,00р.		1 400,00р.		3 500,00р.
	3		3 200,00р.			3 200,00р.
	Итого	2 100,00р.	3 200,00р.	2 700,00р.	3 999,00р.	11 999,00р.
2001	1				4 000,00р.	4 000,00р.
	2			1 500,00р.	4 100,00р.	5 600,00р.
	3			1 600,00р.		1 600,00р.
	Итого	0,00р.	0,00р.	3 100,00р.	8 100,00р.	11 200,00р.
2002	1	1 800,00р.		1 700,00р.		3 500,00р.
	Итого	1 800,00р.	0,00р.	1 700,00р.	0,00р.	3 500,00р.
Итого		7 800,00р.	9 300,00р.	10 800,00р.	12 099,00р.	39 999,00р.

Пример 3. Печать названия месяца вместо номера

В предыдущем примере в матрице печатались номера месяцев. Это происходило потому, что в источнике данных поле `Month` содержит номер месяца, а не его название. Чтобы напечатать название месяца, сделайте следующее:

- выберите ячейку, в которой печатается номер месяца. В нашем примере это ячейка с именем Cell8;
- в окне "Свойства" нажмите кнопку  и сделайте двойной щелчок на событии `BeforePrint`;
- FastReport добавит пустой обработчик события в скрипт отчета. Напишите следующий код:

```
private void Cell8_BeforePrint(object sender, EventArgs e)
{
  string[] monthNames = new string[] {
    "Январь", "Февраль", "Март", "Апрель", "Май", "Июнь",
    "Июль", "Август", "Сентябрь", "Октябрь", "Ноябрь", "Декабрь" };
  // Cell8 - это ячейка, которая печатает номер месяца.
  // Cell8.Value - это значение, которое печатается в ячейке (номер месяца).
  // Это значение типа System.Object, поэтому его надо привести к int
  Cell8.Text = monthNames[(int)Cell8.Value - 1];
}
```

Если запустить отчет на выполнение, вы увидите следующий результат:

		Сотрудник				
Год	Месяц	Andrew Fuller	Janet Leverling	Nancy Davolio	Steven Buchanan	**Итого**
1999	Февраль			1 000,00р.		1 000,00р.
	Октябрь	1 900,00р.	3 000,00р.			4 900,00р.
	Ноябрь	2 000,00р.	3 100,00р.	1 100,00р.		6 200,00р.
	Декабрь			1 200,00р.		1 200,00р.
	Итого	3 900,00р.	6 100,00р.	3 300,00р.	0,00р.	13 300,00р.
2000	Январь			1 300,00р.	3 999,00р.	5 299,00р.
	Февраль	2 100,00р.		1 400,00р.		3 500,00р.
	Март		3 200,00р.			3 200,00р.
	Итого	2 100,00р.	3 200,00р.	2 700,00р.	3 999,00р.	11 999,00р.
2001	Январь				4 000,00р.	4 000,00р.
	Февраль			1 500,00р.	4 100,00р.	5 600,00р.
	Март			1 600,00р.		1 600,00р.
	Итого	0,00р.	0,00р.	3 100,00р.	8 100,00р.	11 200,00р.
2002	Январь	1 800,00р.		1 700,00р.		3 500,00р.
	Итого	1 800,00р.	0,00р.	1 700,00р.	0,00р.	3 500,00р.
Итого		7 800,00р.	9 300,00р.	10 800,00р.	12 099,00р.	39 999,00р.

Пример 4. Условное выделение

Для ячеек матрицы, как и для объекта "Текст", можно задать условное выделение. Подробнее об этом можно прочитать в разделе "Условное выделение".

Рассмотрим на примере 2, как выделить суммы больше 3000 красным цветом. Для этого выберите ячейку с текстом `Revenue` и нажмите кнопку  на панели инструментов "Текст". В редакторе условий задайте условие выделения:

```
Value > 3000
```

и цвет текста - красный. Готовый отчет будет выглядеть так:

		Сотрудник				
Год	Месяц	Andrew Fuller	Janet Leverling	Nancy Davolio	Steven Buchanan	**Итого**
1999	2			1 000,00р.		1 000,00р.
	10	1 900,00р.	3 000,00р.			4 900,00р.
	11	2 000,00р.	3 100,00р.	1 100,00р.		6 200,00р.
	12			1 200,00р.		1 200,00р.
	Итого	3 900,00р.	6 100,00р.	3 300,00р.	0,00р.	13 300,00р.
2000	1			1 300,00р.	3 999,00р.	5 299,00р.
	2	2 100,00р.		1 400,00р.		3 500,00р.
	3		3 200,00р.			3 200,00р.
	Итого	2 100,00р.	3 200,00р.	2 700,00р.	3 999,00р.	11 999,00р.
2001	1				4 000,00р.	4 000,00р.
	2			1 500,00р.	4 100,00р.	5 600,00р.
	3			1 600,00р.		1 600,00р.
	Итого	0,00р.	0,00р.	3 100,00р.	8 100,00р.	11 200,00р.
2002	1	1 800,00р.		1 700,00р.		3 500,00р.
	Итого	1 800,00р.	0,00р.	1 700,00р.	0,00р.	3 500,00р.
Итого		7 800,00р.	9 300,00р.	10 800,00р.	12 099,00р.	39 999,00р.

Как видно, итоговые значения не выделяются. Это происходит потому, что мы задали условие выделения только для одной ячейки. Чтобы выделить остальные значения, надо задать условие для всех ячеек матрицы, как показано на рисунке.

В этом примере мы использовали условное выделение, которое зависит от значения самой ячейки. Кроме того, можно выделить ячейку в зависимости от значений из заголовка матрицы. Покажем на следующем примере, как выделить все суммы за 2000 год. Для этого выделите ячейки матрицы, как показано на рисунке:

		Сотрудник	
Год	Месяц	[Name]	**Итого**
[Year]	[Month]	[Revenue]	
	Итого		
Итого			

Задайте для ячеек такое условие выделения:

```
(int)Matrix1.RowValues[0] == 2000
```

В данном случае Matrix1 - это имя нашей матрицы. Свойство матрицы `RowValues` имеет тип `object[]` и содержит массив значений из заголовка строки, для текущей печатаемой строки. В массиве столько

значений, сколько уровней в заголовке. В нашем примере массив имеет 2 значения, первое из которых - год, второе - месяц.

> Не выделяйте последнюю строку. Свойство `RowValues` для нее имеет неопределенное значение и вызовет ошибку в процессе построения отчета.

Если запустить отчет, мы получим следующий результат:

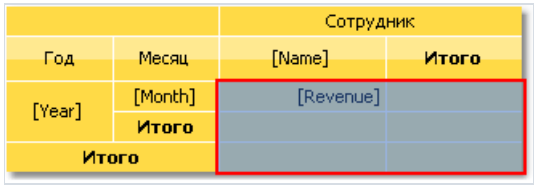
		Сотрудник				
Год	Месяц	Andrew Fuller	Janet Leverling	Nancy Davolio	Steven Buchanan	**Итого**
1999	2			1 000,00р.		1 000,00р.
	10	1 900,00р.	3 000,00р.			4 900,00р.
	11	2 000,00р.	3 100,00р.	1 100,00р.		6 200,00р.
	12			1 200,00р.		1 200,00р.
	Итого	3 900,00р.	6 100,00р.	3 300,00р.	0,00р.	13 300,00р.
2000	1			1 300,00р.	3 999,00р.	5 299,00р.
	2	2 100,00р.		1 400,00р.		3 500,00р.
	3		3 200,00р.			3 200,00р.
	Итого	2 100,00р.	3 200,00р.	2 700,00р.	3 999,00р.	11 999,00р.
2001	1				4 000,00р.	4 000,00р.
	2			1 500,00р.	4 100,00р.	5 600,00р.
	3			1 600,00р.		1 600,00р.
	Итого	0,00р.	0,00р.	3 100,00р.	8 100,00р.	11 200,00р.
2002	1	1 800,00р.		1 700,00р.		3 500,00р.
	Итого	1 800,00р.	0,00р.	1 700,00р.	0,00р.	3 500,00р.
Итого		7 800,00р.	9 300,00р.	10 800,00р.	12 099,00р.	39 999,00р.

> Вы также можете использовать свойство матрицы `ColumnValues` для обращения к значениям колонок.

Пример 5. Выделение четных строк

Для улучшения внешнего вида матрицы можно применить выделение четных строк или колонок другим цветом. Покажем, как это сделать, на примере 2.

Выделите всю область данных матрицы, как показано на рисунке:



Вызовите редактор условного выделения. В качестве условия задайте следующее:

```
Matrix1.RowIndex % 2 != 0
```

и укажите цвет фона чуть темнее, чем текущий. В данном примере Matrix1 - это имя нашей матрицы; свойство `RowIndex` возвращает номер текущей печатаемой строки.

> Для выделения колонок используйте свойство матрицы `ColumnIndex` аналогичным способом.

Если запустить отчет, мы увидим следующее:

		Сотрудник				
Год	Месяц	Andrew Fuller	Janet Leverling	Nancy Davolio	Steven Buchanan	**Итого**
1999	2			1 000,00р.		1 000,00р.
	10	1 900,00р.	3 000,00р.			4 900,00р.
	11	2 000,00р.	3 100,00р.	1 100,00р.		6 200,00р.
	12			1 200,00р.		1 200,00р.
	Итого	3 900,00р.	6 100,00р.	3 300,00р.	0,00р.	13 300,00р.
2000	1			1 300,00р.	3 999,00р.	5 299,00р.
	2	2 100,00р.		1 400,00р.		3 500,00р.
	3		3 200,00р.			3 200,00р.
	Итого	2 100,00р.	3 200,00р.	2 700,00р.	3 999,00р.	11 999,00р.
2001	1				4 000,00р.	4 000,00р.
	2			1 500,00р.	4 100,00р.	5 600,00р.
	3			1 600,00р.		1 600,00р.
	Итого	0,00р.	0,00р.	3 100,00р.	8 100,00р.	11 200,00р.
2002	1	1 800,00р.		1 700,00р.		3 500,00р.
	Итого	1 800,00р.	0,00р.	1 700,00р.	0,00р.	3 500,00р.
Итого		7 800,00р.	9 300,00р.	10 800,00р.	12 099,00р.	39 999,00р.

Пример 6. Использование выражений

В предыдущих примерах мы создавали заголовки и ячейки матрицы, перетаскивая в нее поля из окна "Данные". Вы также можете использовать для этих целей выражения. Для того, чтобы поместить выражение в матрицу, сделайте следующее:

- добавьте в матрицу какой-нибудь элемент из окна "Данные". Это может быть любой элемент, например, системная переменная `Date`;
- сделайте двойной щелчок на элементе и укажите нужное выражение в окне редактора выражений.

> Если ваша матрица содержит выражения вместо полей данных, проследите за тем, чтобы свойство матрицы `DataSource` было корректно заполнено. При работе с полями данных это свойство заполняется автоматически, когда вы переносите поле на матрицу.

Рассмотрим использование выражений на примере. Для этого в качестве источника данных будем использовать таблицу Order Details, которая содержит список проданных товаров, сгруппированный по сотрудникам. Таблица содержит несколько родительских связей, что позволяет получить доступ к имени сотрудника, наименованию товара и его категории.

Наша матрица будет показывать продажи каждого из сотрудников, сгруппированные по категориям товаров. Чтобы построить матрицу, сделайте следующее:

- поле `[Order Details.Products.Categories.CategoryName]` добавьте в заголовок колонки;
- в заголовок строки добавьте любое поле, чтобы создать элемент матрицы. Затем укажите следующее выражение для элемента заголовка:

```
[Order Details.Orders.Employees.FirstName] + " " + [Order Details.Orders.Employees.LastName]
```

- в ячейку данных добавьте любое поле, чтобы создать элемент матрицы. Затем укажите следующее выражение для ячейки:

```
[Order Details.UnitPrice] * [Order Details.Quantity] * (decimal)(1 - [Order Details.Discount])
```

> Почему в качестве имени сотрудника мы указали такое длинное поле данных, хотя имя можно получить из поля `Employees.FirstName`? Потому что матрица подключена к источнику данных Order Details. Используя связи этого источника с другими таблицами, можно обращаться к их полям (подробнее о связях читайте в главе "Данные"). Если обратиться в нашем примере напрямую к полю `Employees.FirstName`, мы получим имя первого сотрудника.

Задайте оформление для матрицы. После этого она будет выглядеть следующим образом:

Employee	[CategoryName]	Total
[Order Details.Orders.Employees.FirstName] + " " + [Order Details.Orders.Employees.LastName]	[[Order Details.UnitPrice] * [Order Details.Quantity] * (decimal)(1 - [Order Details.Discount])]	
Total		

Если запустить отчет, мы увидим довольно большую матрицу, которая занимает 2 листа:

Employee	Beverages	Condiments	Confections	Dairy Products	Grains/Ce
Andrew Fuller	40 248,25р.	14 850,67р.	21 455,69р.	23 812,55р.	11
Anne Dodsworth	19 642,56р.	10 125,55р.	8 053,16р.	21 101,13р.	1
Janet Leverling	44 757,41р.	13 381,64р.	33 622,40р.	32 320,84р.	
Laura Callahan	17 897,85р.	14 637,66р.	21 699,91р.	21 269,47р.	
Margaret Peacock	50 308,21р.	23 314,87р.	27 768,73р.	33 549,80	
Michael Suyama	9 450,20р.	4 648,47р.	6 859,63р.		
Nancy Davolio	46 599,36р.	13 561,56р.	28 568,92		
Robert King	27 963,83р.	8 851,38р.			
Steven Buchanan	11 000,53р.				

Пример 7. Рисунки в ячейках

Ячейки матрицы наследуются от объекта "Текст" и могут отображать текстовые данные. Если вам этого недостаточно, в ячейку можно положить любой объект. Рассмотрим, как показать рисунок в матрице.

Для этого возьмем за основу пример 6. Добавим в заголовки фотографию сотрудника (поле таблицы `Employees.Photo`) и картинку категории (поле `Categories.Picture`). Сделайте следующее:

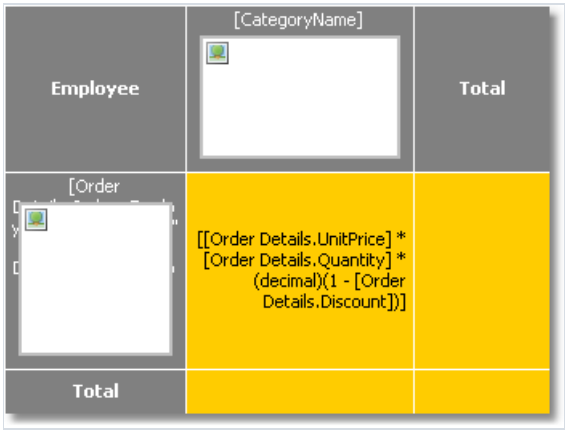
- увеличьте размер ячеек, содержащих имя сотрудника и название категории;
- добавьте в каждую из ячеек по одному объекту "Рисунок";
- чтобы показать фотографию сотрудника, укажите для объекта "Рисунок" следующее поле данных (это можно сделать в редакторе объекта):

```
Order Details.Orders.Employees.Photo
```

- чтобы показать картинку категории, укажите для объекта "Рисунок" следующее поле данных:

```
Order Details.Products.Categories.Picture
```

После этого матрица будет выглядеть так:



Если запустить отчет, мы увидим следующее:

Employee	Beverages	Condiments	Confections
Andrew Fuller 	40 248,25р.	14 850,67р.	
Anne Dodsworth 	19 642,56р.	10 125,5	
Janet Leverling 	44 7		

Пример 8. Объекты в ячейках

Используя объекты, внедренные в ячейки матрицы, можно получить разнообразные визуальные эффекты. Покажем на следующем примере, как нарисовать простую шкалу, показывающую уровень продаж сотрудника.

Матрица будет использовать источник данных MatrixDemo. Для того чтобы построить матрицу, добавьте поля источника данных MatrixDemo следующим образом:

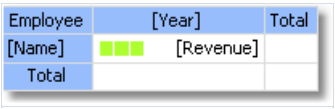
- поле `Year` добавьте в заголовок строки;
- поле `Name` добавьте в заголовок колонки;
- поле `Revenue` добавьте в ячейку матрицы.

Настройте внешний вид матрицы следующим образом:

Теперь добавим в ячейку со значением `Revenue` 3 объекта "Фигура". Эти объекты будут выполнять роль индикатора следующим образом:

- если значение в ячейке меньше 100, показывается один объект красного цвета;
- если значение в ячейке меньше 3000, показывается два объекта желтого цвета;
- если значение в ячейке больше или равно 3000, показывается три объекта зеленого цвета.

Матрица теперь выглядит так:



Чтобы управлять состоянием объектов, создадим обработчик события `BeforePrint` ячейки матрицы. Для этого выделите ячейку `Revenue`, в окне "Свойства" нажмите кнопку  и сделайте двойной щелчок на событии `BeforePrint`. FastReport создаст пустой обработчик события и переключится на закладку "Код". Напишите в обработчике следующий код:

```
private void Cell4_BeforePrint(object sender, EventArgs e)
{
  // В нашем примере ячейка имеет имя Cell4.
  // Получим значение ячейки, которое содержится в свойстве Cell4.Value.
  // Часть ячеек в нашей матрице будут пустыми, учтем это (проверка на null).
  // Значение надо привести к типу decimal, потому что исходное поле данных [MatrixDemo.Revenue]
  // имеет тип System.Decimal.
  decimal value = Cell4.Value == null ? 0 : (decimal)Cell4.Value;

  // Shape1..3 - это объекты "Фигура"
  // Включаем или выключаем объекты в зависимости от значения:
  // один объект, если значение < 100; два, если оно < 3000; три, если оно >= 3000
  Shape1.Visible = true;
  Shape2.Visible = value >= 100;
  Shape3.Visible = value >= 3000;

  // Выбираем цвет: красный < 100; желтый < 3000; зеленый >= 3000
  Color color = Color.Red;
  if (value >= 100)
    color = Color.Yellow;
  if (value >= 3000)
    color = Color.GreenYellow;

  // устанавливаем цвет для всех трех объектов
  Shape1.Fill = new SolidFill(color);
  Shape2.Fill = new SolidFill(color);
  Shape3.Fill = new SolidFill(color);
}
```

Если запустить отчет, мы увидим следующее:

Employee	1999	2000	2001	2002	Total
Andrew Fuller	3 900,00р.	2 100,00р.		1 800,00р.	7 800,00р.
Janet Leverling	6 100,00р.	3 200,00р.			9 300,00р.
Nancy Davolio	3 300,00р.	2 700,00р.	3 100,00р.	1 700,00р.	10 800,00р.
Steven Buchanan		3 999,00р.	8 100,00р.		12 099,00р.
Total	13 300,00р.	11 999,00р.	11 200,00р.	3 500,00р.	39 999,00р.

Пример 9. Заполнение матрицы вручную

Во всех рассмотренных примерах матрица заполнялась данными автоматически, потому что была привязана к источнику данных. Источник данных для матрицы указывается в свойстве `DataSource`. Хотя мы и не устанавливали значение этого свойства вручную, это происходило неявно при добавлении поля данных в матрицу.

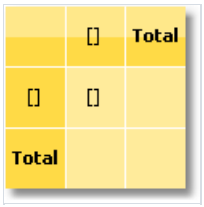
Используя скрипт, можно заполнять матрицу вручную. Для этого надо создать обработчик события `ManualBuild` матрицы. В коде обработчика надо вызывать метод AddValue для добавления значения. Покажем на примере, как создать матрицу, которая будет печатать таблицу 10x10 с значениями, расположенными по диагонали:

	1	2	3	...
1	1			
2		2		
3			3	
...				...

Выполните следующие действия:

- добавьте в отчет пустую матрицу;
- в строку, колонку и ячейку матрицы поместите любой элемент из окна "Данные". Затем вызовите редактор выражения, сделав двойной щелчок на элементе матрицы, и очистите выражение;
- очистите свойство `DataSource` матрицы.

В результате матрица будет выглядеть следующим образом:



Теперь создайте обработчик события `ManualBuild`. Для этого выделите матрицу, в окне "Свойства" нажмите кнопку  и сделайте двойной щелчок на событии `ManualBuild`. FastReport создаст пустой обработчик события. Напишите в нем следующий код:

```
private void Matrix1_ManualBuild(object sender, EventArgs e)
{
  // В нашей матрице 1 измерение в строке, колонке и ячейке данных.
  // Создадим 3 массива типа object[], каждый с 1 элементом (по количеству измерений).
  object[] columnValues = new object[1];
  object[] rowValues = new object[1];
  object[] cellValues = new object[1];

  for (int i = 1; i <= 10; i++)
  {
    // Заполняем массивы
    columnValues[0] = i;
    rowValues[0] = i;
    cellValues[0] = i;

    // Добавляем данные в матрицу
    Matrix1.AddValue(columnValues, rowValues, cellValues);
  }
}
```

В обработчике необходимо добавить нужные данные в матрицу с помощью метода `AddValue` объекта "Матрица". Этот метод имеет три параметра, каждый из которых является массивом значений типа `object`. Первый параметр – это значения колонки, второй – значения строки, третий – значения ячеек. Заметьте – количество значений в каждом массиве должно соответствовать настройке объекта! В нашем случае объект имеет по одному уровню в заголовках колонок, строк и ячеек – соответственно, мы передаем в `AddValue` по одному значению для колонок, строк и ячеек.

Если запустить отчет на выполнение, мы увидим следующее:

	1	2	3	4	5	6	7	8	9	10	**Total**
1	1										1
2		2									2
3			3								3
4				4							4
5					5						5
6						6					6
7							7				7
8								8			8
9									9		9
10										10	10
Total	1	2	3	4	5	6	7	8	9	10	**55**

Покажем на примере, как добавить в матрицу значение "21" на пересечении колонки 7 и строки 3. Для этого измените код следующим образом:

```
private void Matrix1_ManualBuild(object sender, EventArgs e)
{
  object[] columnValues = new object[1];
  object[] rowValues = new object[1];
  object[] cellValues = new object[1];

  for (int i = 1; i <= 10; i++)
  {
    columnValues[0] = i;
    rowValues[0] = i;
    cellValues[0] = i;

    Matrix1.AddValue(columnValues, rowValues, cellValues);
  }
  columnValues[0] = 7;
  rowValues[0] = 3;
  cellValues[0] = 21;
  Matrix1.AddValue(columnValues, rowValues, cellValues);
}
```

В результате получится следующее:

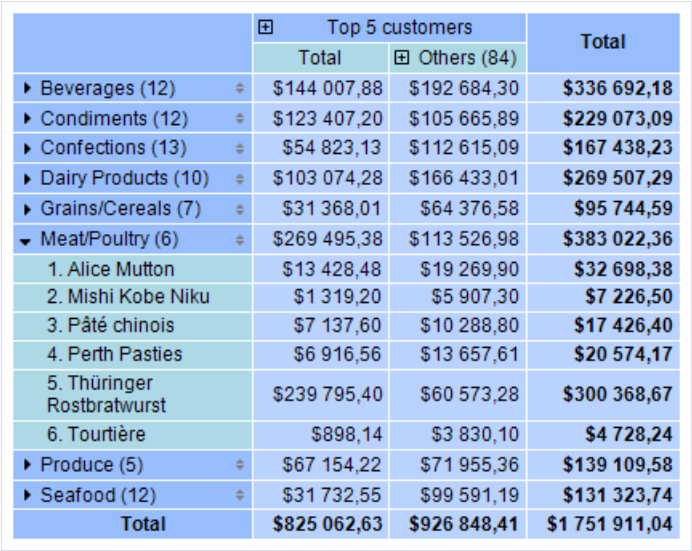
	1	2	3	4	5	6	7	8	9	10	**Total**
1	1										1
2		2									2
3			3				21				24
4				4							4
5					5						5
6						6					6
7							7				7
8								8			8
9									9		9
10										10	10
Total	1	2	3	4	5	6	28	8	9	10	**76**

Как видно, матрица автоматически рассчитывает итоги по строкам и колонкам.

Вы можете использовать обработчик события `ManualBuild` для матрицы, которая подключена к данным. В этом случае сначала вызывается обработчик события, затем происходит заполнение матрицы данными из источника.

Объект "Улучшенная матрица"

Данный объект, как и объект "Матрица", позволяет строить сводные отчеты.



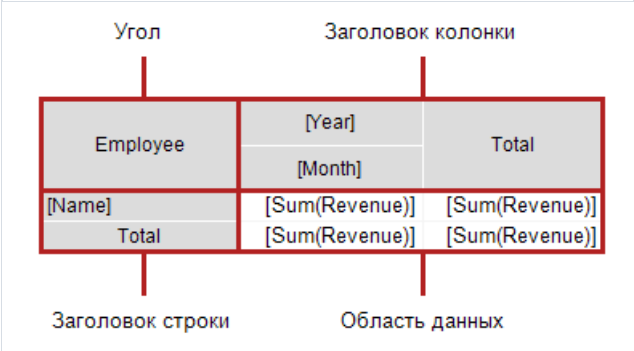
	Top 5 customers		Total
	Total	Others (84)	
Beverages (12)	$144 007,88	$192 684,30	**$336 692,18**
Condiments (12)	$123 407,20	$105 665,89	**$229 073,09**
Confections (13)	$54 823,13	$112 615,09	**$167 438,23**
Dairy Products (10)	$103 074,28	$166 433,01	**$269 507,29**
Grains/Cereals (7)	$31 368,01	$64 376,58	**$95 744,59**
Meat/Poultry (6)	$269 495,38	$113 526,98	**$383 022,36**
1. Alice Mutton	$13 428,48	$19 269,90	**$32 698,38**
2. Mishi Kobe Niku	$1 319,20	$5 907,30	**$7 226,50**
3. Pâté chinois	$7 137,60	$10 288,80	**$17 426,40**
4. Perth Pasties	$6 916,56	$13 657,61	**$20 574,17**
5. Thüringer Rostbratwurst	$239 795,40	$60 573,28	**$300 368,67**
6. Tourtière	$898,14	$3 830,10	**$4 728,24**
Produce (5)	$67 154,22	$71 955,36	**$139 109,58**
Seafood (12)	$31 732,55	$99 591,19	**$131 323,74**
Total	**$825 062,63**	**$926 848,41**	**$1 751 911,04**

Вот список его ключевых особенностей:

- заголовки строк и колонок могут содержать группы и простые элементы в произвольном порядке. Это позволяет строить асимметричные отчеты;
- кнопки сворачивания позволяют интерактивно управлять видимостью отдельных элементов;
- кнопки сортировки позволяют интерактивно сортировать матрицу по выбранным значениям, в том числе по значениям итогов;
- группировка Top N позволяет отобразить N значений в заголовке, а остальные значения сгруппировать в отдельный элемент с возможностью разворачивания;
- вывод заголовков матрицы в ступенчатом виде;
- сортировка заголовков по значениям итогов;
- широкий набор агрегатных функций;
- поддержка пользовательских агрегатных функций;
- широкий набор специальных функций, позволяющих получить значения итогов, соседних ячеек, а также функции для расчета процентов;
- поддержка объектов "Искрографик" и "Индикатор прогресса" в ячейках данных.

Структура матрицы

Объект "Улучшенная матрица" состоит из следующих элементов:



Угол

Ячейки, расположенные в углу матрицы, могут содержать произвольную информацию. Вы также можете разбивать/объединять их по своему усмотрению.

Заголовок

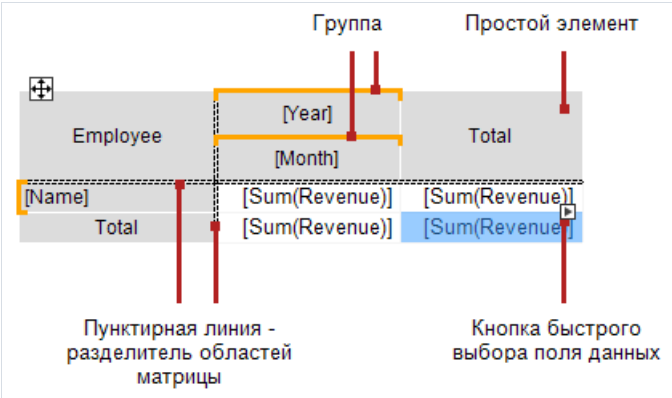
Заголовок матрицы может содержать два типа элементов:

- простой элемент: отображает статическую информацию, например, текст "Итого".
- группа: отображает список значений, сгруппированных по определенному критерию.

Заголовок имеет древовидную структуру. Корневой элемент невидимый, он содержит видимые элементы первого уровня.

Допускается произвольное расположение элементов; например, заголовок может не иметь ни одной группы, либо иметь несколько, расположенных рядом. Итоги также могут располагаться произвольным образом.

В режиме дизайна матрица отображает визуальные подсказки в области заголовков:



В данном случае структура заголовков следующая:

```
Заголовок строки
- Группа "Name"
- Элемент "Total"

Заголовок колонки
- Группа "Year"
  - Группа "Month"
- Элемент "Total"
```

См. раздел Настройка заголовка для дополнительной информации.

Область данных

Ячейки в области данных, как правило, содержат агрегатную функцию. См. раздел Настройка области данных.

Настройка матрицы

Настройка структуры

Для настройки матрицы проделайте следующие шаги:

1. Настройте заголовки (см. раздел Настройка заголовка).
2. Настройте ячейки данных (см. раздел Настройка области данных).
3. Добавьте итоги (см. раздел Настройка заголовка). Этот шаг лучше выполнять в последнюю очередь, чтобы сэкономить время на настройку новых ячеек данных.

> Матрица должна быть подключена к источнику данных - за это отвечает свойство `DataSource`. Как правило, это свойство устанавливается автоматически на этапе настройки заголовка и ячеек.

Контекстное меню

Чтобы вызвать контекстное меню, выделите любой элемент матрицы, затем щелкните правой кнопкой мыши на области в левом верхнем углу матрицы:

В меню доступны следующие команды:

- "Стиль" - выберите один из предустановленных стилей;
- "Поменять местами колонки и строки" - позволяет быстро поменять местами колонки и строки в матрице;
- "Повторять заголовки на новой странице" - если матрица занимает несколько страниц, на каждой странице будут напечатаны заголовки колонок и строк.

Настройки, доступные в окне "Свойства"

В окне "Свойства" доступны следующие свойства, специфичные для объекта "Улучшенная матрица":

Свойство	Значение	Описание
DataRowPriority	Rows	Приоритет заголовков при обращении к полям БД из ячеек данных. См. раздел Свойства, доступные из ячеек данных.
DataSource		Источник данных.
EvenStylePriority	Rows	Приоритет номера строк или колонок для работы свойства `EvenStyle`.
Filter		Выражение фильтрации данных. См. раздел Фильтрация данных.

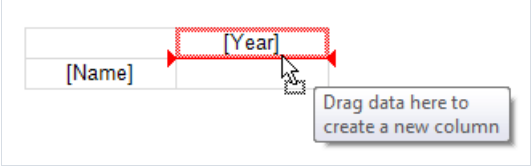
Свойство	Значение	Описание
Layout	AcrossThenDown	См. раздел Развертка таблицы.
PrintIfEmpty	True	Печатать матрицу, если она пустая.
RepeatHeaders	True	Повторять заголовки на новой странице.
ResetDataOnRun	False	Сбрасывать данные при каждом запуске отчета. По умолчанию матрица не перестраивается при интерактивных операциях (см. раздел Интерактивные возможности).
Style		Стиль матрицы.
WrappedGap	0	Зазор между частями матрицы при режиме развертки `Layout = Wrapped`.

Настройка заголовка

Добавление элемента

Добавить элемент в заголовок можно двумя способами:

- перетаскиванием поля из окна "Данные". При перетаскивании будет показано, в какую часть заголовка будет добавлен новый элемент:



- с помощью контекстного меню заголовка. Выделите элемент, рядом с которым вы хотите добавить новый элемент, и выберите в контекстном меню одну из команд "Добавить новый элемент". Будет добавлен пустой новый элемент;
- также с помощью контекстного меню можно добавить элемент "Итог" (перед или после выбранного элемента). Будет добавлен новый элемент с текстом "Итого" (текст зависит от текущей локализации).

> Добавление итога эквивалентно добавлению пустого элемента и редактированию его текста.

> При добавлении поля БД из окна "Данные" в пустую матрицу автоматически настраивается ее свойство `DataSource`.

Удаление элемента

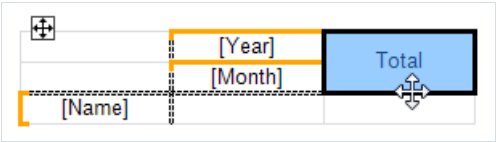
Удалить элемент можно, выбрав в контекстном меню пункт "Удалить". Вы можете удалить только выбранный элемент, или дерево элементов (выбранный элемент и все его дочерние элементы).

Также можно удалить элемент нажав клавишу `Delete`. В этом случае удаляется только выбранный элемент.

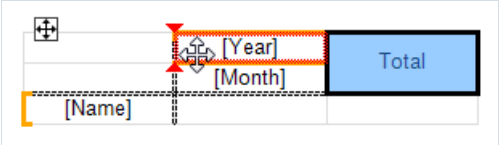
> Элемент со значком замка описанным способом удалить нельзя. См. раздел Группировка TopN.

Перемещение элемента

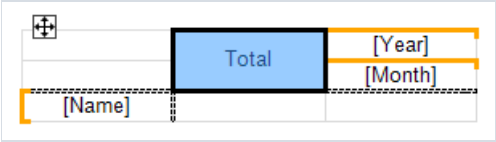
Чтобы переместить элемент на новое место, выделите его с помощью левой кнопки мыши. Элемент будет выделен толстой черной рамкой:



Схватите элемент за рамку и переместите на новое место. При перемещении будет показано, в какую часть заголовка будет перемещен элемент:



Отпустите кнопку мыши, и элемент будет перемещен на новое место:



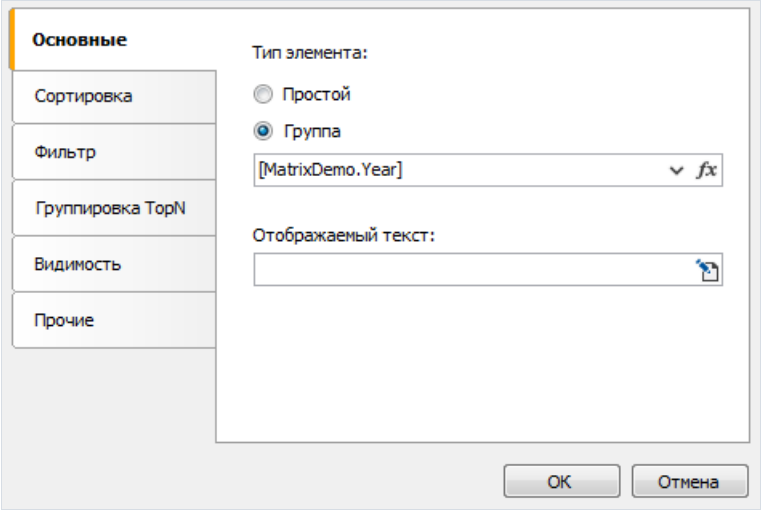
> Если элемент имеет дочерние элементы, они будут перемещены вместе с основным элементом, если это возможно.

Редактирование элемента

Для вызова редактора элемента сделайте на нем двойной щелчок левой кнопкой мыши, или выберите пункт "Редактировать..." в контекстном меню. Также вызвать редактор можно клавишей `Enter`.

Группировка

Как было описано выше, заголовок матрицы может содержать элементы двух типов - группа и простой элемент. Тип элемента задается в редакторе заголовка:



> Простой элемент можно сделать группой и наоборот. Для простого элемента недоступна часть настроек в окне редактора.

Группа позволяет вывести список значений, сгруппированных по условию. В примере выше указано условие `[MatrixDemo.Year]` - это означает, что элемент выведет список лет. Список будет содержать не повторяющиеся значения; одинаковые значения будут сгруппированы.

По умолчанию группа отображает значения, указанные в условии группировки. Например, если указано условие `[MatrixDemo.Month]`, будут показаны номера месяцев. Свойство "Отображаемый текст" позволяет показать другое значение, например:

```
[MonthName([MatrixDemo.Month])]
```

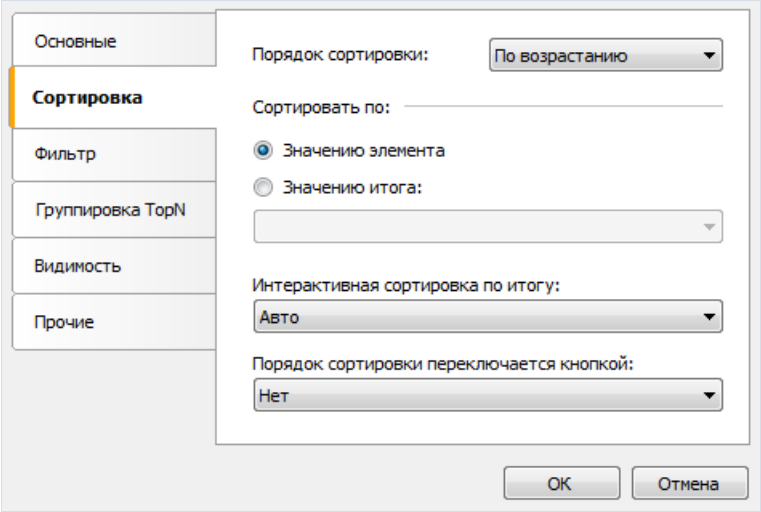
выведет название месяца вместо его номера.

В этом свойстве можно обращаться к специальным функциям матрицы (см. раздел Свойства, доступные из ячеек заголовка), например:

```
[Matrix1.RowNo]. [MatrixDemo.Name]
```

Сортировка

Настройки сортировки выводимых значений представлены на одноименной вкладке редактора заголовка. Настройки активны, если выбран тип элемента - "Группа":



Описание настроек:

"Порядок сортировки" - задает порядок сортировки - по возрастанию, по убыванию или без сортировки.

"Сортировать по" - выберите один из двух вариантов сортировки: по выводимым значениям, либо по значению итога. На картинках ниже показано, как происходит сортировка элемента "Год":

По значению элемента:

	2011	2012	2013	2014	2015
Andrew Fuller	$3,900.00	$2,100.00			$1,800.00
Janet Leverling	$6,100.00	$3,200.00			
Nancy Davolio	$3,300.00	$2,700.00	$3,100.00		$1,700.00
Steven Buchanan			$3,999.00	$8,100.00	
Total	$13,300.00	$8,000.00	$7,099.00	$8,100.00	$3,500.00

По значению итога:

	2015	2013	2012	2014	2011
Andrew Fuller	$1,800.00		$2,100.00		$3,900.00
Janet Leverling			$3,200.00		$6,100.00
Nancy Davolio	$1,700.00	$3,100.00	$2,700.00		$3,300.00
Steven Buchanan		$3,999.00		$8,100.00	
Total	$3,500.00	$7,099.00	$8,000.00	$8,100.00	$13,300.00

"Интерактивная сортировка по итогу" - определяет, как сортировать значения заголовка, если активна интерактивная сортировка (см. раздел Интерактивная сортировка). Возможные варианты - "Нет", "Авто" и имя итога.

"Порядок сортировки переключается кнопкой" - здесь указывается имя кнопки типа `MatrixSortButton`, которая находится в углу матрицы или за ее пределами. При нажатии на кнопку в окне просмотра отчета изменяется порядок сортировки и матрица обновляется.

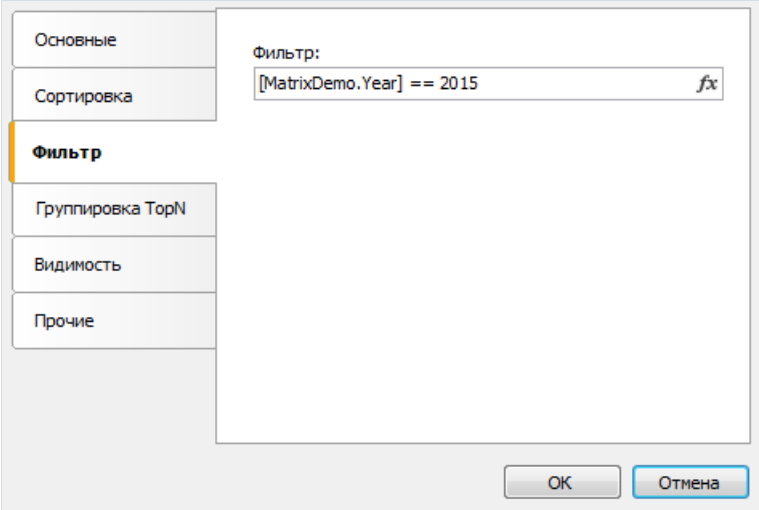
Фильтрация данных

Есть два способа фильтрации данных, отображаемых в матрице.

Способ 1: фильтрация с помощью свойства матрицы `Filter`. Для этого выделите объект "Матрица" и в окне "Свойства" заполните свойство `Filter`:

```
[MatrixDemo.Year] == 2015
```

Способ 2: фильтрация в редакторе заголовка, с помощью аналогичного выражения:



В обоих случаях выражение фильтрации возвращает тип `bool`.

Разница между двумя описанными способами заключается в том, какой объем данных попадает в матрицу. В первом случае отбрасываются все значения, кроме тех, где год равен 2015. Результат может выглядеть следующим образом:

	2015
Andrew Fuller	$1,800.00
Nancy Davolio	$1,700.00

Во втором случае фильтруются значения конкретного элемента в заголовке. В отличие от предыдущего варианта, результат может содержать пустые значения:

	2015
Andrew Fuller	$1,800.00
Janet Leverling	
Nancy Davolio	$1,700.00
Steven Buchanan	

Группировка TopN

Если количество значений в группе заголовка велико, это приведет к генерации чрезмерного количества страниц отчета. Группировка TopN позволяет отобразить N первых значений, а остальные значения показать в свернутом виде:

	Top 5 customers						Total
	Lonesome Pine Restaurant	Rattlesnake Canyon Grocery	Save-a-lot Markets	QUICK-Stop	Ernst Handel	⊞ Others (84)	
▸ Beverages (12)	$2 708,00	$26 876,15	$65 498,00	$36 216,43	$12 709,30	$192 684,30	**$336 692,18**
▸ Condiments (12)	$82 480,00	$6 075,20	$11 567,00	$9 214,94	$14 070,06	$105 665,89	**$229 073,09**
▸ Confections (13)	$549,00	$10 947,21	$11 981,07	$18 530,09	$12 815,76	$112 615,09	**$167 438,23**
▸ Dairy Products (10)	$815,00	$42 854,87	$21 107,10	$13 800,85	$24 496,46	$166 433,01	**$269 507,29**
▸ Grains/Cereals (7)	$190,00	$4 831,31	$8 298,10	$5 310,90	$12 737,70	$64 376,58	**$95 744,59**
▸ Meat/Poultry (6)	$140 098,40	$83 657,28	$27 659,18	$9 754,96	$8 325,56	$113 526,98	**$383 022,36**
▸ Produce (5)	$16 379,20	$13 836,05	$16 387,90	$8 081,40	$12 469,67	$71 955,36	**$139 109,58**
▸ Seafood (12)	$625,00	$884,73	$13 604,60	$9 367,74	$7 250,48	$99 591,19	**$131 323,74**
Total	**$243 844,60**	**$189 962,80**	**$176 102,95**	**$110 277,31**	**$104 874,98**	**$926 848,41**	**$1 751 911,04**

Принцип работы

Механизм TopN использует четыре элемента для отображения данных:

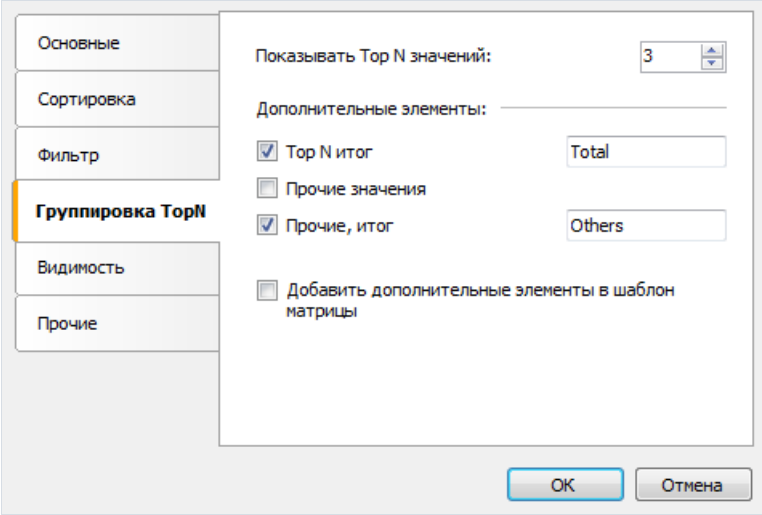
1. Группа TopN - исходная группа, содержащая большое количество значений.
2. Итог TopN, в котором отображается итог по значениям TopN.
3. Группа "Остальные", которая отображает значения, не вошедшие в TopN.
4. Итог группы "Остальные".

Если в исходной группе меньше значений, чем указано в свойстве `TopN.Count`, она выводится обычным образом, без задействования группировки TopN. Иначе происходит следующее:

- в основной группе оставляется N значений;
- остальные значения переносятся в группу "Остальные";
- производится агрегация данных в основной группе и в группе "Остальные";
- полученные значения выводятся в итоге TopN и итоге группы "Остальные".

Настройка

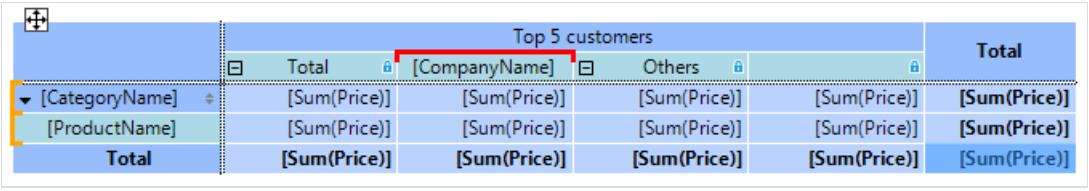
Настройка TopN выполняется для основной группы. Для этого сделайте двойной щелчок на элементе или выберите пункт "Редактировать..." из контекстного меню.



Возможны два варианта работы с дополнительными элементами:

- элементы "Итог TopN", "Остальные", "Итог остальных" создаются автоматически при построении матрицы. Их визуальное оформление копируется из основного элемента. Вы можете управлять видимостью элементов, а также указать заголовки для элементов-итогов. Другие возможности настройки внешнего вида отсутствуют;
- вышеуказанные элементы добавляются в шаблон матрицы. Это дает возможность полной настройки внешнего вида, а также изменения порядка элементов. Вы можете добавить кнопки сворачивания для интерактивного управления видимостью отдельных элементов.

Ниже показано, как выглядит шаблон матрицы при добавлении в него дополнительных элементов:



В данном случае основная группа подсвечивается красным цветом, а дополнительные элементы помечаются значком замка. Удалить их можно в окне редактора основного элемента, сняв флажок "Добавить дополнительные элементы в шаблон матрицы".

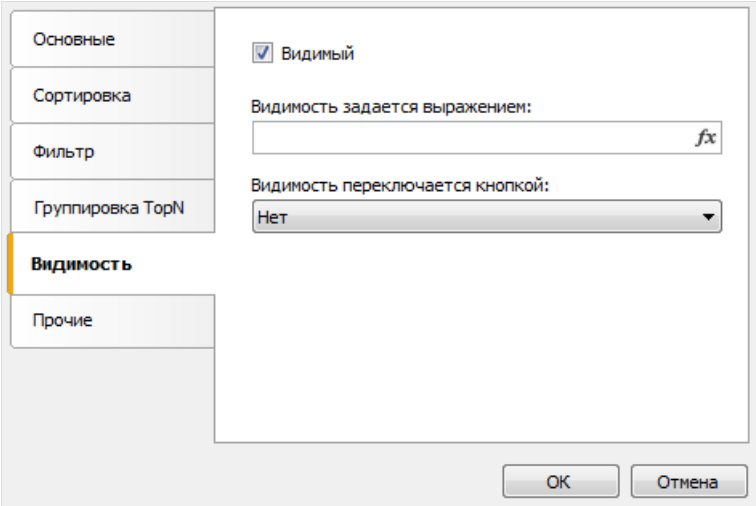
TopN, BottomN, FirstN, LastN

Механизм TopN использует первые N значений в исходной группе. От того, как была отсортирована исходная группа, зависит смысл получаемых значений:

- группа отсортирована по значению заголовка: отображается N первых (сортировка по возрастанию) или N последних значений (сортировка по убыванию);
- группа отсортирована по значению итога: отображается N наибольших (сортировка по убыванию) или N наименьших значений (сортировка по возрастанию).

Видимость элемента

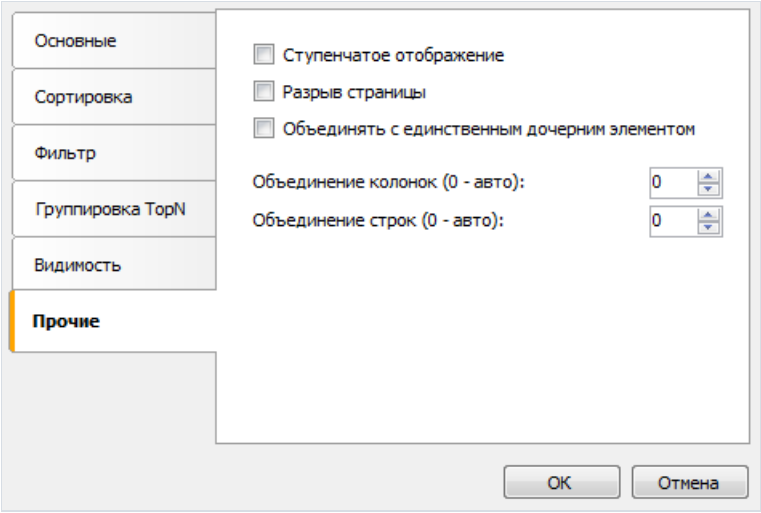
На закладке "Видимость" вы можете управлять видимостью элемента:



- "Видимый" - устанавливает начальную видимость.
- "Видимость задается выражением" - выражение, возвращающее тип `bool`, которое задает видимость. Если выражение не пустое, его значение используется вместо опции "Видимый".
- "Видимость переключается кнопкой" - имя кнопки типа `MatrixCollapseButton`, которая управляет видимостью данного элемента. См. раздел Сворачивание/разворачивание элементов.

Прочие настройки

На закладке "Прочие" редактора заголовка сосредоточены прочие настройки:



Опция "Ступенчатое отображение" переключает отображение вложенных элементов между блочным расположением (по умолчанию) и ступенчатым. Рассмотрим пример матрицы с вложенным заголовком строк (Year, Month). Блочное расположение выглядит следующим образом:

		[Name]
[Year]	[Month]	[Sum(Revenue)]
Total		[Sum(Revenue)]

Готовый отчет выглядит следующим образом:

		Andrew Fuller	Janet Leverling
2011	2		
	10	$1,900.00	$3,000.00
	11	$2,000.00	$3,100.00
	12		
2012	1		
	2	$2,100.00	
	3		$3,200.00
2013	1		
	2		
	3		
2014	1		
	2		
2015	1	$1,800.00	
Total		$7,800.00	$9,300.00

Если для элемента "Year" включить опцию "Ступенчатое отображение", расположение элементов поменяется следующим образом:

	[Name]
[Year]	[Sum(Revenue)]
[Month	[Sum(Revenue)]
Total	[Sum(Revenue)]

Готовый отчет будет выглядеть следующим образом:

	Andrew Fuller	Janet Leverling
2011	$3,900.00	$6,100.00
2		
10	$1,900.00	$3,000.00
11	$2,000.00	$3,100.00
12		
2012	$2,100.00	$3,200.00
1		
2	$2,100.00	
3		$3,200.00
2013		
1		
2		
3		
2014		
1		
2		
2015	$1,800.00	
1	$1,800.00	
Total	$7,800.00	$9,300.00

При включении опции "Ступенчатое отображение" у вложенного элемента меняется свойство `Padding.Left`, отвечающее за отступ текста. Вы можете настроить это значение в окне "Свойства".

Эту опцию можно использовать для любого элемента, имеющего вложенные элементы.

Опция "Разрыв страницы" вставляет новую страницу перед печатью элемента. Перед первым элементом новая страница не вставляется.

Опция "Объединять с единственным дочерним элементом" применяется в случае динамически сворачиваемых заголовков (см. раздел Сворачивание/разворачивание элементов). Опция позволяет скрыть элемент "Итого". Ниже приведен вид готового отчета с отключенной опцией (по умолчанию).

		Andrew Fuller
⊞ 2011	Total	$3,900.00
⊞ 2012	Total	$2,100.00
⊟ 2013	1	
	2	
	3	
	Total	
⊞ 2014	Total	
⊞ 2015	Total	$1,800.00
Total		$7,800.00

и с включенной, обратите внимание на вывод элемента "Год":

		Andrew Fuller
⊞ 2011		$3,900.00
⊞ 2012		$2,100.00
⊟ 2013	1	
	2	
	3	
	Total	
⊞ 2014		
⊞ 2015		$1,800.00
Total		$7,800.00

Опции "Объединение строк" и "Объединение колонок" позволяют объединять ячейки в строках и колонках при печати элемента. По умолчанию элемент управляет этими параметрами самостоятельно.

Свойства, доступные из ячеек заголовка

Матрица имеет набор свойств, которые можно использовать при печати ячеек заголовка. Для обращения к свойству используется имя матрицы:

```
[Matrix1.RowNo]
```

Свойство	Возвращаемое значение	Описание
RowNo	int	Порядковый номер элемента заголовка.
ItemCount	int	Количество дочерних элементов в текущем элементе заголовка.

Эти свойства можно использовать в поле "Отображаемый текст" ячейки заголовка, например:

```
[Matrix1.RowNo].[MatrixDemo.Name]
```

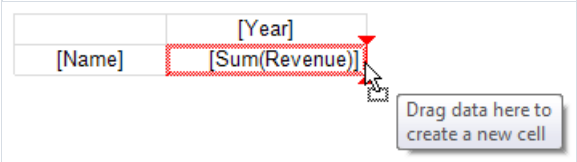
выведет текст

```
1. Andrew Fuller
```

Настройка области данных

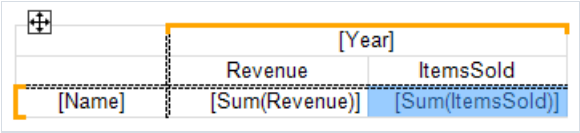
Добавление элемента

Добавить элемент можно путем перетаскивания поля БД из окна "Данные". При перетаскивании будет показано, в какую часть области данных будет добавлен новый элемент:



> При добавлении элемента в пустую матрицу автоматически настраивается ее свойство `DataSource`.

При добавлении нового элемента рядом с существующим будет также добавлен его заголовок:



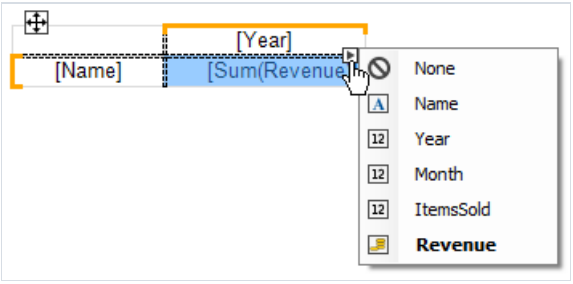
Удаление элемента

Очистить текст элемента можно, нажав клавишу `Delete` или выбрав в меню элемента пункт "Очистить". Сам элемент данных удалить нельзя, для этого надо удалить соответствующий элемент в заголовке матрицы. Так, в примере выше, для удаления элемента `[Sum(Revenue)]` из области данных надо удалить элемент "Revenue" из заголовка.

Редактирование элемента

Для редактирования текста элемента сделайте двойной щелчок мыши - это вызовет окно редактора текста. Также редактор можно вызвать, нажав клавишу `Enter`.

Быстро выбрать поле БД, отображаемое в элементе, можно с помощью кнопки быстрого выбора (smarttag), которая отображается, когда курсор мыши находится внутри элемента:



Также с помощью контекстного меню элемента можно выполнить следующие операции:

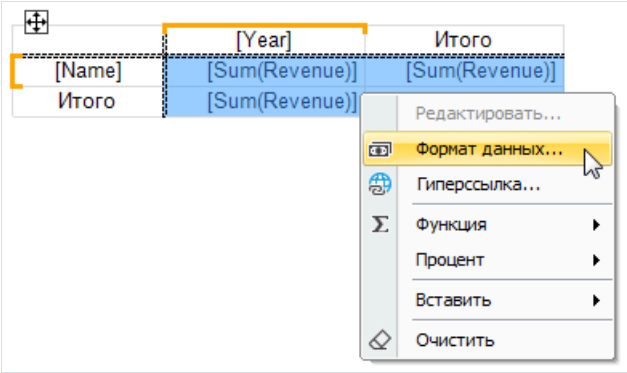
- вызвать редактор элемента;
- настроить форматирование данных;
- сменить тип агрегатной функции (см. раздел Агрегатные функции);
- добавить расчет процентов;
- вставить в ячейку индикатор прогресса или искрографик.

Редактирование нескольких элементов

Для массового редактирования ячеек можно выделить всю область данных или несколько ячеек. Это можно сделать с помощью клавиши `Shift` или с помощью мыши:

- выделите начальную ячейку;
- нажмите левую кнопку мыши и, не отпуская ее, двигайте мышь, чтобы выделить группу ячеек.

Для выделенных ячеек можно вызвать контекстное меню или воспользоваться кнопкой быстрого выбора поля БД:



Агрегатные функции

Агрегатные функции используются в ячейках данных для агрегации значений ячеек и для расчета итогов. Вызов агрегатной функции имеет следующий вид:

```
[Функция(Выражение)]
```

> Квадратные скобки нужны для обозначения выражений в тексте ячейки. Вы можете использовать несколько агрегатных функций в одной ячейке вперемешку с обычным текстом.

Выражение, как правило, является полем источника данных. Пример использования агрегатной функции:

```
[Sum([MatrixDemo.Revenue])]
```

Ниже приведен список агрегатных функций:

Функция	Описание
Sum	Возвращает сумму значений.
Min	Возвращает минимальное значение.
Max	Возвращает максимальное значение.
Avg	Возвращает среднее значение.
Count	Возвращает количество значений.
CountDistinct	Возвращает количество разных (уникальных) значений.
StDev	Возвращает стандартное отклонение по выборке.
StDevP	Возвращает стандартное отклонение по популяции.
Var	Возвращает дисперсию по выборке.
VarP	Возвращает дисперсию по популяции.
First	Возвращает первое значение.
Last	Возвращает последнее значение.
ValuesList	Возвращает список всех значений, попавших в ячейку. Этот агрегат применяется для совместной работы с объектами "Диаграмма" и "Искрографик".
_имя	Пользовательская агрегатная функция, определенная в коде отчета.

Пользовательская функция имеет имя, начинающееся со знака подчеркивания. Ее код должен быть размещен в теле главного класса отчета, `ReportScript`. Функция определена следующим образом:

```
object _FuncName(List<dynamic> l)
```

Пример пользовательской функции `_Sum`:

```
public class ReportScript
{
    public object _Sum(List<dynamic> l)
    {
        dynamic value = 0;
        foreach (dynamic v in l)
            value += v;
        return value;
    }
}
```

Специальные функции

Специальные функции могут использоваться в ячейке данных матрицы. Они позволяют получить значение другой ячейки, расположенной в той же строке или колонке.

GrandColumnTotal

Возвращает значение общего итога по колонке.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.

Примеры:

```
GrandColumnTotal(Sum([MatrixDemo.Revenue]))
Sum([MatrixDemo.Revenue]) / GrandColumnTotal()
```

GrandRowTotal

Возвращает значение общего итога по строке.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.

Примеры:

```
GrandRowTotal(Sum([MatrixDemo.Revenue]))
Sum([MatrixDemo.Revenue]) / GrandRowTotal()
```

GrandTotal

Возвращает значение общего итога.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.

Примеры:

```
GrandTotal(Sum([MatrixDemo.Revenue]))
Sum([MatrixDemo.Revenue]) / GrandTotal()
```

ColumnTotal

Возвращает значение итога по колонке для текущей группы.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.

Примеры:

```
ColumnTotal(Sum([MatrixDemo.Revenue]))
Sum([MatrixDemo.Revenue]) / ColumnTotal()
```

RowTotal

Возвращает значение итога по строке для текущей группы.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.

Примеры:

```
RowTotal(Sum([MatrixDemo.Revenue]))
Sum([MatrixDemo.Revenue]) / RowTotal()
```

ColumnMaxValue

Возвращает максимальное значение итога по колонке для текущей группы.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.

Примеры:

```
ColumnMaxValue(Sum([MatrixDemo.Revenue]))
Sum([MatrixDemo.Revenue]) / ColumnMaxValue()
```

ColumnMinValue

Возвращает минимальное значение итога по колонке для текущей группы.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.

Примеры:

```
ColumnMinValue(Sum([MatrixDemo.Revenue]))
Sum([MatrixDemo.Revenue]) / ColumnMinValue()
```

RowMaxValue

Возвращает максимальное значение итога по строке для текущей группы.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.

Примеры:

```
RowMaxValue(Sum([MatrixDemo.Revenue]))
Sum([MatrixDemo.Revenue]) / RowMaxValue()
```

RowMinValue

Возвращает минимальное значение итога по строке для текущей группы.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.

Примеры:

```
RowMinValue(Sum([MatrixDemo.Revenue]))
Sum([MatrixDemo.Revenue]) / RowMinValue()
```

FirstColumn

Возвращает значение первой ячейки в колонке.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.

Параметр	Описание
`useInteractiveSort=false`	(необязательно) Учитывать результаты интерактивной сортировки, при которой порядок элементов может меняться.
`useThisGroup=true`	(необязательно) Использовать группу на том же уровне для поиска значения.

Возвращаемое значение зависит от того, в каком месте используется функция:

- ячейка итога: возвращает значение первой ячейки в группе, если параметр `useThisGroup` равен `true`, иначе возвращает значение первого итога;
- ячейка группы: возвращает значение первой ячейки в группе.

Примеры:

```
FirstColumn(Sum([MatrixDemo.Revenue]))
FirstColumn(Sum([MatrixDemo.Revenue]), true)
Sum([MatrixDemo.Revenue]) / FirstColumn()
```

FirstRow

Возвращает значение первой ячейки в строке.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.
`useInteractiveSort=false`	(необязательно) Учитывать результаты интерактивной сортировки, при которой порядок элементов может меняться.
`useThisGroup=true`	(необязательно) Использовать группу на том же уровне для поиска значения.

Возвращаемое значение зависит от того, в каком месте используется функция:

- ячейка итога: возвращает значение первой ячейки в группе, если параметр `useThisGroup` равен `true`, иначе возвращает значение первого итога;
- ячейка группы: возвращает значение первой ячейки в группе.

Примеры:

```
FirstRow(Sum([MatrixDemo.Revenue]))
FirstRow(Sum([MatrixDemo.Revenue]), true)
Sum([MatrixDemo.Revenue]) / FirstRow()
```

LastColumn

Возвращает значение последней ячейки в колонке.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.
`useInteractiveSort=false`	(необязательно) Учитывать результаты интерактивной сортировки, при которой порядок элементов может меняться.
`useThisGroup=true`	(необязательно) Использовать группу на том же уровне для поиска значения.

Возвращаемое значение зависит от того, в каком месте используется функция:

- ячейка итога: возвращает значение последней ячейки в группе, если параметр `useThisGroup` равен `true`, иначе возвращает значение последнего итога;
- ячейка группы: возвращает значение последней ячейки в группе.

Примеры:

```
LastColumn(Sum([MatrixDemo.Revenue]))
LastColumn(Sum([MatrixDemo.Revenue]), true)
Sum([MatrixDemo.Revenue]) / LastColumn()
```

LastRow

Возвращает значение последней ячейки в строке.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.
`useInteractiveSort=false`	(необязательно) Учитывать результаты интерактивной сортировки, при которой порядок элементов может меняться.
`useThisGroup=true`	(необязательно) Использовать группу на том же уровне для поиска значения.

Возвращаемое значение зависит от того, в каком месте используется функция:

- ячейка итога: возвращает значение последней ячейки в группе, если параметр `useThisGroup` равен `true`, иначе возвращает значение последнего итога;
- ячейка группы: возвращает значение последней ячейки в группе.

Примеры:

```
LastRow(Sum([MatrixDemo.Revenue]))
LastRow(Sum([MatrixDemo.Revenue]), true)
Sum([MatrixDemo.Revenue]) / LastRow()
```

PreviousColumn

Возвращает значение предыдущей ячейки в колонке.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.
`useInteractiveSort=false`	(необязательно) Учитывать результаты интерактивной сортировки, при которой порядок элементов может меняться.
`useThisGroup=true`	(необязательно) Использовать группу на том же уровне для поиска значения.

Возвращаемое значение зависит от того, в каком месте используется функция:

- ячейка итога: возвращает значение предыдущей ячейки в группе, если параметр `useThisGroup` равен `true`, иначе возвращает значение предыдущего итога;
- ячейка группы: возвращает значение предыдущей ячейки в группе.

Примеры:

```
PreviousColumn(Sum([MatrixDemo.Revenue]))
PreviousColumn(Sum([MatrixDemo.Revenue]), true)
Sum([MatrixDemo.Revenue]) / PreviousColumn()
```

PreviousRow

Возвращает значение предыдущей ячейки в строке.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.
`useInteractiveSort=false`	(необязательно) Учитывать результаты интерактивной сортировки, при которой порядок элементов может меняться.
`useThisGroup=true`	(необязательно) Использовать группу на том же уровне для поиска значения.

Возвращаемое значение зависит от того, в каком месте используется функция:

- ячейка итога: возвращает значение предыдущей ячейки в группе, если параметр `useThisGroup` равен `true`, иначе возвращает значение предыдущего итога;
- ячейка группы: возвращает значение предыдущей ячейки в группе.

Примеры:

```
PreviousRow(Sum([MatrixDemo.Revenue]))
PreviousRow(Sum([MatrixDemo.Revenue]), true)
Sum([MatrixDemo.Revenue]) / PreviousRow()
```

NextColumn

Возвращает значение следующей ячейки в колонке.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.
`useInteractiveSort=false`	(необязательно) Учитывать результаты интерактивной сортировки, при которой порядок элементов может меняться.
`useThisGroup=true`	(необязательно) Использовать группу на том же уровне для поиска значения.

Возвращаемое значение зависит от того, в каком месте используется функция:

- ячейка итога: возвращает значение следующей ячейки в группе, если параметр `useThisGroup` равен `true`, иначе возвращает значение следующего итога;
- ячейка группы: возвращает значение следующей ячейки в группе.

Примеры:

```
NextColumn(Sum([MatrixDemo.Revenue]))
NextColumn(Sum([MatrixDemo.Revenue]), true)
Sum([MatrixDemo.Revenue]) / NextColumn()
```

NextRow

Возвращает значение следующей ячейки в строке.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.
`useInteractiveSort=false`	(необязательно) Учитывать результаты интерактивной сортировки, при которой порядок элементов может меняться.
`useThisGroup=true`	(необязательно) Использовать группу на том же уровне для поиска значения.

Возвращаемое значение зависит от того, в каком месте используется функция:

- ячейка итога: возвращает значение следующей ячейки в группе, если параметр `useThisGroup` равен `true`, иначе возвращает значение следующего итога;
- ячейка группы: возвращает значение следующей ячейки в группе.

Примеры:

```
NextRow(Sum([MatrixDemo.Revenue]))
NextRow(Sum([MatrixDemo.Revenue]), true)
Sum([MatrixDemo.Revenue]) / NextRow()
```

SpecificColumn

Возвращает значение ячейки с указанным индексом в колонке.

Параметр	Описание
`index`	Значение индекса
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.

Возвращаемое значение зависит от того, в каком месте используется функция:

- ячейка итога: возвращает значение ячейки в группе, к которой относится итог;
- в остальных случаях возвращает значение ячейки в колонке текущей группы.

Примеры:

```
SpecificColumn("Andrew Fuller", Sum([MatrixDemo.Revenue]))
SpecificColumn(2011, Sum([MatrixDemo.Revenue]))
Sum([MatrixDemo.Revenue]) / SpecificColumn(2011)
```

SpecificRow

Возвращает значение ячейки с указанным индексом в строке.

Параметр	Описание
`index`	Значение индекса
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.

Возвращаемое значение зависит от того, в каком месте используется функция:

- ячейка итога: возвращает значение ячейки в группе, к которой относится итог;
- в остальных случаях возвращает значение ячейки в строке текущей группы.

Примеры:

```
SpecificRow("Andrew Fuller", Sum([MatrixDemo.Revenue]))
SpecificRow(2011, Sum([MatrixDemo.Revenue]))
Sum([MatrixDemo.Revenue]) / SpecificRow(2011)
```

PercentOfColumnTotal

Возвращает значение текущей ячейки, поделенное на значение итога по колонке.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.

Вызов функции

```
PercentOfColumnTotal(Sum([MatrixDemo.Revenue]))
```

эквивалентен следующему коду:

```
Sum([MatrixDemo.Revenue]) / ColumnTotal()
```

Примеры:

```
PercentOfColumnTotal(Sum([MatrixDemo.Revenue]))
[Sum([MatrixDemo.Revenue])] [PercentOfColumnTotal()]
```

PercentOfRowTotal

Возвращает значение текущей ячейки, поделенное на значение итога по строке.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.

Примеры:

```
PercentOfRowTotal(Sum([MatrixDemo.Revenue]))
[Sum([MatrixDemo.Revenue])]  [PercentOfRowTotal()]
```

PercentOfGrandTotal

Возвращает значение текущей ячейки, поделенное на значение общего итога.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.

Примеры:

```
PercentOfGrandTotal(Sum([MatrixDemo.Revenue]))
[Sum([MatrixDemo.Revenue])]  [PercentOfGrandTotal()]
```

PercentOfPreviousColumn

Возвращает значение текущей ячейки, поделенное на значение предыдущей ячейки в колонке.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.
`useInteractiveSort=false`	(необязательно) Учитывать результаты интерактивной сортировки, при которой порядок элементов может меняться.
`useThisGroup=true`	(необязательно) Использовать группу на том же уровне для поиска значения.

Возвращаемое значение зависит от того, в каком месте используется функция:

- ячейка итога: используется значение предыдущей ячейки в группе, если параметр `useThisGroup` равен `true`, иначе используется значение предыдущего итога;
- ячейка группы: используется значение предыдущей ячейки в группе.

Примеры:

```
PercentOfPreviousColumn(Sum([MatrixDemo.Revenue]))
PercentOfPreviousColumn(Sum([MatrixDemo.Revenue]), true)
[Sum([MatrixDemo.Revenue])]  [PercentOfPreviousColumn()]
```

PercentOfPreviousRow

Возвращает значение текущей ячейки, поделенное на значение предыдущей ячейки в строке.

Параметр	Описание
`aggregate`	(необязательно) Агрегатная функция. Параметр можно опустить, если к агрегату уже было обращение в ячейке.
`useInteractiveSort=false`	(необязательно) Учитывать результаты интерактивной сортировки, при которой порядок элементов может меняться.
`useThisGroup=true`	(необязательно) Использовать группу на том же уровне для поиска значения.

Возвращаемое значение зависит от того, в каком месте используется функция:

- ячейка итога: используется значение предыдущей ячейки в группе, если параметр `useThisGroup` равен `true`, иначе используется значение предыдущего итога;
- ячейка группы: используется значение предыдущей ячейки в группе.

Примеры:

```
PercentOfPreviousRow(Sum([MatrixDemo.Revenue]))
PercentOfPreviousRow(Sum([MatrixDemo.Revenue]), true)
[Sum([MatrixDemo.Revenue])]  [PercentOfPreviousRow()]
```

Свойства, доступные из ячеек данных

Матрица имеет набор свойств, которые можно использовать при печати ячеек данных. Для обращения к свойству используется имя матрицы:

```
[Matrix1.RowIndex]
```

Свойство	Возвращаемое значение	Описание
ColumnIndex	int	Индекс текущей колонки.
RowIndex	int	Индекс текущей строки.
ColumnValues	object[]	Массив значений из заголовка колонки.
RowValues	object[]	Массив значений из заголовка строки.

Эти свойства могут быть полезны для выделения ячеек цветом в зависимости от какого-либо условия.

Вы также можете обращаться к полям источника данных из ячейки. Как правило, это требуется для работы условного выделения (см. раздел Условное выделение). Так, для ячейки данных можно указать следующее условие для выделения значений, относящихся к 2012 году:

```
[MatrixDemo.Year] == 2012
```

Значение поля БД, которое использовалось для печати ячейки, берется из заголовка матрицы. Так как заголовка 2 (строка и колонка), необходимо указать, значения какого заголовка будут иметь приоритет. За это отвечает свойство матрицы `DataRowPriority`. По умолчанию свойство имеет значение `Rows`.

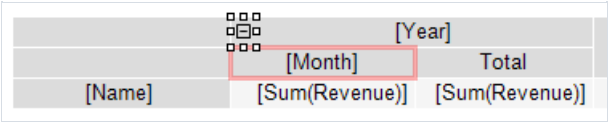
Интерактивные возможности

В этом разделе рассматриваются интерактивные возможности объекта "Улучшенная матрица":

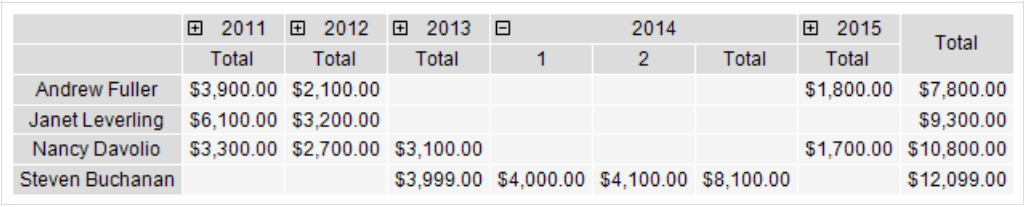
- Сворачивание/разворачивание элементов заголовка;
- Интерактивная сортировка.

Сворачивание/разворачивание элементов

Вы можете управлять видимостью отдельных элементов заголовка интерактивно, с помощью специальной кнопки типа `MatrixCollapseButton`. Кнопка вставляется в элемент заголовка и управляет видимостью других элементов. На картинке ниже показана кнопка и управляемый ею элемент (выделяется красной рамкой при выборе кнопки):



При нажатии на кнопку в окне предварительного просмотра связанные с ней элементы скрываются или показываются. При этом отчет перестраивается:



Добавление кнопки

Добавить кнопку в элемент заголовка можно с помощью контекстного меню. Выделите элемент, нажмите правую кнопку мыши и выберите пункт "Кнопка сворачивания". Кнопка будет добавлена в левую часть элемента.

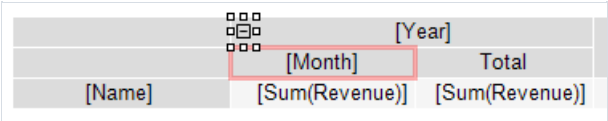
> При добавлении кнопки у элемента меняется свойство `Padding.Left`, чтобы текст не перекрывался кнопкой.

Настройка кнопки

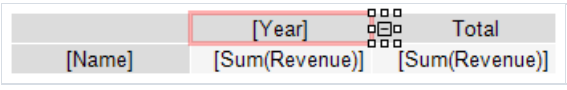
При добавлении кнопки FastReport автоматически настраивает связь кнопки и управляемого ей элемента. В некоторых случаях может понадобиться настроить связь вручную. Для этого откройте редактор элемента, который должен быть зависим от кнопки, и укажите имя кнопки на закладке "Видимость/Видимость управляется кнопкой".

> Кнопка может управлять несколькими элементами одновременно.

Кнопка может располагаться выше управляемого ею элемента:



или на одном уровне с ним:



Начальное состояние управляемого элемента - его видимость - задается в редакторе элемента на закладке "Видимость/Видимый".

Удаление кнопки

Удалить кнопку можно двумя способами:

- выделите кнопку и нажмите клавишу `Delete`;
- в контекстном меню элемента снимите галочку у пункта "Кнопка сворачивания".

Перемещение кнопки

По умолчанию кнопка имеет свойство `Dock = Left`. Это значит, что она прижата к левому краю элемента. Чтобы переместить кнопку на новое место, в окне "Свойства" установите свойство `Dock = None`.

> Вы также можете использовать свойство `Anchor` кнопки для привязки ее к определенному месту элемента.

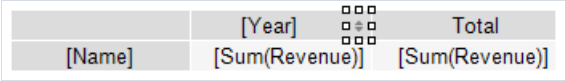
Настройка внешнего вида кнопки

С помощью панели инструментов "Рамка" вы можете настроить значок кнопки: цвет и стиль рамки, цвет фона. Кроме того, в окне "Свойства" можно установить следующие свойства кнопки:

Свойство	Значение по умолчанию	Описание
Cursor	Hand	Вид курсора мыши.
Exclusive	False	Если `true`, то только один элемент может находиться в развернутом состоянии.
Exportable	False	Если `true`, кнопка будет отображаться при экспорте отчета.
Printable	False	Если `true`, кнопка будет отображаться при печати отчета.
ShowCollapseExpandMenu	False	Определяет, надо ли показывать меню с пунктами "Свернуть/развернуть все" при нажатии правой кнопки мыши на данной кнопке.
Symbol	PlusMinus	Символ, отображаемый внутри кнопки.
SymbolSize	5	Размер символа кнопки.

Интерактивная сортировка

С помощью кнопки сортировки `MatrixSortButton` можно интерактивно сортировать строки или колонки матрицы. Кнопка вставляется в элемент заголовка нижнего уровня:



При нажатии на кнопку в окне предварительного просмотра происходит сортировка противоположного заголовка. В примере ниже сортируются строки по значению в выбранной колонке:

	2011	2012	2013	2014	2015	Total
Andrew Fuller	$3,900.00	$2,100.00			$1,800.00	$7,800.00
Janet Leverling	$6,100.00	$3,200.00				$9,300.00
Nancy Davolio	$3,300.00	$2,700.00	$3,100.00		$1,700.00	$10,800.00
Steven Buchanan			$3,999.00	$8,100.00		$12,099.00

Каждое нажатие кнопки переключает режим сортировки: по возрастанию/по убыванию/без сортировки.

Добавление кнопки

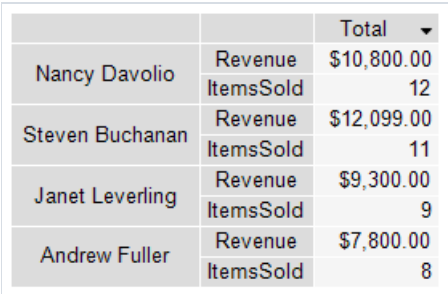
Добавить кнопку в элемент заголовка можно с помощью контекстного меню. Выделите элемент, нажмите правую кнопку мыши и выберите пункт "Кнопка сортировки". Кнопка будет добавлена в правую часть элемента.

> При добавлении кнопки у элемента меняется свойство `Padding.Right`, чтобы текст не перекрывался кнопкой.

Настройка кнопки

Режим сортировки заголовка задается в его редакторе на вкладке "Сортировка/Интерактивная сортировка по итогу". Возможны следующие значения:

- "Нет" - сортировка данного заголовка не выполняется;
- "Авто" - режим по умолчанию. Сортировка идет по значению первого итога (агрегата);
- Имя итога (агрегата): если у заголовка несколько выводимых значений, вы можете выбрать одно из них для сортировки. В примере ниже, чтобы отсортировать заголовок строки по значению `ItemsSold`, надо выбрать имя агрегата `Sum([MatrixDemo.ItemsSold])`:



		Total
Nancy Davolio	Revenue	$10,800.00
	ItemsSold	12
Steven Buchanan	Revenue	$12,099.00
	ItemsSold	11
Janet Leverling	Revenue	$9,300.00
	ItemsSold	9
Andrew Fuller	Revenue	$7,800.00
	ItemsSold	8

Удаление кнопки

Удалить кнопку можно двумя способами:

- выделите кнопку и нажмите клавишу `Delete`;

- в контекстном меню элемента снимите галочку у пункта "Кнопка сортировки".

Перемещение кнопки

По умолчанию кнопка имеет свойство `Dock = Right`. Это значит, что она прижата к правому краю элемента. Чтобы переместить кнопку на новое место, в окне "Свойства" установите свойство `Dock = None`.

> Вы также можете использовать свойство `Anchor` кнопки для привязки ее к определенному месту элемента.

Настройка внешнего вида кнопки

С помощью панели инструментов "Рамка" вы можете настроить значок кнопки: цвет и стиль рамки, цвет фона. Кроме того, в окне "Свойства" можно установить следующие свойства кнопки:

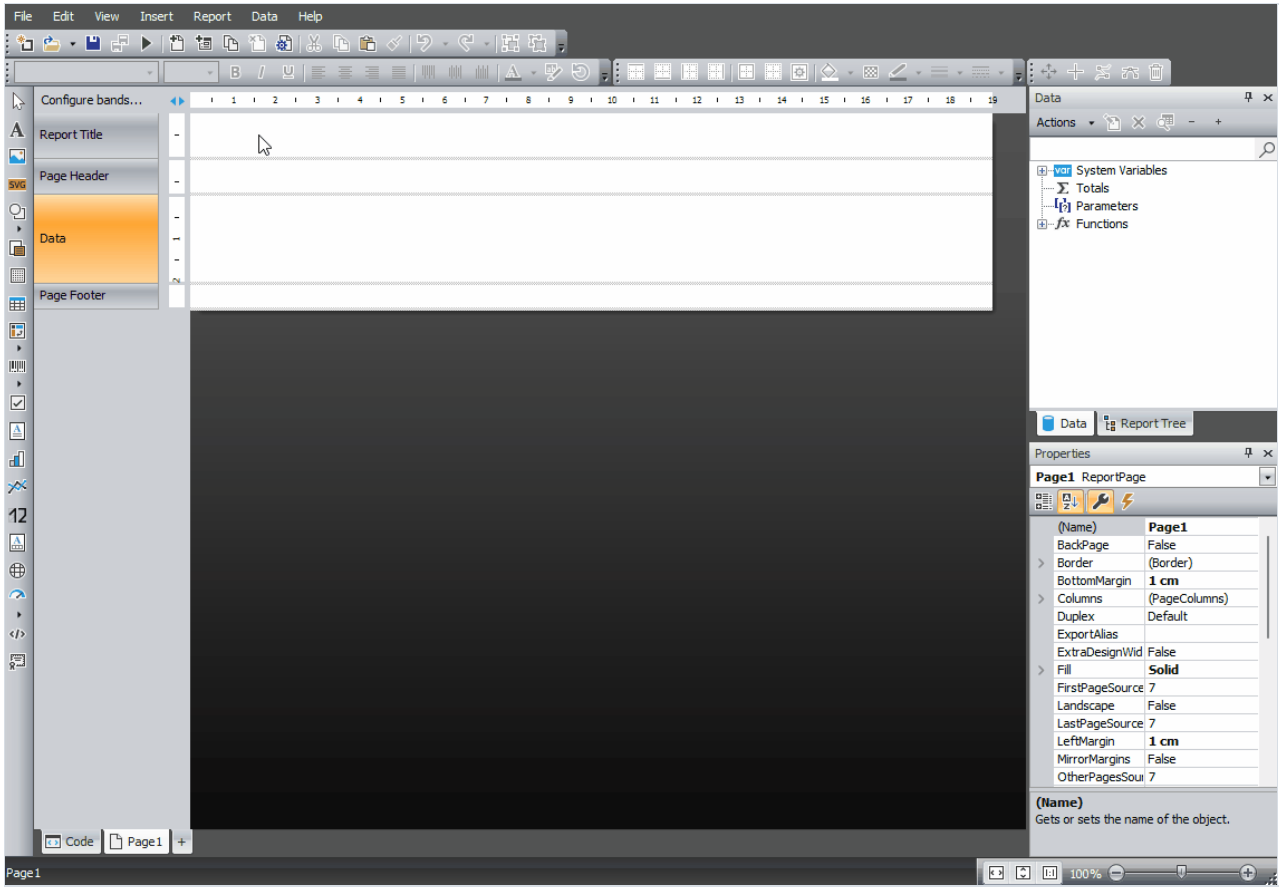
Свойство	Значение по умолчанию	Описание
AllowInactiveSort	True	Определяет, может ли кнопка иметь неактивное состояние (режим "без сортировки").
Cursor	Hand	Вид курсора мыши.
Exportable	False	Если `true`, кнопка будет отображаться при экспорте отчета.
InactiveSortColor	Gray	Цвет кнопки в неактивном состоянии.
Printable	False	Если `true`, кнопка будет отображаться при печати отчета.
Symbol	Arrow	Символ кнопки.
SymbolSize	7	Размер символа кнопки.

Примеры использования

Ниже приведены примеры использования объекта "Улучшенная матрица" в виде анимированных gif.

Пример 1. Простая матрица

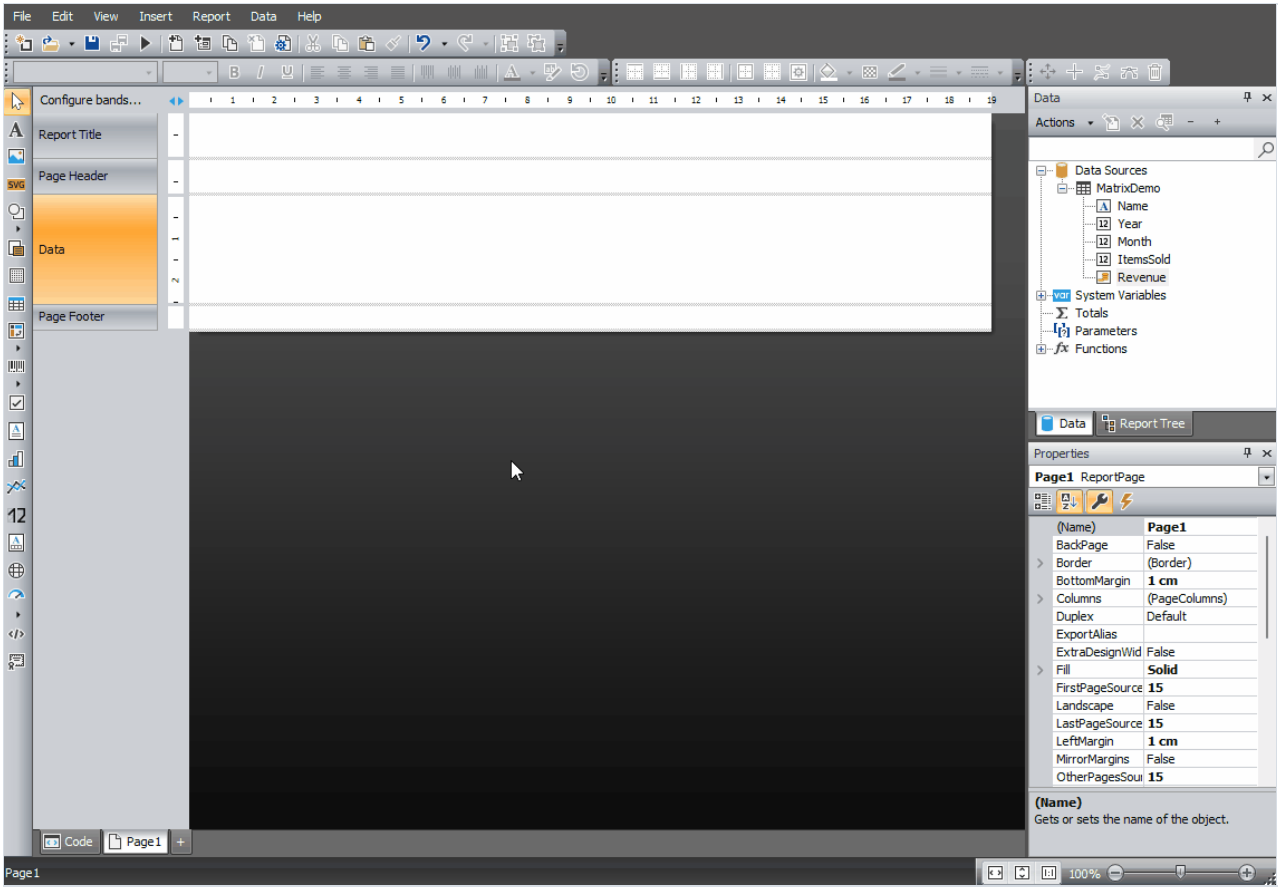
Создание простой матрицы, выбор стиля, добавление итогов:



Ссылка на gif файл

Пример 2. Матрица с составными заголовками

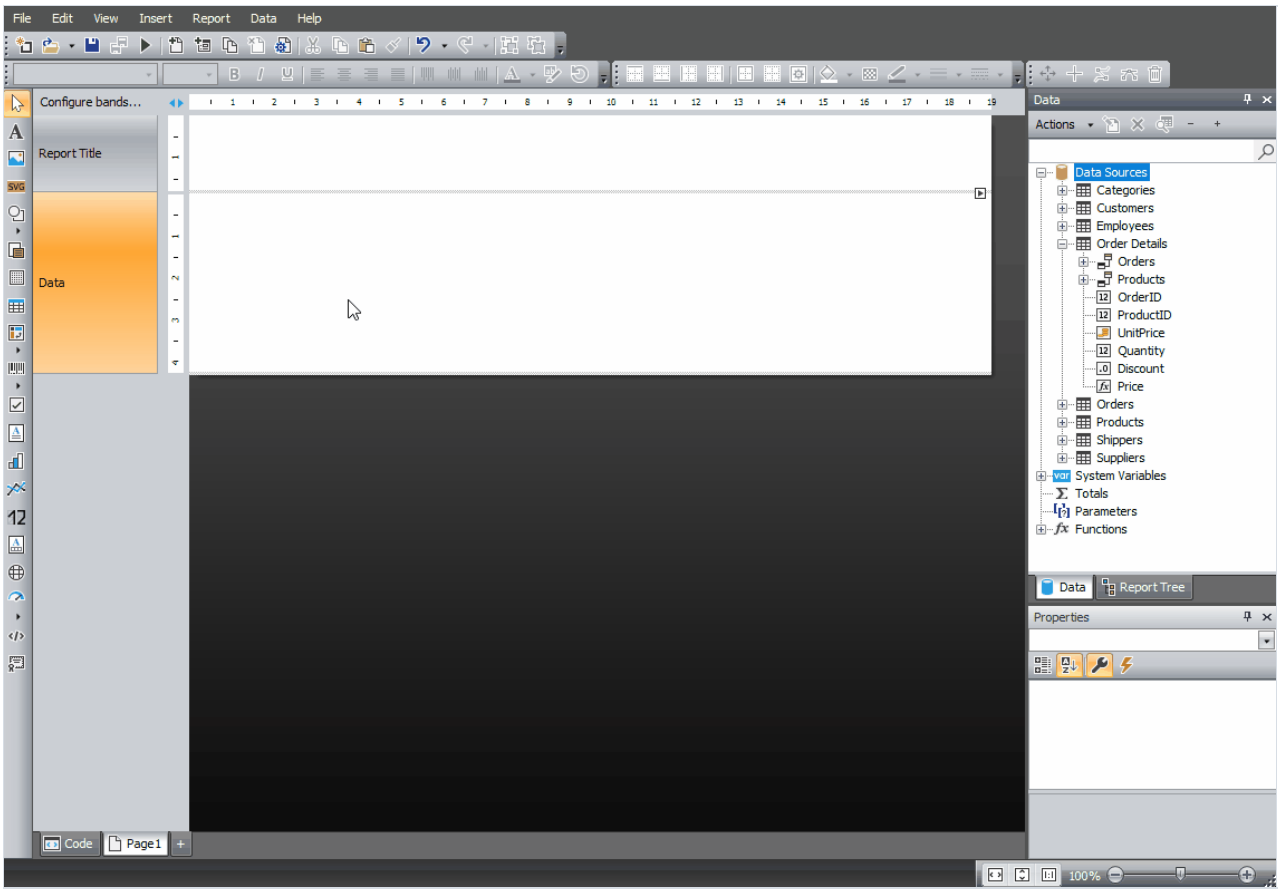
Создание матрицы с составными заголовками, добавление итогов, использование свойства `DisplayText`, ступенчатое отображение заголовка, кнопки сворачивания/разворачивания:



Ссылка на gif файл

Пример 3. Группировка TopN

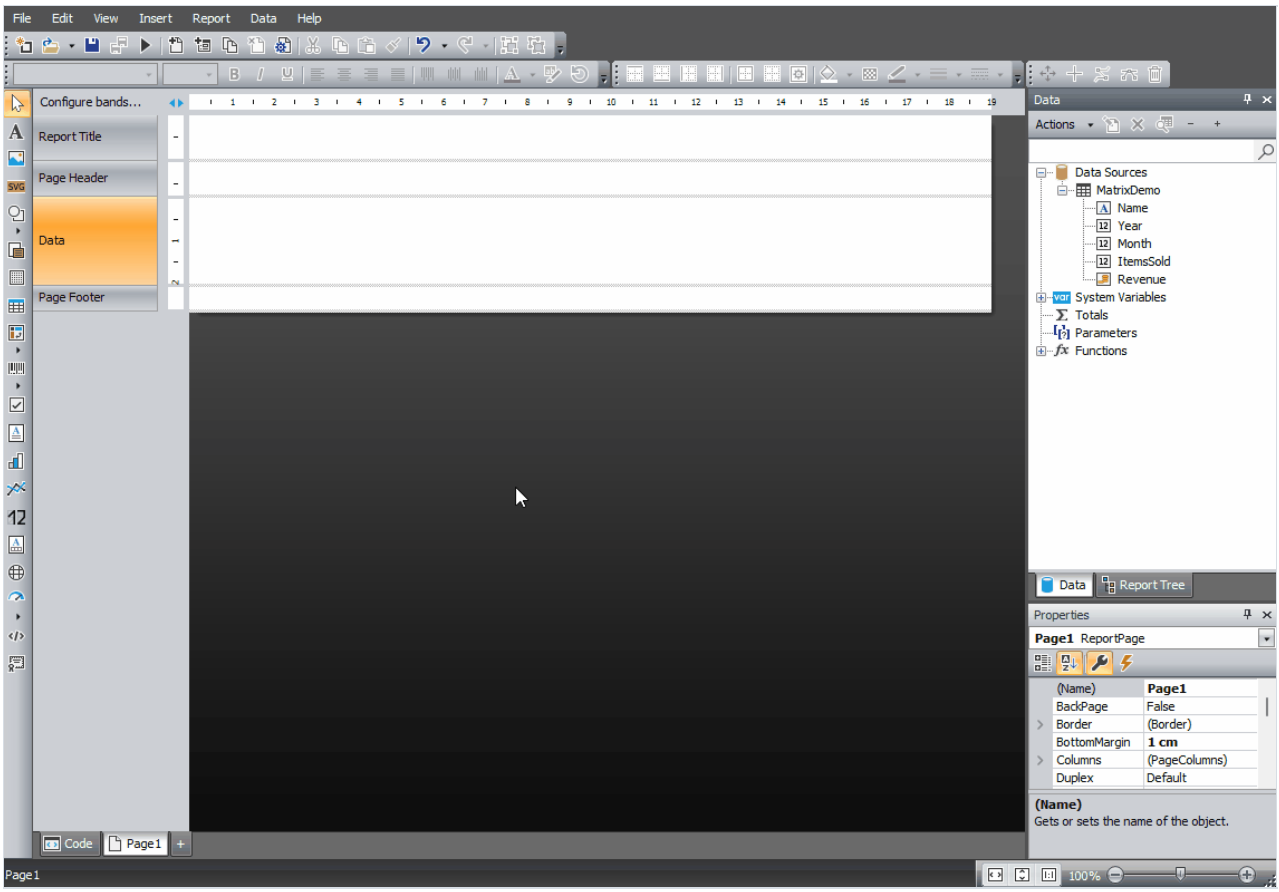
Создание и настройка группы TopN (два варианта - простой и настраиваемый), разворачивание элементов, использование свойства матрицы `ItemCount` :



Ссылка на gif файл

Пример 4. Интерактивная сортировка

Добавление кнопок интерактивной сортировки, добавление второго поля БД в область данных (`ItemsSold`), настройка интерактивной сортировки по этому полю.



Ссылка на gif файл

Интерактивные отчеты

Готовый отчет FastReport можно сделать интерактивным. Это значит, что он будет реагировать на действия пользователя (а именно, на нажатие кнопки мыши). Вы можете использовать следующие виды взаимодействия:

- при нажатии на элемент отчета (бэнд или объект) выполняется какое-либо действие. Например, строится детальный отчет и показывается в отдельном окне;
- сбоку окна просмотра отображается структура отчета, которую можно использовать для навигации по отчету.

Гиперссылки

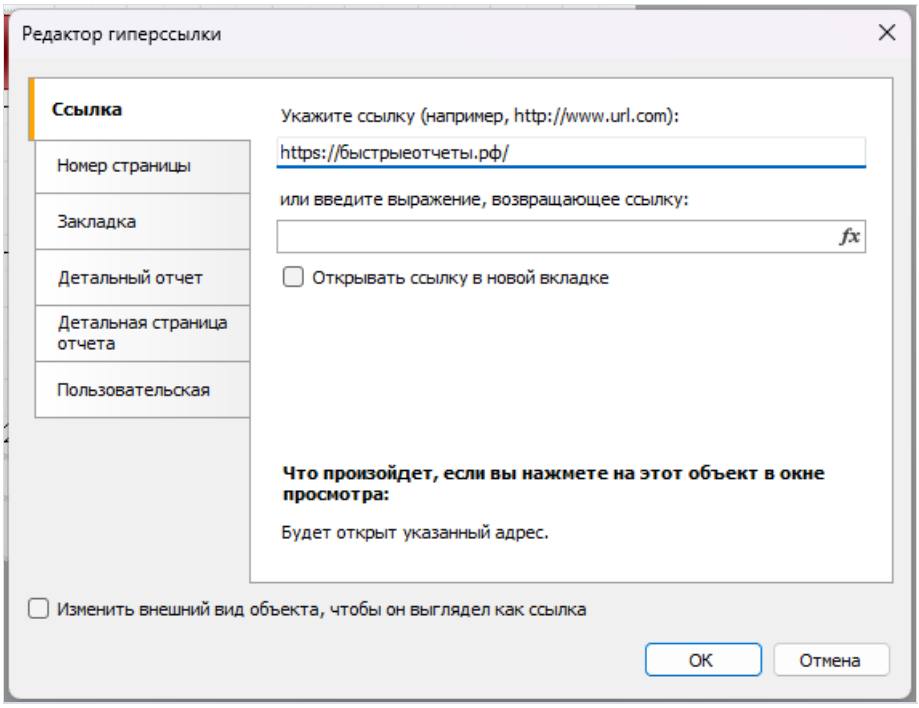
В этом разделе речь пойдет о свойстве "Гиперссылка" (`Hyperlink`), которое имеется практически у всех объектов отчета, включая бэнды. Используя это свойство, можно определить реакцию объекта на нажатие мыши в окне предварительного просмотра.

При нажатии на такой объект может быть выполнено одно из следующих действий:

- переход на адрес URL;
- отсылка e-mail;
- выполнение какой-нибудь системной команды;
- переход на страницу отчета с указанным номером;
- переход на закладку, определенную в другом объекте отчета;
- запуск детального отчета в отдельном окне просмотра;
- реакция, определенная пользователем в скрипте.

Настройка гиперссылки

Для настройки гиперссылки выделите объект, который будет реагировать на нажатие мыши, и щелкните на нем правой кнопкой мыши. В контекстном меню выберите пункт "Гиперссылка...". Откроется окно редактора гиперссылки:



Выбрать тип гиперссылки можно с помощью закладок в левой части окна. После того как вы настроили ссылку, можно отметить флажок "Изменить внешний вид объекта..." внизу окна. Внешний вид объекта изменится при этом следующим образом:

- будет установлен синий цвет текста и подчеркивание;
- будет установлена форма курсора (свойство `Cursor`) в виде руки.

В некоторых случаях гиперссылку нужно показывать в окне просмотра, но не нужно печатать на принтере. Это легко сделать, выключив свойство объекта `Printable`. Это можно сделать в окне "Свойства".

Ссылка на адрес URL

Используя этот тип ссылки, вы можете:

- перейти на заданный адрес Интернет;
- выполнить какую-либо системную команду, например, `mailto:` для отправки электронной почты.

> При нажатии на ссылку этого типа выполняется команда `System.Diagnostics.Process.Start`, которой в качестве параметра передается значение ссылки.

Вы можете указать значение ссылки двумя способами:

- указать значение напрямую, например, https://быстрыеотчеты.рф;
- указать выражение, которое возвращает значение ссылки. Это выражение будет вычислено в момент построения отчета, когда обрабатывается данный объект.

Ссылка на номер страницы

Используя этот тип ссылки, вы можете организовать навигацию по страницам готового отчета. Чаще всего используется переход на первую страницу отчета. Для этого в качестве значения ссылки укажите номер страницы (в данном случае 1).

Вы можете указать номер страницы двумя способами:

- указать номер напрямую, например, 1;
- указать выражение, которое возвращает номер страницы. Это выражение будет вычислено в момент построения отчета, когда обрабатывается данный объект.

Ссылка на закладку

Используя этот тип ссылки, вы можете переходить на закладку, определенную в другом объекте или программно.

Для тех, кто знаком с языком разметки HTML, достаточно упомянуть, что принцип работы закладки такой же, как у якоря (`anchor`). Закладка имеет имя и определенную позицию в готовом отчете (номер страницы и позиция на странице). При переходе на закладку по ее имени вы перемещаетесь на указанную позицию.

Чтобы использовать этот тип ссылки, сначала нужно определить закладку. Чтобы это сделать, выделите объект отчета, на который вы хотите перейти при щелчке на ссылку. Закладка есть практически у всех объектов отчета; ее значение указывается в свойстве `Bookmark`. Изменить это свойство можно с помощью окна "Свойства".

Свойство `Bookmark` имеет тип "Выражение", которое вы можете использовать следующим образом:

- указать имя закладки в виде строки, т.е. в апострофах:

```
"МояЗакладка"
```

- указать выражение, которое вернет имя закладки. Например, в качестве выражения можно использовать поле данных. Значение выражения будет вычислено при запуске отчета, в момент обработки данного объекта.

После того как закладка определена, вы можете указать ее имя в окне настройки гиперссылки. Это можно сделать двумя способами:

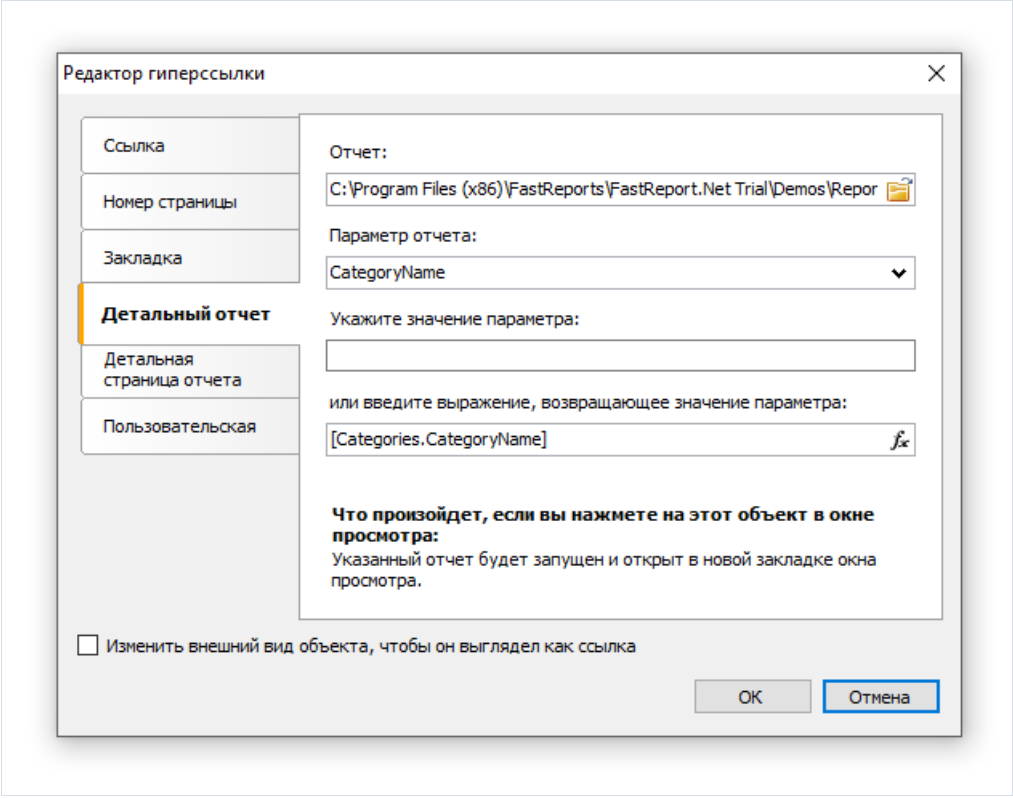
- указать имя закладки напрямую;
- указать выражение, которое вернет имя закладки. Например, это может быть поле данных. Это выражение будет вычислено при запуске отчета, в момент обработки данного объекта.

Ссылка на детальный отчет

Используя этот тип ссылки, вы можете выполнять другой отчет и показывать его в отдельном окне просмотра.

В настройках данного типа ссылки вы должны указать следующую информацию:

- имя детального отчета;
- имя параметра в отчете, которому будет передано значение из гиперссылки;
- значение гиперссылки, которое надо передать в параметр отчета.



При нажатии на ссылку произойдет следующее:

- будет загружен указанный отчет;
- в параметр отчета будет передано значение гиперссылки;
- отчет будет построен и запущен в отдельном окне просмотра.

Значение параметра отчета можно указать двумя способами:

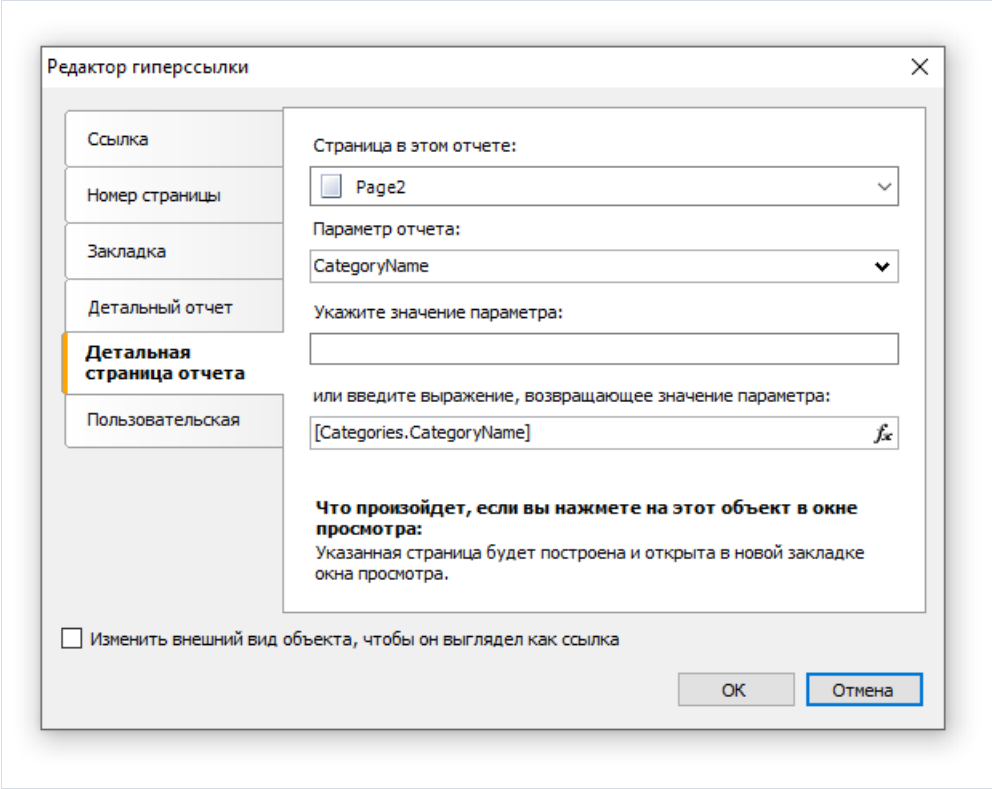
- указать значение напрямую;
- указать выражение, которое вернет значение. Это выражение будет вычислено при запуске отчета, в момент обработки данного объекта.

Ссылка на детальную страницу

Этот тип ссылки работает аналогичным образом, за исключением того, что в качестве детального отчета используется другая страница в текущем отчете. Для этого ваш отчет должен содержать как минимум две страницы: одну с основным отчетом, другую - с детальным.

В настройках данного типа ссылки вы должны указать следующую информацию:

- имя страницы в этом отчете;
- имя параметра в отчете, которому будет передано значение из гиперссылки;
- значение гиперссылки, которое надо передать в параметр отчета.



При нажатии на ссылку произойдет следующее:

- в параметр отчета будет передано значение гиперссылки;
- указанная страница отчета будет построена и показана в отдельном окне просмотра.

Значение параметра отчета можно указать двумя способами:

- указать значение напрямую;
- указать выражение, которое вернет значение. Это выражение будет вычислено при запуске отчета, в момент обработки данного объекта.

> Когда вы выбираете страницу отчета, ее свойство `Visible` сбрасывается в `false`. Это означает, что при построении основного отчета эта страница будет пропущена.

Прочая ссылка

Используя этот тип ссылки, вы можете определить собственную реакцию на нажатие мыши. Для этого используется обработчик события `Click` объекта, который может быть написан в скрипте отчета. Чтобы сделать это:

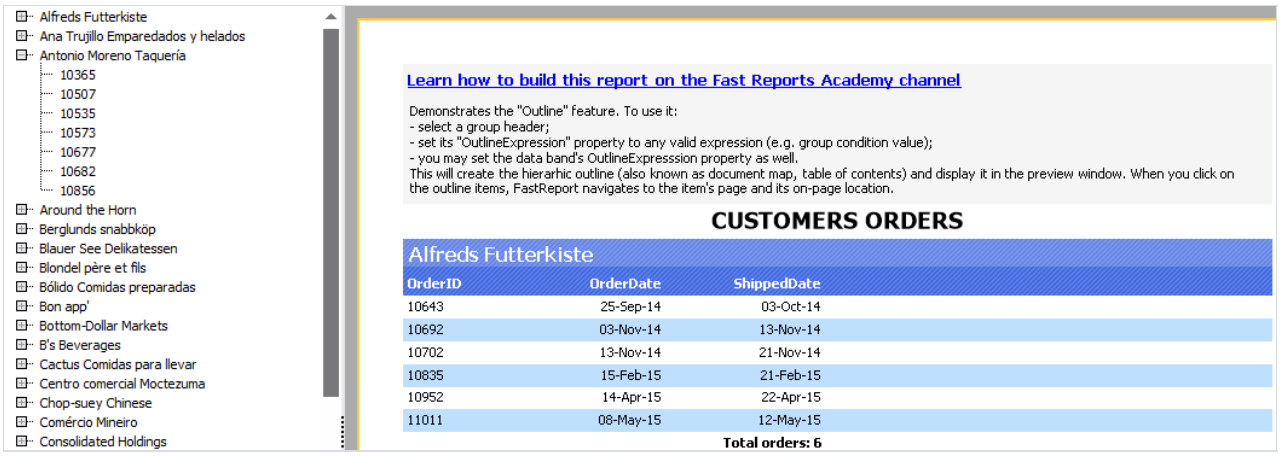
- выделите объект и откройте окно "Свойства";
- нажмите кнопку  , чтобы показать события объекта;
- сделайте двойной щелчок мышью на событии `Click`. FastReport переключится на окно "Код" и создаст пустой обработчик события.

В коде обработчика вы можете делать все, что считаете нужным. Вам, скорее всего, понадобится ссылка на объект, который вызвал обработчик, и параметры гиперссылки. Используйте параметр обработчика `sender`:

```
private void Text2_Click(object sender, EventArgs e)
{
  // sender - это объект, который был кликнут.
  // Чтобы получить значение гиперссылки, надо привести sender к типу ReportComponentBase.
  object hyperlinkValue = (sender as ReportComponentBase).Hyperlink.Value;

  // здесь выполняем какие-либо действия, например, показываем значение в диалоге:
  MessageBox.Show("Hyperlink value = " + hyperlinkValue.ToString());
}
```

Структура отчета

Структура отчета (также известная как дерево отчета, оглавление) - это элемент управления типа TreeView, который отображается в окне просмотра:



Элемент управления показывает древовидную структуру, которая была сформирована при работе отчета. При щелчке на элементе дерева происходит переход на соответствующий элемент отчета.

Если отчет имеет структуру, она будет показана автоматически. Показать или спрятать структуру можно, нажав кнопку  на панели управления. Автоматически структура отчета не создается; для этого вы должны настроить некоторые элементы отчета.

Для создания структуры используется свойство `OutlineExpression`, которое имеется у страницы отчета и бэндов. В этом свойстве надо указать выражение, которое возвращает текст элемента структуры. При печати бэнда, у которого установлено это свойство, происходит следующее:

- вычисляется значение выражения, которое указано в свойстве `OutlineExpression`;
- в структуру отчета добавляется элемент с полученным текстом;
- при печати подчиненных бэндов (например, в отчете master-detail) структура отчета формируется таким образом, что повторяет структуру бэндов.

Свойство `OutlineExpression` можно настроить в окне "Свойства", предварительно выделив нужный бэнд.

Ниже приведены способы настройки структуры для печати разных типов отчетов:

- если вы хотите отобразить в структуре список страниц готового отчета, настройте свойство `OutlineExpression` у страницы отчета. Как правило, выражение возвращает номер страницы:

```
[PageN]
```

- в отчете типа "Список" с одним бэндом "Данные" настройте свойство `OutlineExpression` у этого бэнда. В качестве выражения можно использовать любое поле данных, которое печатается в бэнде;
- в отчете типа master-detail с двумя бэндами "Данные" настройте свойство `OutlineExpression` у обоих бэндов. В качестве выражений используйте поля данных, которые печатаются в соответствующих бэндах. Например, в отчете типа "Категории/Продукты" выражение `OutlineExpression` для первого бэнда будет содержать название категории, для второго - название продукта;
- в отчете с группой настройте свойство `OutlineExpression` у заголовка группы и бэнда "Данные". В качестве выражения для заголовка группы обычно используется условие группировки. Для бэнда

"Данные" укажите одной из поле данных, которое печатается в этом бэнде.

Примеры использования

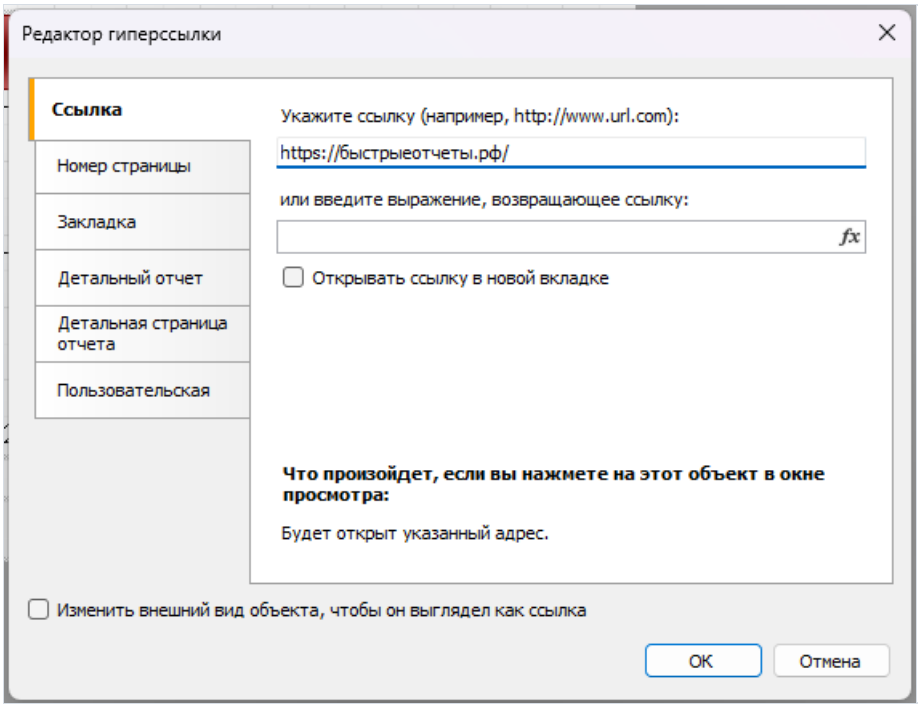
Пример 1. Ссылка на веб-страницу

В этом примере мы создадим простой отчет с одним объектом "Текст". При нажатии на объект в окне просмотра мы перейдем на веб-страницу FastReport.

Создайте новый отчет и добавьте в него объект "Текст". Напишите в нем следующий текст:

```
Перейти на домашнюю страницу FastReport
```

Щелкните правой кнопкой мыши на объекте и выберите в контекстном меню пункт "Гиперссылка...". Настройте ссылку следующим образом:



После этого включите флажок "Изменить внешний вид объекта...", чтобы применить к объекту атрибуты ссылки (синий цвет текста, подчеркивание и форма курсора в виде руки).

Запустите отчет и щелкните на объекте. Откроется окно веб-браузера, и вы попадете на страницу FastReport.

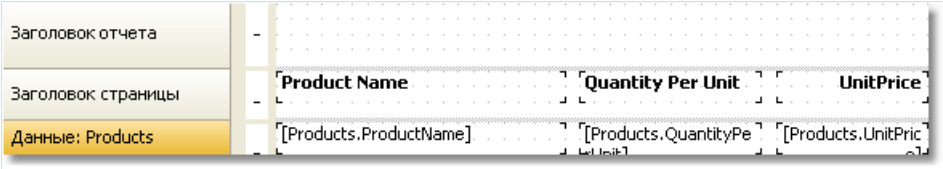
Пример 2. Построение детального отчета

В этом примере мы построим отчет - список категорий продуктов. При щелчке на названии категории будет показан детальный отчет - список продуктов в данной категории.

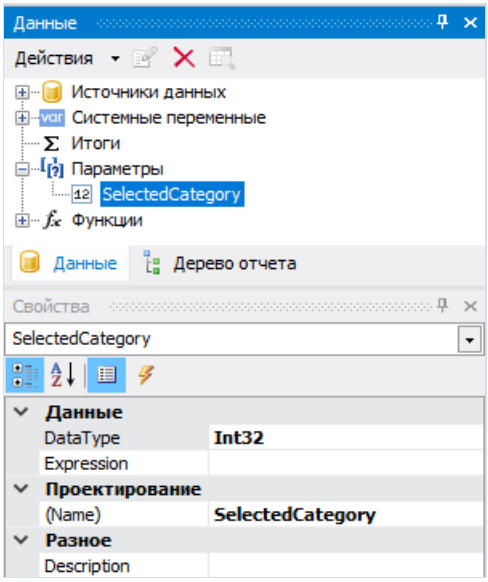
Вам необходимо сделать следующее:

- создать детальный отчет;
- определить в нем параметр - идентификатор категории;
- задать условие фильтрации строк данных в детальном отчете по этому параметру;
- создать основной отчет;
- в основном отчете настроить гиперссылку так, чтобы запускался детальный отчет с параметром, равным выбранной категории.

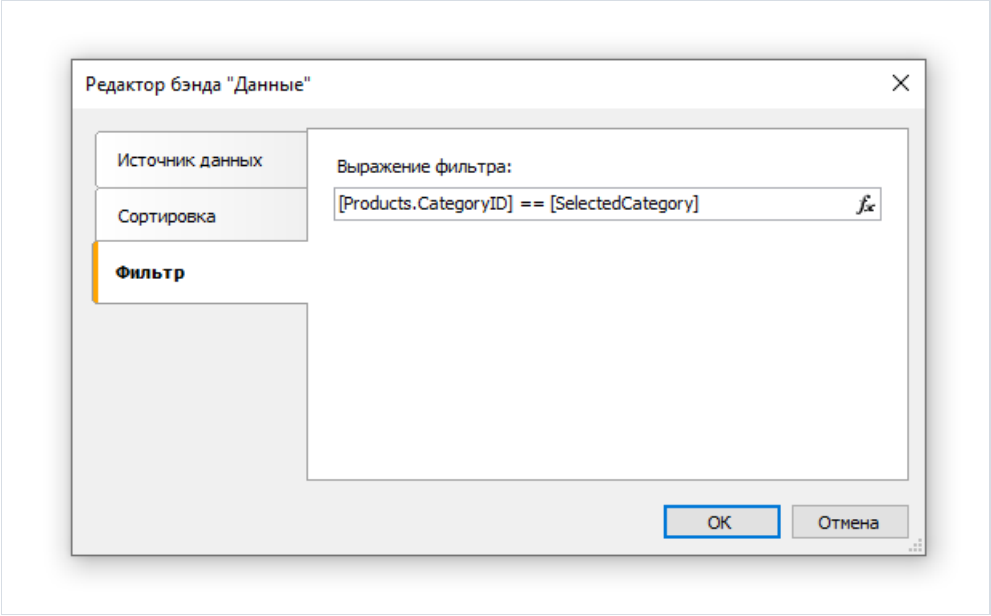
Сначала займемся созданием детального отчета, который печатает список продуктов. Для этого создадим новый отчет и выберем для него источник данных - таблицу Products. Расположим объекты следующим образом:



Для идентификации категории проще всего использовать значение поля `CategoryID`, которое имеется в обеих таблицах - Categories и Products. Создадим параметр отчета, с помощью которого будет передаваться выбранная категория. Настроим его следующим образом:



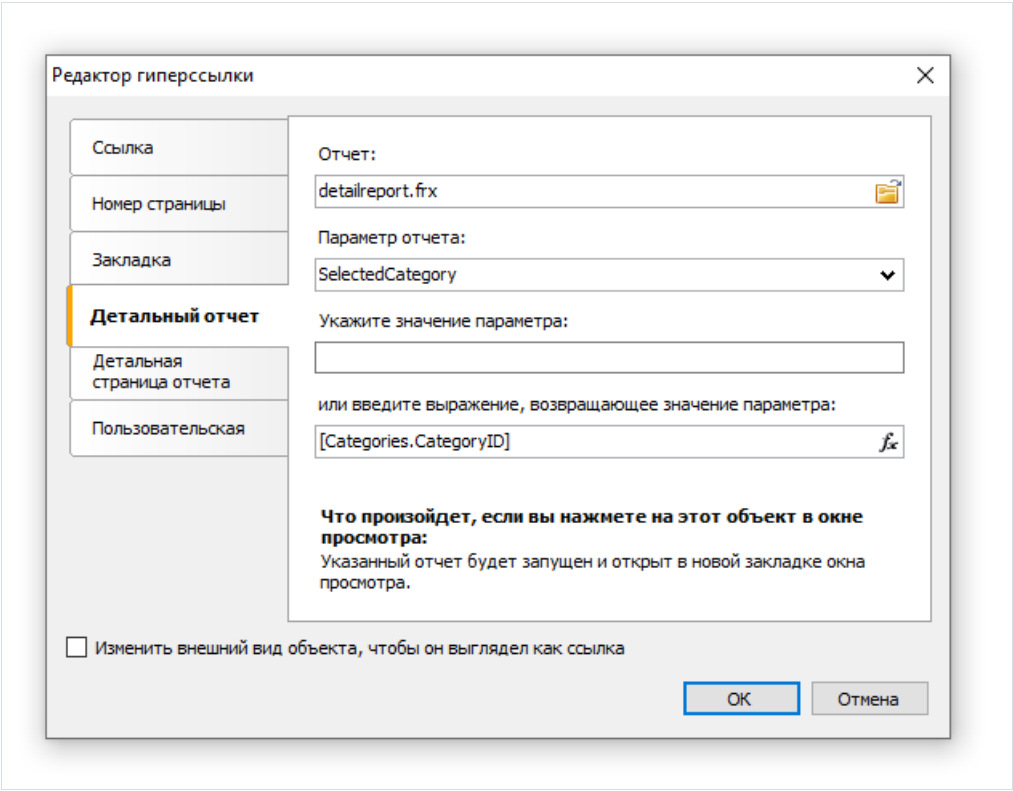
Теперь нужно указать условие фильтрации, с помощью которого будут отобраны все продукты, относящиеся к выбранной категории. Для этого сделайте двойной щелчок на бэнде "Данные" и на закладке "Фильтр" укажите следующее условие (его можно построить визуально, нажав кнопку "fx" в правой части строки ввода):



На этом создание детального отчета завершено.

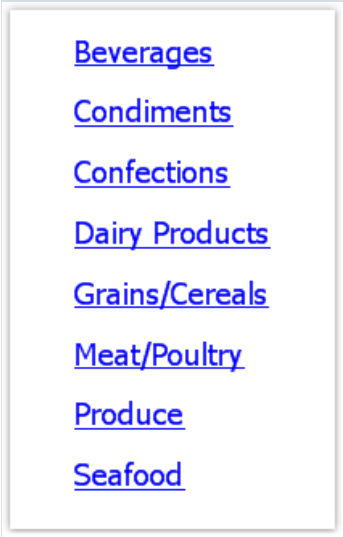
Приступим к созданию основного отчета. Для этого создайте новый отчет и выберите для него источник данных - Categories. Расположите объекты следующим образом:

Щелкните правой кнопкой мыши на объекте "Текст" и выберите пункт меню "Гиперссылка...". Настройте ссылку следующим образом:

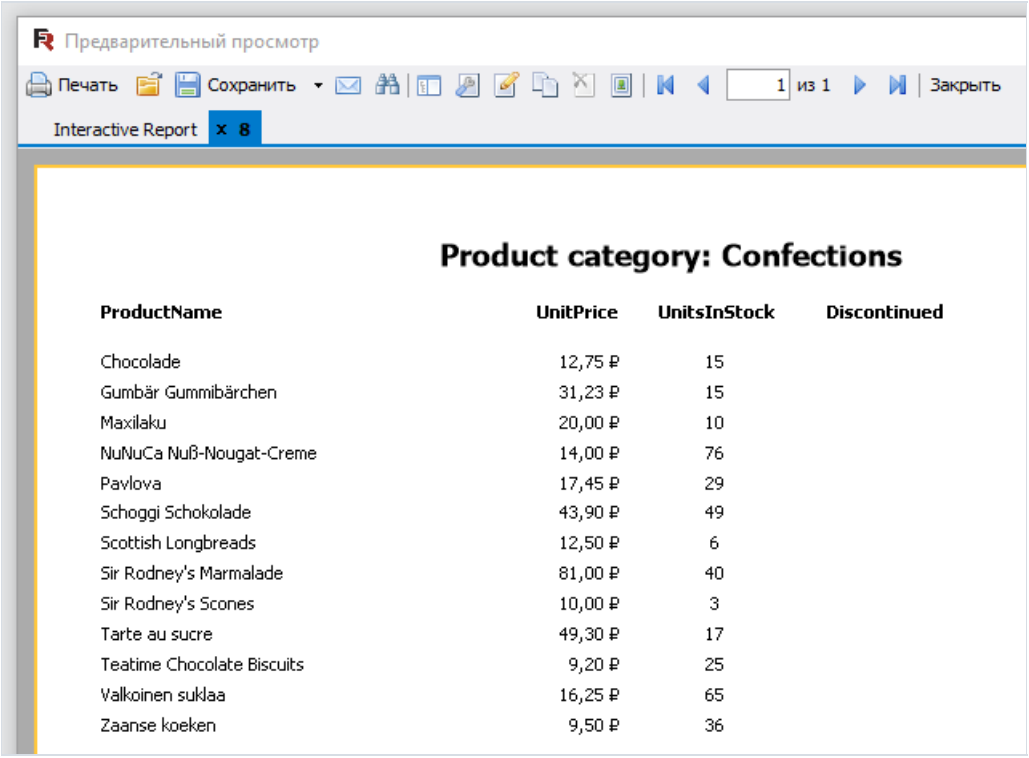


Как видно на рисунке, в качестве отчета надо выбрать имя файла с детальным отчетом. Параметр отчета можно выбрать из выпадающего списка, нажав кнопку в правой части списка. В качестве значения параметра укажем поле `CategoryID` таблицы Categories.

На этом построение отчета завершено. Запустите отчет, и вы увидите список категорий:



Если щелкнуть на одной из категорий, будет построен детальный отчет. Он будет показан на отдельной закладке окна просмотра:



Как видно на рисунке, в качестве текста закладки используется значение, передаваемое в параметр детального отчета. В нашем случае это поле `CategoryID` числового типа. Это выглядит неинформативно и некрасиво. Можно переделать наш отчет так, чтобы передавалось название категории. Для этого сделайте следующее:

В детальном отчете:

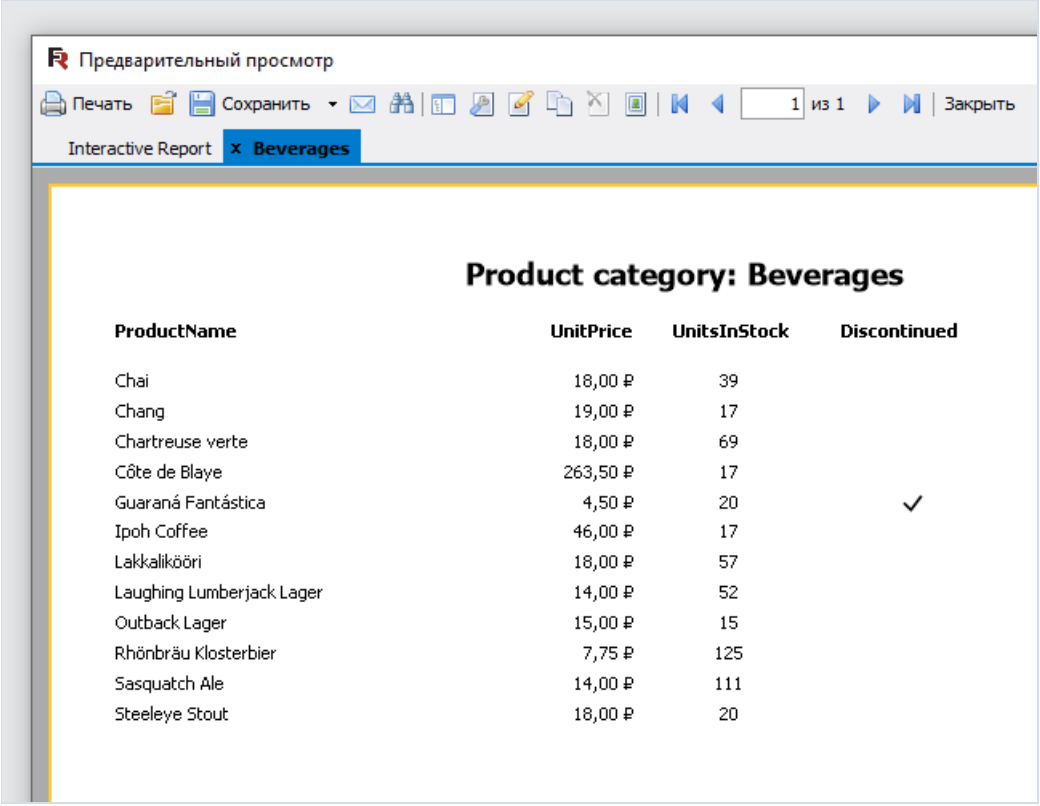
- поменяйте тип параметра (свойство `DataType`) на `String`;
- добавьте в отчет источник данных - Categories. Он будет использован для обращения к полю `CategoryName` при фильтрации данных;
- поменяйте выражение фильтрации у бэнда "Данные":

```
[Products.Categories.CategoryName] == [SelectedCategory]
```

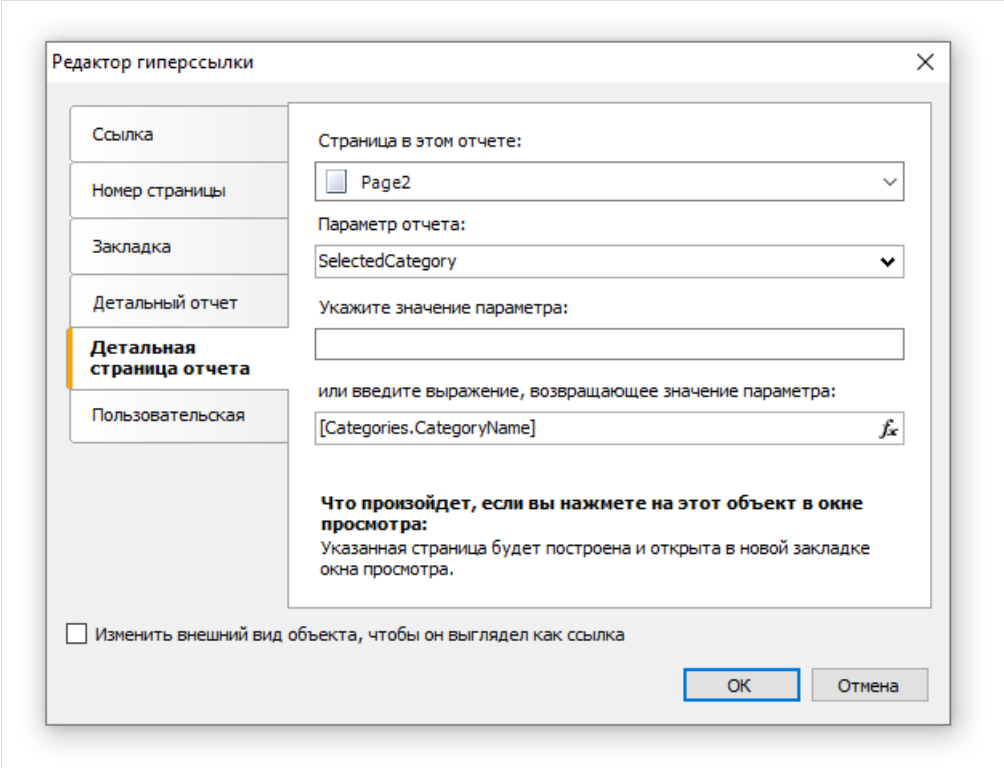
В основном отчете:

- настройте гиперссылку так, чтобы передавать значение `[Categories.CategoryName]` в параметр `SelectedCategory`.

Если запустить отчет теперь, мы увидим, что закладка детального отчета имеет текст с названием категории, а не ее идентификатором. Теперь в детальный отчет можно добавить объект "Текст", который будет печатать название выбранной категории в заголовке отчета:



В процессе работы над этим примером нам пришлось создать два отчета и переключаться между ними несколько раз. Это не очень удобно. Для того чтобы облегчить задачу, можно разместить два отчета в одном: основной отчет будет содержаться на первой странице, детальный - на второй. В этом случае гиперссылку надо настроить следующим образом:



В данном случае надо выбрать страницу отчета, на которой содержатся объекты детального отчета, а именно - Page2.

Пример 3. Интерактивный объект "Матрица"

В этом примере мы рассмотрим, как построить детальный отчет, щелкнув на значение в ячейке объекта "Матрица". В качестве примера мы будем использовать матрицу, которая показывает объемы продаж сотрудников по годам.

В качестве источника данных для матрицы используется таблица MatrixDemo. В ней представлены продажи сотрудников, сгруппированные по году и месяцу:

Name	Year	Month	ItemsSold	Revenue
Nancy Davolio	1999	2	1	1000
Nancy Davolio	1999	11	1	1100
Nancy Davolio	1999	12	1	1200
Nancy Davolio	2000	1	1	1300
Nancy Davolio	2000	2	2	1400
Nancy Davolio	2001	2	2	1500
Nancy Davolio	2001	3	2	1600
Nancy Davolio	2002	1	2	1700
Andrew Fuller	2002	1	2	1800
Andrew Fuller	1999	10	2	1900
Andrew Fuller	1999	11	2	2000
Andrew Fuller	2000	2	2	2100
Janet Leverling	1999	10	3	3000
Janet Leverling	1999	11	3	3100
Janet Leverling	2000	3	3	3200
Steven Buchanan	2001	1	3	4000
Steven Buchanan	2001	2	4	4100
Steven Buchanan	2000	1	4	3999

На основе этих данных построим матрицу, которая печатает следующую информацию:

- в заголовке колонки печатается поле `MatrixDemo.Year`;

- в заголовке строки печатается поле `MatrixDemo.Name`;
- в ячейке печатается поле `MatrixDemo.Revenue`.

Готовая матрица выглядит следующим образом:

	Год				
Сотрудник	**1999**	**2000**	**2001**	**2002**	**Итого**
Andrew Fuller	3 900,00р.	2 100,00р.		1 800,00р.	7 800,00р.
Janet Leverling	6 100,00р.	3 200,00р.			9 300,00р.
Nancy Davolio	3 300,00р.	2 700,00р.	3 100,00р.	1 700,00р.	10 800,00р.
Steven Buchanan		3 999,00р.	8 100,00р.		12 099,00р.
Итого	13 300,00р.	11 999,00р.	11 200,00р.	3 500,00р.	39 999,00р.

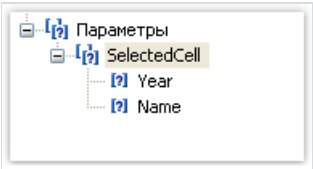
Как видно, значения в ячейках представляют собой сумму продаж сотрудника за весь год. Давайте сделаем так, чтобы при щелчке на ячейке матрицы показывался более подробный отчет. Этот отчет должен давать ответ на вопрос - "откуда в ячейке матрицы взялось это значение?". В нашем случае подробный отчет может содержать продажи выбранного сотрудника за каждый месяц выбранного года.

Как связать ячейку с данными, на основе которых она была напечатана? Каждая ячейка матрицы имеет адрес. Это совокупность значений из заголовка колонки и строки. В нашем примере адрес ячейки - это совокупность года и имени сотрудника. Именно эти данные можно передать в детальный отчет. Как это сделать? Очень просто: настройте гиперссылку, указав только имя отчета и название параметра. Значение параметра указывать не надо: для ячеек матрицы FastReport сам формирует значение и передает его в параметр.

Предположим, мы щелкнули на верхней левой ячейке, содержащей число 3900. Это сумма продаж сотрудника "Andrew Fuller" за 1999 год. В каком виде значение передается в параметр? FastReport комбинирует значение колонки и значение строки, используя точку с запятой в качестве разделителя:

```
1999;Andrew Fuller
```

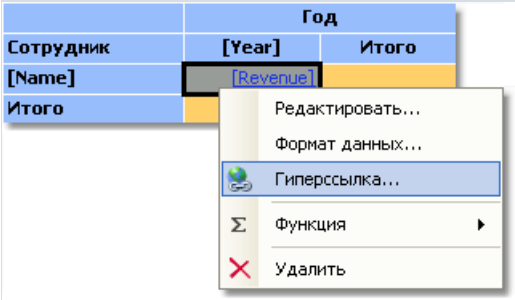
То есть, в детальном отчете мы должны извлечь из этого значения год и имя сотрудника, преобразовать год в число, и эти значения использовать для фильтрации данных? Ничего подобного! Все, что нужно сделать в детальном отчете - создать параметр, имеющий вложенные параметры. Как это сделать, смотрите в главе "Данные". В данном случае родительский параметр может выглядеть следующим образом:



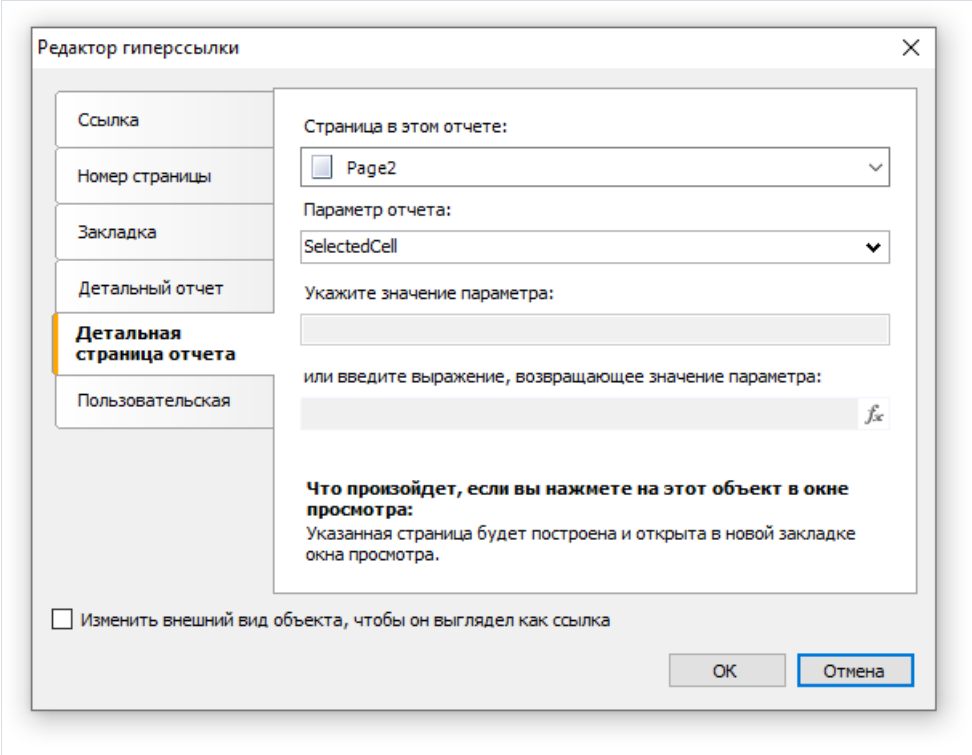
При создании параметра учитывайте следующие моменты:

- родительский параметр настраивать никак не нужно. Назовите его понятным для вас именем, и все;
- родительский параметр должен иметь столько вложенных параметров, сколько значений передается из матрицы. В данном случае передается два значения;
- порядок вложенных параметров соответствует порядку передаваемых из матрицы значений. Напоминаем, что сначала передаются значения колонок, затем строк матрицы. В нашем случае в первый параметр будет передан год, во второй - имя сотрудника;
- вложенные параметры можно называть как угодно, но желательно давать им имена, совпадающие с именами полей из матрицы;
- очень важно правильно настроить тип данных для каждого вложенного параметра. Тип данных должен соответствовать значению, которое передается в параметр. В нашем случае первый параметр (год) должен быть целого типа (`Int32`), второй (имя сотрудника) - строкового (`String`).

Итак, после того как мы прояснили все необходимые моменты, приступим к созданию отчета. Выделите ячейку матрицы и вызовите редактор гиперссылки:

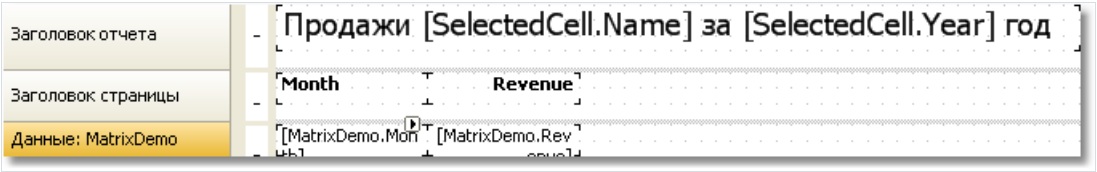


В настройке гиперссылки вам нужно указать в качестве параметра родительский параметр (в нашем примере - `SelectedCell`):



FastReport сам передаст значения во вложенные параметры `SelectedCell.Year` и `SelectedCell.Name`. Эти значения будут преобразованы в тип данных, указанный в настройке параметра - вот почему так важно правильно настроить тип данных параметра.

Детальный отчет располагается на отдельной странице основного отчета и использует тот же источник данных, что и матрица - MatrixDemo:



Осталось отфильтровать строки данных, чтобы показать продажи выбранного сотрудника за выбранный год. Для этого откройте редактор бэнда "Данные" и укажите следующее условие фильтрации:

```
[MatrixDemo.Year] == [SelectedCell.Year] && [MatrixDemo.Name] == [SelectedCell.Name]
```

Отчет готов. Запустите его на выполнение и щелкните на верхней левой ячейке. Откроется детальный отчет,

который содержит следующие данные:

Продажи Andrew Fuller за 1999 год

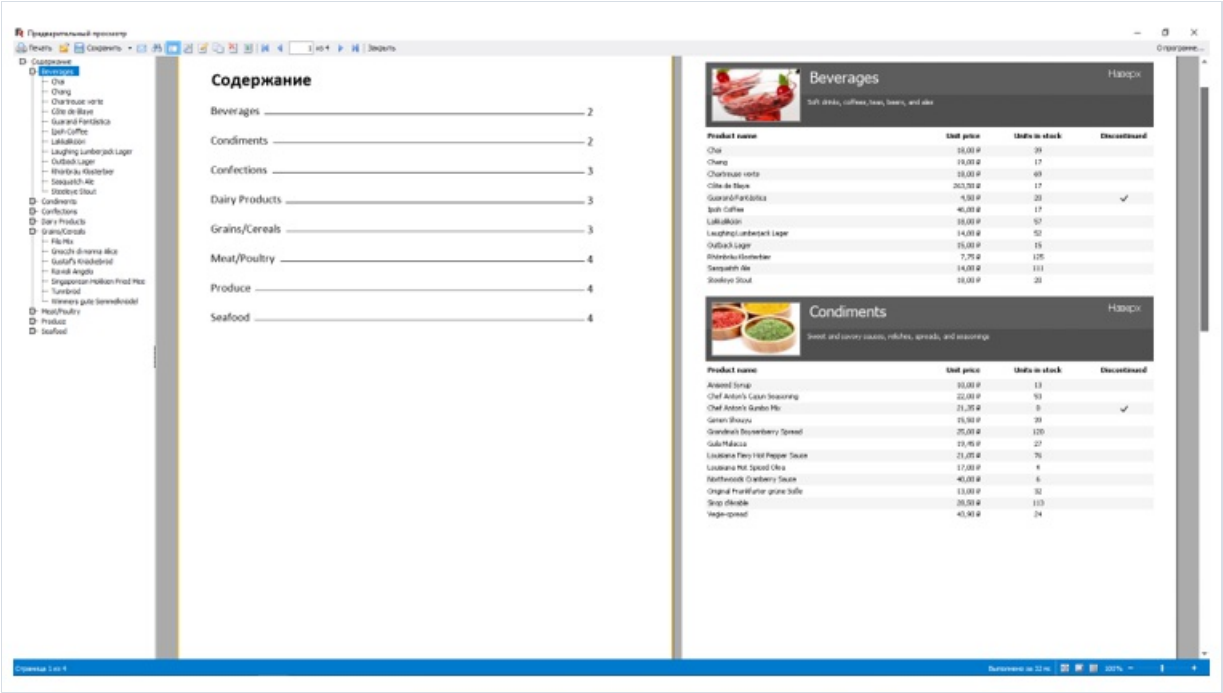
Month	Revenue
10	1 900,00р.
11	2 000,00р.

Как видно, сумма значений соответствует значению ячейки матрицы, на которой мы щелкнули.

Пример 4. Отчет с оглавлением, навигацией и структурой

В этом примере рассмотрим построение отчета, который имеет следующие особенности:

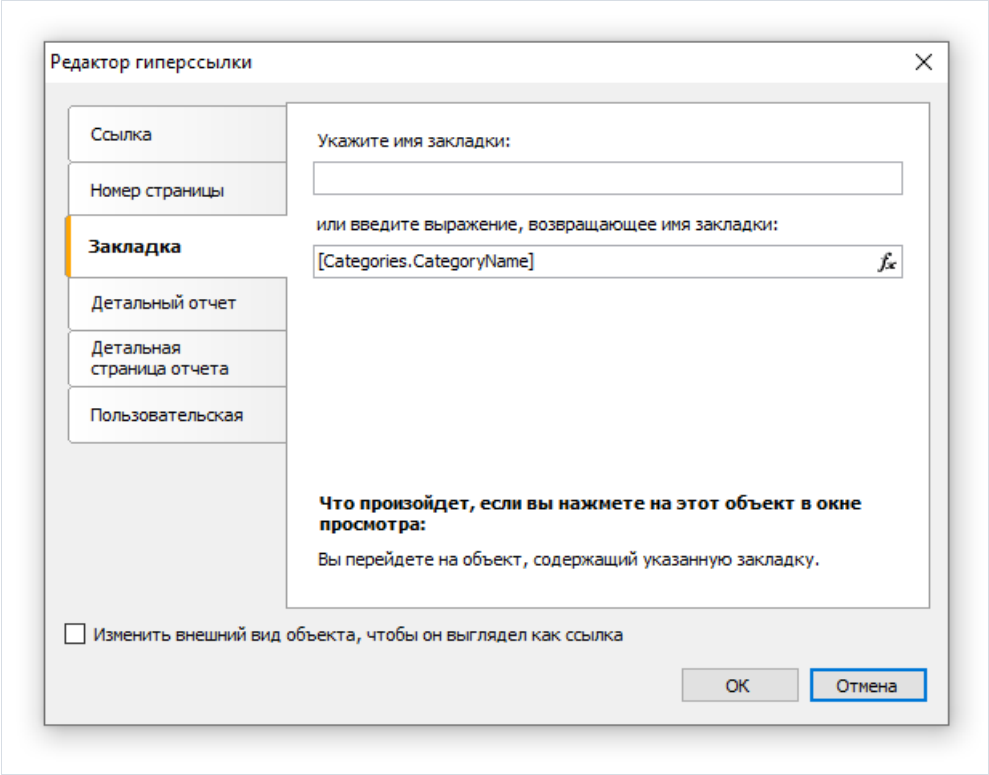
- на первой странице отчета располагается оглавление, или "Содержание", на элементы которого можно нажимать для перехода к нужной странице;
- сбоку в окне просмотра отображается структура отчета, на элементы которой также можно нажимать.



Отчет будет использовать данные из таблиц Categories и Products. В содержании будет напечатан список категорий, в основной части отчета - список категорий и продуктов. Шаблон отчета будет состоять из двух страниц: на первой странице будем печатать содержание, на второй - основную часть отчета.

Сначала займемся оглавлением. Создайте новый отчет и добавьте в него источники данных Categories и Products. Подключите бэнд "Данные" к таблице Categories и расположите объекты следующим образом:

Чтобы сделать элементы содержания интерактивными, настройте их свойство "Гиперссылка":



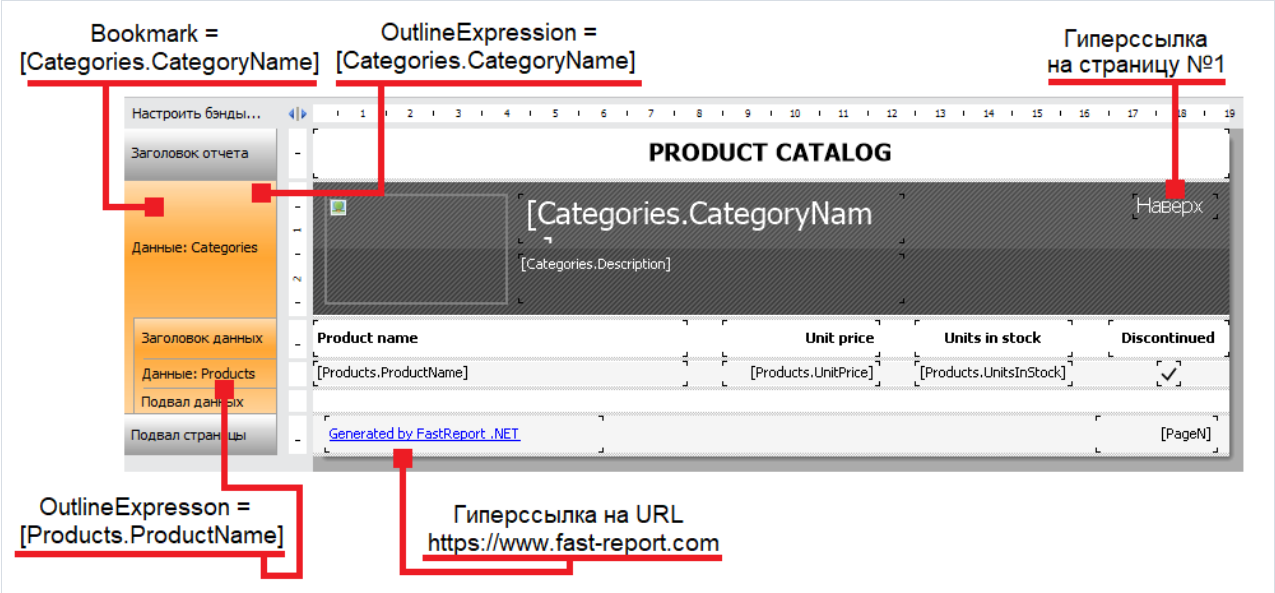
В качестве закладки укажите имя категории. Сама закладка (свойство `Bookmark`) будет определена нами позже у объектов на второй странице отчета.

Для того чтобы напечатать в содержании номер страницы, на которую мы ссылаемся с помощью закладки, нужно сделать следующее:

- включить у отчета двойной проход (это можно сделать в меню "Отчет|Свойства..."). Это нужно сделать, потому что страница с содержанием находится перед остальными страницами. В момент печати содержания еще не известно, на каких страницах будут напечатаны категории;
- использовать функцию `Engine.GetBookmarkPage`, которая возвращает номер страницы по имени закладки. В данном случае имя закладки содержится в поле `[Categories.CategoryName]`, поэтому вызов функции выглядит так:

```
[Engine.GetBookmarkPage([Categories.CategoryName])]
```

На второй странице отчета располагается отчет типа master-detail следующего вида:



Нам надо настроить закладку, на которую будет совершен переход при нажатии на элемент оглавления. Для этого выделите первый бэнд "Данные" и в его свойстве `Bookmark` укажите следующее выражение:

```
[Categories.CategoryName]
```

Для настройки структуры отчета сделайте следующее:

- выделите первую страницу (сам объекта "Страница"). Это можно сделать, переключившись на страницу;
- в окне "Свойства" задайте следующее значение свойства `OutlineExpression`:

```
"Содержание"
```

- переключитесь на вторую страницу отчета;
- выделите первый бэнд "Данные" и настройте его свойство `OutlineExpression`:

```
[Categories.CategoryName]
```

- выделите второй бэнд "Данные" и настройте его свойство `OutlineExpression`:

```
[Products.ProductName]
```

Наследование отчетов

При разработке отчетов мы часто сталкиваемся с ситуацией, когда в каждом отчете встречаются одни и те же данные - реквизиты предприятия, логотипы, одно и то же оформление бланков и т.п. Теперь представьте себе, что таких отчетов - не один десяток, и возникает необходимость что-то поправить (например, поменялись реквизиты). Придется открывать каждый отчет и вносить в него исправления. Для избежания подобных ситуаций можно использовать наследование отчетов. Что это такое?

Допустим, у нас есть какие-то общие для всех отчетов элементы. Это может быть шапка отчета с реквизитами и логотипом, подвал страницы. Их оформление стандартно для вашего предприятия и не меняется от отчета к отчету. Такие общие элементы можно вынести в отдельный файл отчета (базовый отчет). В FastReport есть средства, которые позволяют создать новый отчет на основе базового отчета. При этом новый отчет будет содержать все элементы базового плюс свои собственные.

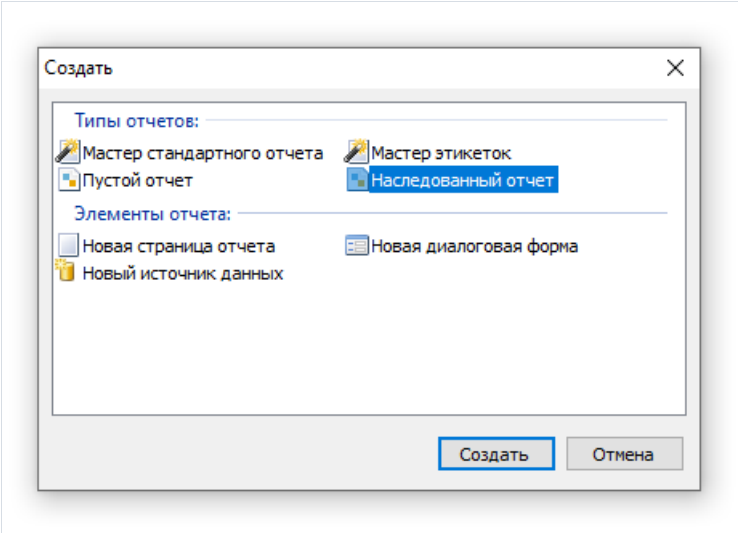
Как такой подход поможет сэкономить время при масштабных изменениях? Очень просто: менять придется только базовый отчет! Все остальные отчеты, которые наследованы от базового, автоматически "подхватят" изменения. Такое поведение обеспечивается механизмом наследования - фактически, при открытии наследованного отчета сначала загружается базовый отчет.

Создание отчета

Для использования наследования необходимо:

- создать базовый отчет и сохранить его в файл;
- создать отчет, наследованный от базового.

Для того чтобы создать наследованный отчет, выберите пункт меню "Файл|Новый..." и в окне выберите "Наследованный отчет":



При этом вам будет предложено выбрать файл базового отчета, который к этому моменту должен быть создан. После этого базовый отчет будет загружен в дизайнер, и в него можно будет внести необходимые изменения. Как видно, объекты базового отчета помечены значком "замок":

Это означает, что такой объект нельзя удалить, переименовать или переместить на другой бэнд.

Вы можете добавлять новые объекты или бэнды, изменять оформление, размеры и позицию уже существующих объектов. После того как вы внесли все изменения, сохраните отчет.

Изменение базового отчета

Посмотрим, что произойдет, если изменить базовый отчет. Вы можете:

- удалить объект в базовом отчете. Этот объект также будет удален из наследованного отчета;
- добавить объект в базовый отчет. Этот объект добавится в наследованный отчет автоматически. Учтите, что место, куда был добавлен объект в базовом отчете, может быть занято в наследованном отчете;
- изменить размеры, позицию, текст, оформление объекта. Все изменения будут отражены в наследованном отчете при условии, что этот объект в наследованном отчете не изменялся.

Последний пункт требует пояснений. Рассмотрим два примера использования. В первом примере:

- создадим базовый отчет с объектом Text1;
- создадим наследованный отчет, и, не изменяя позиции объекта Text1, сохраним отчет;
- в базовом отчете сдвинем объект Text1;
- при открытии наследованного отчета мы увидим, что объект Text1 также сдвинулся.

Во втором примере:

- создадим базовый отчет с объектом Text1;
- создадим наследованный отчет;
- сдвинем объект Text1 и сохраним отчет;
- в базовом отчете сдвинем объект Text1;
- при открытии наследованного отчета мы увидим, что объект Text1 не сдвинулся.

Это произошло потому, что мы изменили объект базового отчета в наследованном. Эти изменения были записаны в файл наследованного отчета. Теперь все, что мы будем делать с исходным объектом в базовом отчете, будет игнорировано в наследованном. В данном примере будет игнорирована позиция объекта Text1. Все остальные изменения свойств (например, изменение размера) по-прежнему будут отражаться в наследованном отчете.

Принцип работы этого механизма станет понятен, если заглянуть в содержимое файла наследованного отчета. Например, в таком виде сохраняется объект базового отчета, который не менялся в наследованном:

```
<inherited Name="Text1"/>
```

А в таком - при изменении позиции объекта:

```
<inherited Name="Text1" Left="255.15" Top="28.35"/>
```

При открытии наследованного отчета в объект Text1 будут загружены все свойства, определенные в базовом отчете, плюс те, что изменились в наследованном.

Ограничения

Наследование отчетов разрабатывалось для одной цели: вынести общие элементы отчетов, такие как заголовки и подвалы, в отдельные файлы и использовать их при построении новых отчетов. В связи с этим, не рекомендуется использовать наследование для выполнения более сложных задач, например:

- наследовать отчет от другого наследованного отчета (т.е. применять двойное наследование);
- использовать в базовом отчете сложные объекты типа "Таблица" и "Матрица";
- использовать скрипт в базовом отчете;
- использовать параметры в базовом отчете.

Отчеты с диаграммами

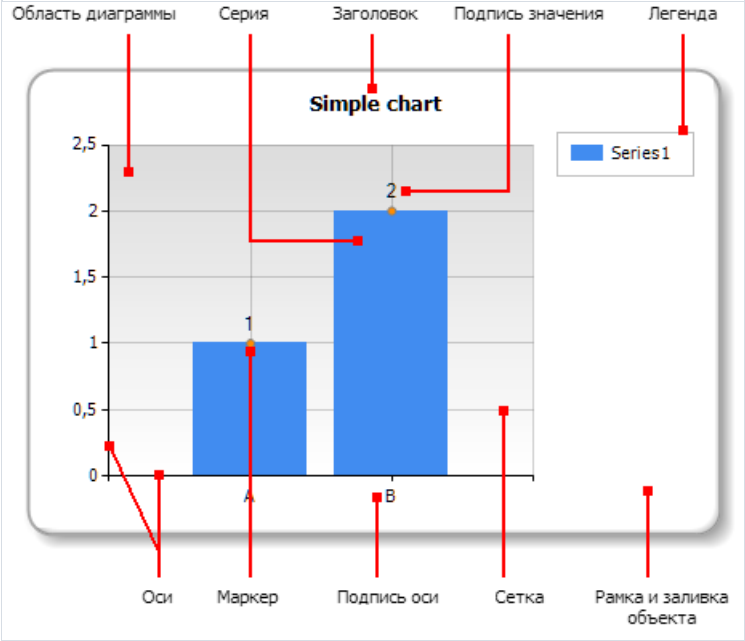
Для построения диаграмм в FastReport используется библиотека `System.Windows.Forms.DataVisualization`.

Обзор возможностей библиотеки в кратком изложении:

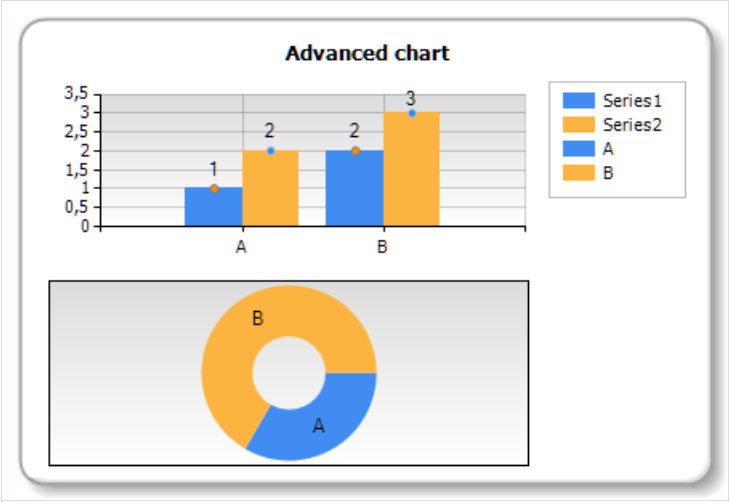
- более 30 видов диаграмм (гистограммы, области, линии, пузырьки, круговые, лепестковые, финансовые, пирамидальные, диапазоны);
- поддержка 3D;
- поддержка нескольких серий разных типов в одной диаграмме;
- полное управление внешним видом и поведением каждого элемента диаграммы.

Элементы диаграммы

Диаграмма Microsoft Chart состоит из следующих элементов:



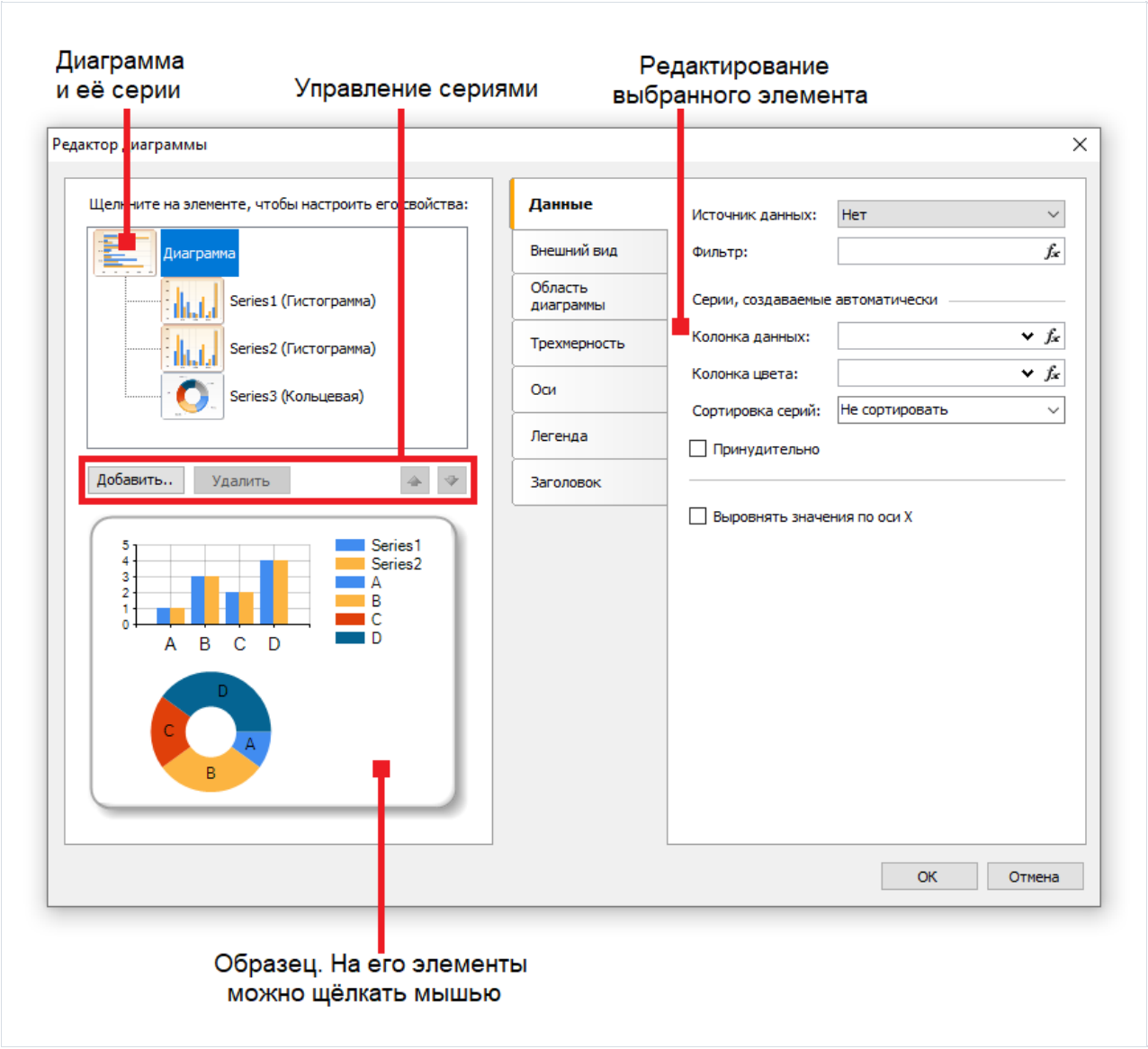
Одна диаграмма может иметь одну или несколько областей диаграммы. В каждой области может находиться одна или несколько серий. На рисунке ниже приведен пример диаграммы, которая содержит две области (в первой области - две серии, во второй - одна):



Следует отметить, что некоторые типы серий (например, круговые) требуют, чтобы им была выделена отдельная область диаграммы.

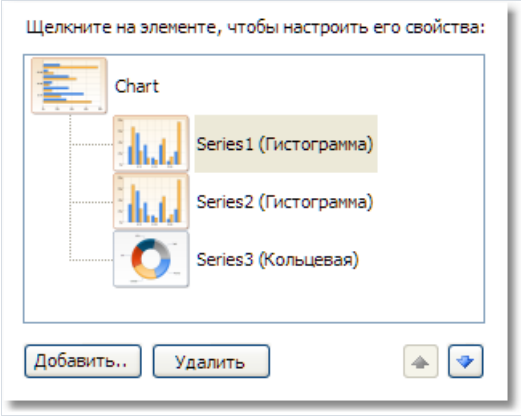
Редактор диаграммы

Объект "Диаграмма" содержит огромное количество настроек, которыми можно управлять, вызвав редактор объекта. Для этого сделайте двойной щелчок мышью на объекте или выберите пункт "Редактировать..." в его контекстном меню:

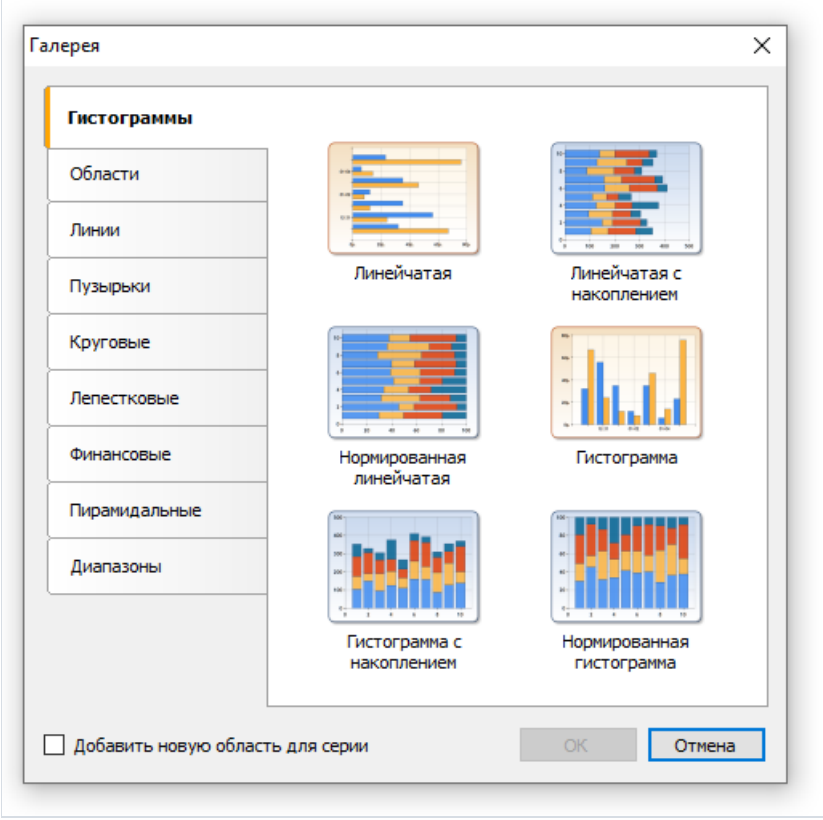


Управление сериями

Объект "Диаграмма" может содержать одну или несколько серий. Список серий отображается в левом верхнем углу редактора:



Для добавления новой серии нажмите кнопку "Добавить...". Будет показано окно - галерея доступных типов серий:



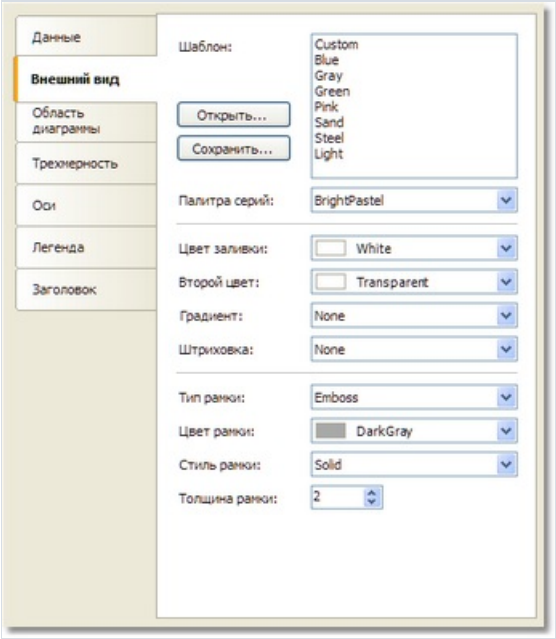
Выберите нужную категорию, а в ней - нужный тип серии. Если серию надо разместить в отдельной области диаграммы, включите флажок "Добавить новую область для серии". Для некоторых типов серий (круговые, лепестковые, финансовые, пирамидальные) новая область создается по умолчанию, независимо от выбранной вами настройки.

Для удаления серии выделите ее в списке и нажмите кнопку "Удалить". Чтобы изменить порядок серий, используйте кнопки "Вверх" и "Вниз".

Настройка внешнего вида

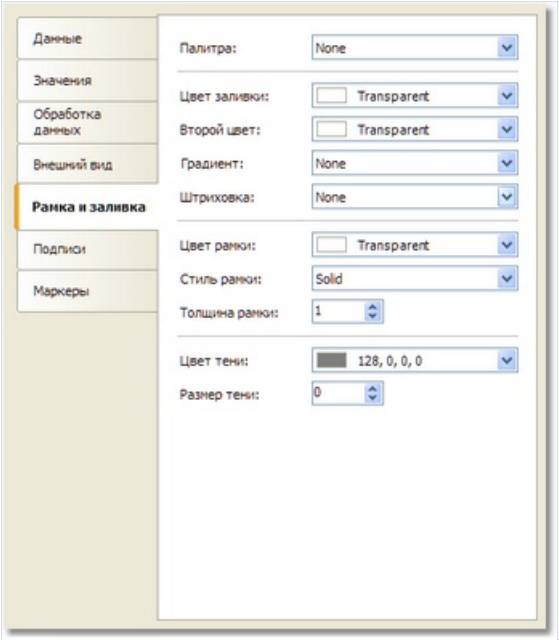
С помощью редактора вы можете настраивать внешний вид каждого элемента диаграммы. Все настройки (а их общее количество свыше 100) разбиты на несколько категорий. Часть из них относится к объекту "Диаграмма", другая часть - к серии.

При выборе элемента "Диаграмма" в списке серий, будут доступны следующие настройки:



- "Внешний вид" - рамка и заливка;
- "Область диаграммы" - рамка, заливка, тень;
- "Трехмерность" - настройки трехмерного вида;
- "Оси" - настройки внешнего вида оси, ее заголовка, подписей, сетки, насечек, дополнительных подписей, индикаторных полос;
- "Легенда" - стиль легенды, ее расположение, рамка, заливка, тень, шрифт;
- "Заголовок" - стиль текста, расположение, рамка, заливка, тень, шрифт.

При выборе серии в списке серий, будут доступны следующие настройки:



- "Внешний вид" - настройки, специфичные для выбранного типа серии;
- "Рамка и заливка" - рамка и заливка для значений серии;
- "Подписи" - подписи к значениям. Здесь можно выбрать тип подписи, шрифт, цвет и заливку;
- "Маркеры" - маркеры значений. Здесь можно выбрать тип маркера, его цвет и рамку.

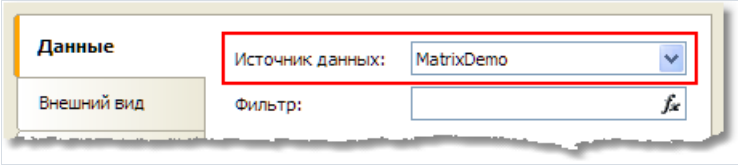
Подключение к данным

Заполнить объект данными можно с помощью нескольких способов:

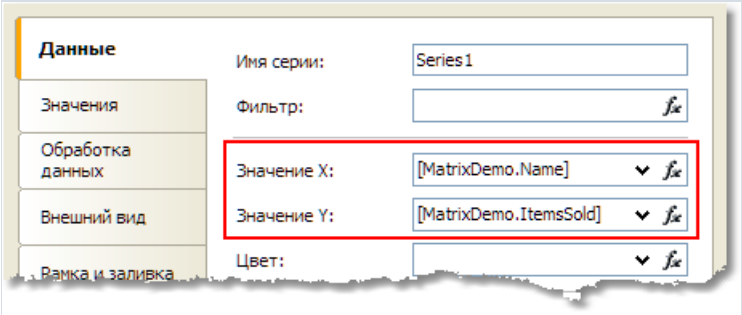
1. Подключить объект к источнику данных. Для этого в настройках элемента "Диаграмма" указывается источник данных, а в настройках серии - поля источника данных.
2. Указать список значений для серии вручную.
3. Заполнить объект данными, используя скрипт.

Для того чтобы подключить объект к источнику данных, проделайте следующие шаги:

- в списке серий выберите элемент "Диаграмма";
- переключитесь на закладку "Данные";
- выберите нужный источник данных:

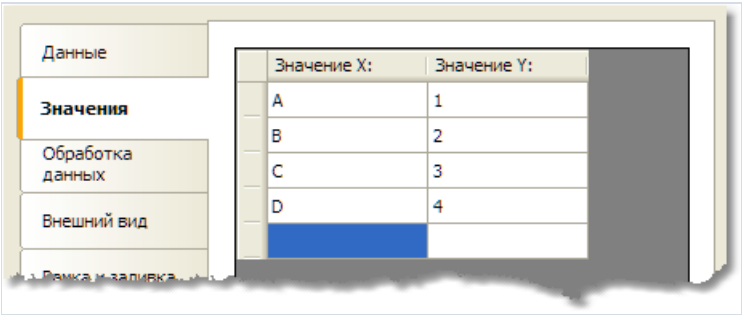


- если необходимо, укажите условие фильтрации в поле "Фильтр". Фильтр будет применен ко всем сериям;
- в списке серий выберите серию;
- переключитесь на закладку "Данные";
- выберите колонки данных для значений серии. В зависимости от типа серии, количество значений может варьироваться. Как правило, это два значения - для осей X и Y:



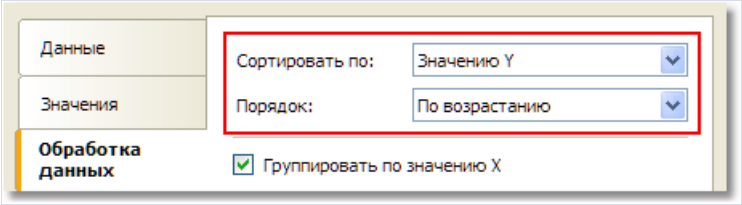
- если необходимо, укажите дополнительное условие фильтрации в поле "Фильтр". Фильтр будет применен к текущей серии;
- в поле "Цвет" можно указать колонку данных, возвращающую цвет значения.

Если подключение к источнику данных не требуется, можно указать список значений вручную. Для этого выберите серию в списке серий и переключитесь на закладку "Значения". Заполните таблицу нужными значениями:



Сортировка данных

По умолчанию диаграмма отображает данные в том порядке, в каком они расположены в источнике данных. Вы можете изменить порядок сортировки, выбрав серию в списке серий и переключившись на закладку "Обработка данных":



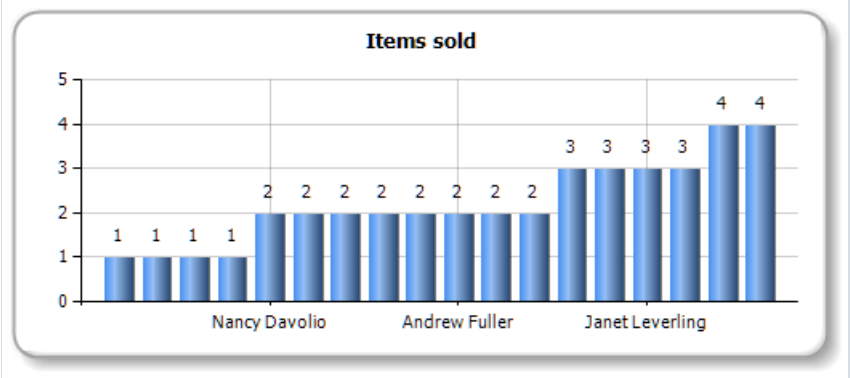
Вы можете выбрать один из режимов сортировки - не сортировать, по значению X или по значению Y, а также порядок сортировки (по возрастанию, по убыванию).

Группировка данных

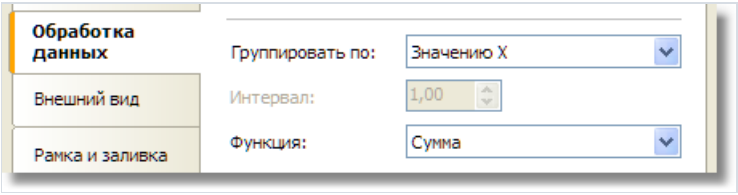
Часто возникают ситуации, когда в диаграмму попадают данные с одинаковым значением по оси X. Например, в таблице MatrixDemo, которая используется для демонстрации возможностей диаграмм, данные представлены в таком виде:

Name	Year	Month	ItemsSold	Revenue
Andrew Fuller	2002	1	2	1800
Andrew Fuller	1999	10	2	1900
Andrew Fuller	1999	11	2	2000
Andrew Fuller	2000	2	2	2100
Janet Leverling	1999	10	3	3000
Janet Leverling	1999	11	3	3100
Janet Leverling	2000	3	3	3200
...				

При попытке построить диаграмму, которая показывает продажи сотрудника (по оси X - поле Name, по оси Y - поле ItemsSold) получится примерно такой результат:

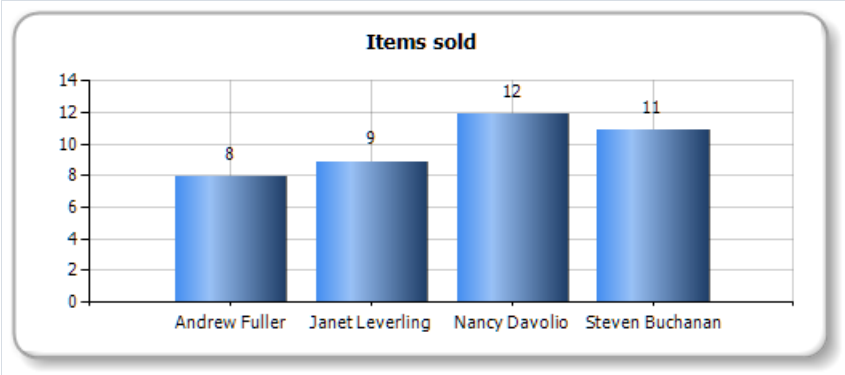


В диаграмму попали данные с одинаковыми значениями по оси X, что дало в итоге неправильный результат. Чтобы исправить ситуацию, надо сгруппировать одинаковые значения X (в данном случае - имена сотрудников), просуммировав результаты. Для этого в редакторе объекта выберите серию и переключитесь на закладку "Обработка данных". Выберите тип группировки - по значению X, и выберите функцию группировки - "Сумма":



В результате сотрудники с одинаковыми именами будут сгруппированы в одно значение, их данные продаж

- просуммированы. Получится следующий результат:

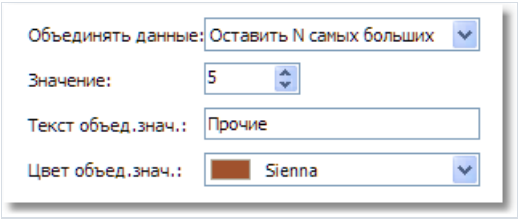


Объединение значений

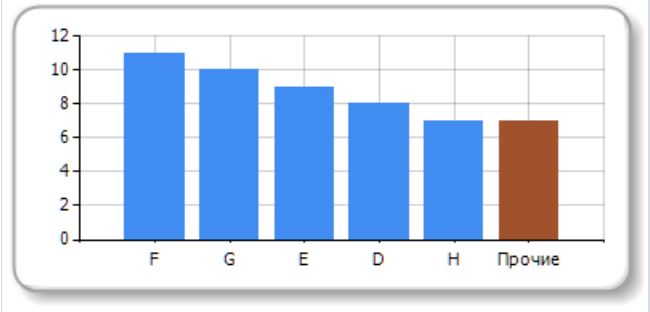
Этот инструмент обработки данных позволяет объединить несколько значений серии в одно, используя те или иные критерии. Вы можете выбрать один из следующих вариантов объединения:

Вариант	Описание
Оставить N самых больших	В серии остается N самых больших значений (количество N задается). Остальные значения суммируются и выводятся в виде одного значения (заголовок для значения задается, обычно это "Прочие").
Оставить N самых маленьких	То же, но в серии остается N самых маленьких значений. Если текст заголовка для собранного значения не задан, это значение не выводится.
Меньше чем значение	Значения серии, меньшие заданного значения, объединяются в одно.
Меньше чем процент	Значения серии, меньшие заданного процента от общего количества, объединяются в одно.
Больше чем значение	Значения серии, большие заданного значения, объединяются в одно.
Больше чем процент	Значения серии, большие заданного процента от общего количества, объединяются в одно.

Так, чтобы показать 5 самых больших значений, настройте серию следующим образом:

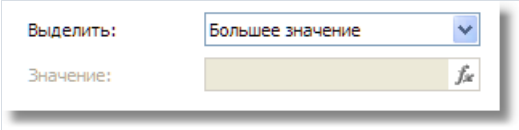


В результате получится примерно следующее:



Выделение значений

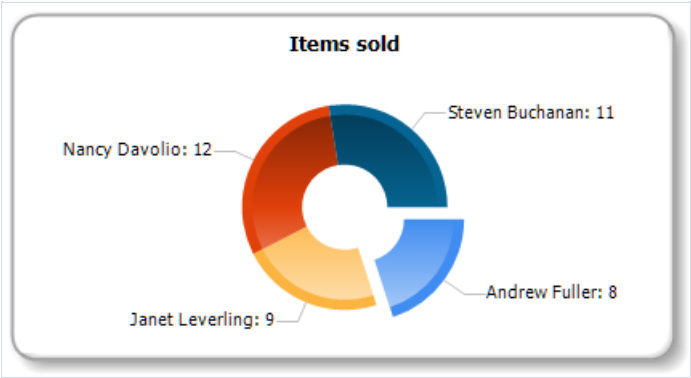
Для круговых диаграмм можно настроить выделение сегментов диаграммы. Для этого выберите серию в списке серий и переключитесь на закладку "Обработка данных":



Доступны следующие режимы выделения: большее значение, меньшее значение, указанное значение. При выборе последнего режима предлагается указать значение, которое нужно выделить. Это может быть любое корректное с точки зрения FastReport выражение. Например, чтобы выделить сотрудника Andrew Fuller, укажите следующее значение:

```
"Andrew Fuller"
```

При этом диаграмма будет выглядеть так:



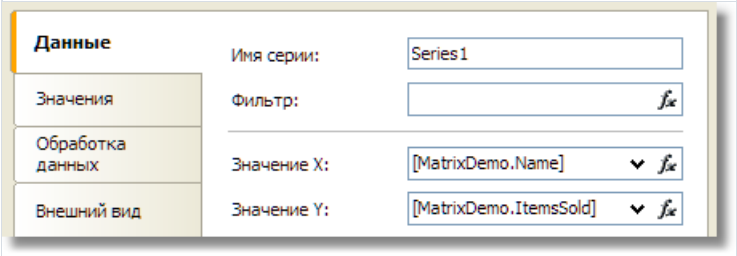
Настройка авто-серий

Диаграмму можно настроить таким образом, чтобы она создавала серии автоматически, в зависимости от имеющихся данных. Для создания авто-серий нужно проделать следующее:

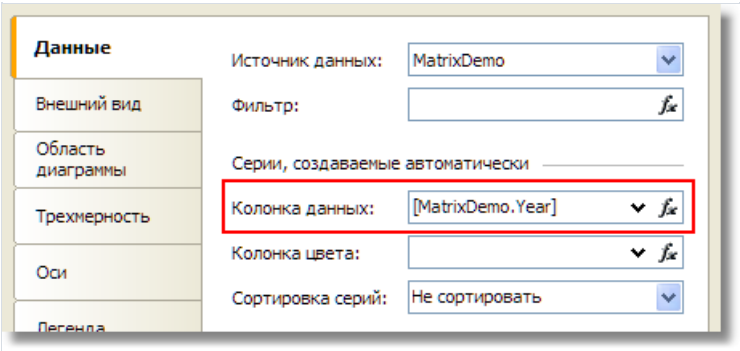
- создать и настроить одну серию. Эта серия будет использована как образец для создания автоматических серий;
- у объекта "Диаграмма" указать колонку данных для авто-серии. Значение этой колонки будет являться именем серии. Если серии с таким значением еще не существует, будет автоматически создана новая серия.

Продемонстрируем создание авто-серий на примере. Используем таблицу MatrixDemo, чтобы получить график продаж сотрудников с разбивкой по годам. Одна серия будет представлять один год. Для этого проделайте следующие шаги:

- подключите диаграмму к источнику данных MatrixDemo;
- создайте одну серию и настройте ее данные:

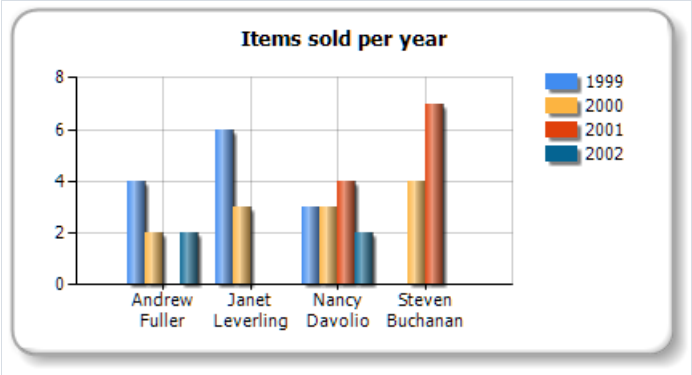


- на закладке "Обработка данных" выберите тип группировки - по значению X, так как в нашем источнике данных есть строки с одинаковыми именами сотрудников;
- выделите диаграмму в списке серий и настройте создание авто-серий на закладке "Данные":



- в разных сериях может оказаться разное количество значений (не у всех сотрудников есть продажи за все года). Чтобы выровнять значения в сериях, включите флажок "Выровнять значения по оси X".

В результате получится диаграмма следующего вида:


Items sold per year
8
6
4
2
0
Andrew Fuller
Janet Leverling
Nancy Davolio
Steven Buchanan
1999
2000
2001
2002

Интерактивные диаграммы

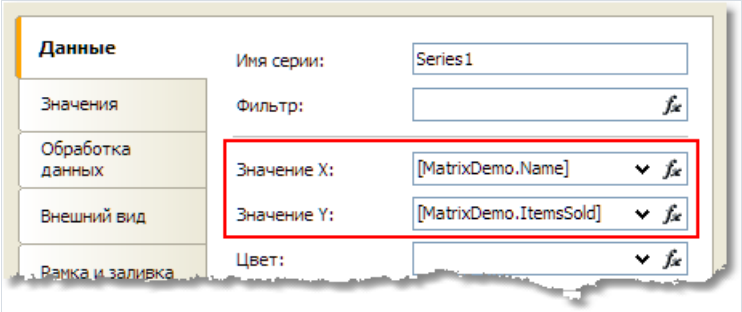
Диаграмма, как и остальные объекты FastReport, поддерживает интерактивность. Вы можете настроить диаграмму таким образом, чтобы при щелчке мышью на значении серии открывался новый отчет с детальными данными по выбранному значению. Для этого надо настроить свойство "Гиперссылка", как описано в разделе "Интерактивные отчеты". Значение в гиперссылку диаграмма подставляет сама, при щелчке на ее элементе.

Рассмотрим в качестве примера отчет "Interactive Chart" из демо FastReport.

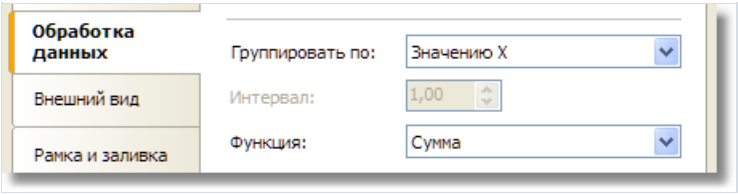
Создайте отчет с двумя страницами. На первой странице будет находиться диаграмма, на второй - отчет типа "Список", который показывает список продаж выбранного сотрудника. Эта страница будет строиться при щелчке на элементе диаграммы.

На первую страницу поместите диаграмму и настройте ее свойства в редакторе:

- выберите диаграмму и укажите источник данных - MatrixDemo;
- выберите серию и укажите значение `X = [MatrixDemo.Name]`, значение `Y = [MatrixDemo.ItemsSold]`:

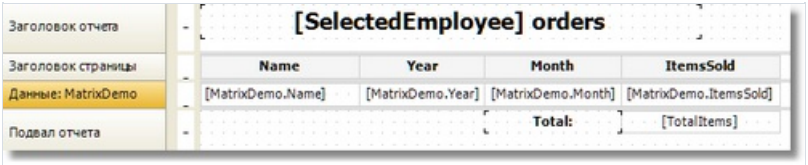


- на закладке "Обработка данных" выберите группировку по значению X:



На второй странице разместите отчет типа "Список":

- в окне "Данные" создайте параметр отчета - `SelectedEmployee`;
- создайте отчет следующего вида:



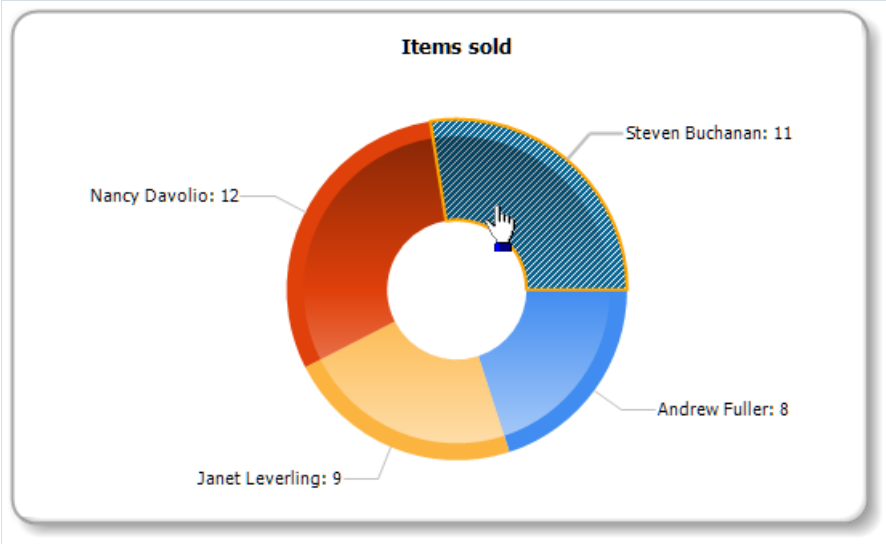
- откройте редактор бэнда "Данные" и укажите фильтр:

```
[MatrixDemo.Name] == [SelectedEmployee]
```

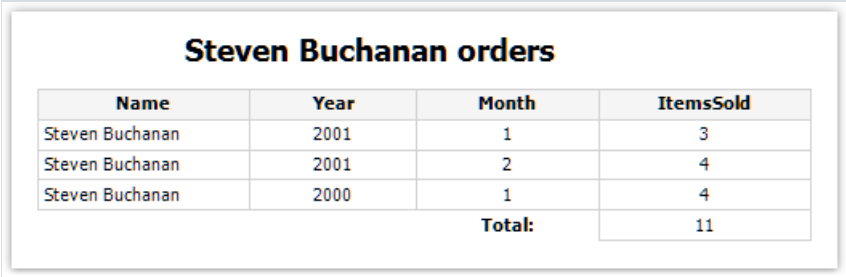
Теперь настройте гиперссылку у объекта "Диаграмма". Для этого:

- вызовите контекстное меню объекта и выберите пункт "Гиперссылка...";
- выберите тип ссылки - "Детальная страница отчета";
- выберите вторую страницу отчета и имя параметра - `SelectedEmployee`.

На этом создание отчета завершено. Запустите отчет и подведите указатель мыши к какому-нибудь значению диаграммы. Значение будет визуально выделено, а курсор мыши поменяет форму:



Если кликнуть на значении, будет построен отчет, содержащий подробные данные о выбранном сотруднике:



Steven Buchanan orders

Name	Year	Month	ItemsSold
Steven Buchanan	2001	1	3
Steven Buchanan	2001	2	4
Steven Buchanan	2000	1	4
		Total:	11

Отчеты с картами

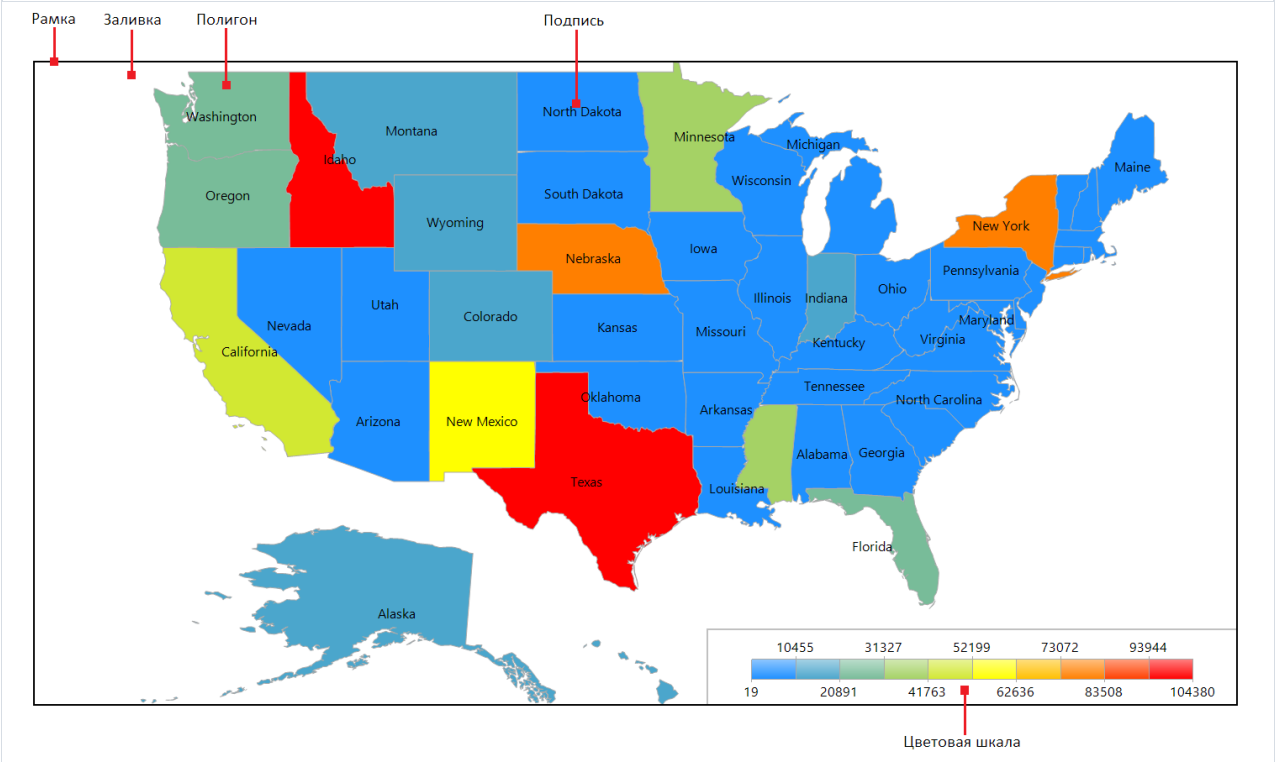
Компонент "Карта" (MapObject) предназначен для отображения двумерных графических карт в формате ESRI shapefile. Подробнее о формате можно прочитать здесь:

http://ru.wikipedia.org/wiki/Shapefile

Для работы необходимы файлы .shp (геометрия) и .dbf (описание).

Элементы карты

Объект "Карта" состоит из следующих элементов:



Один объект "Карта" может отображать один или несколько слоев. Каждый слой содержит отдельную карту.

Управление отображением

В режиме дизайнера и в окне просмотра готового отчета вы можете управлять отображением карты с помощью мыши:

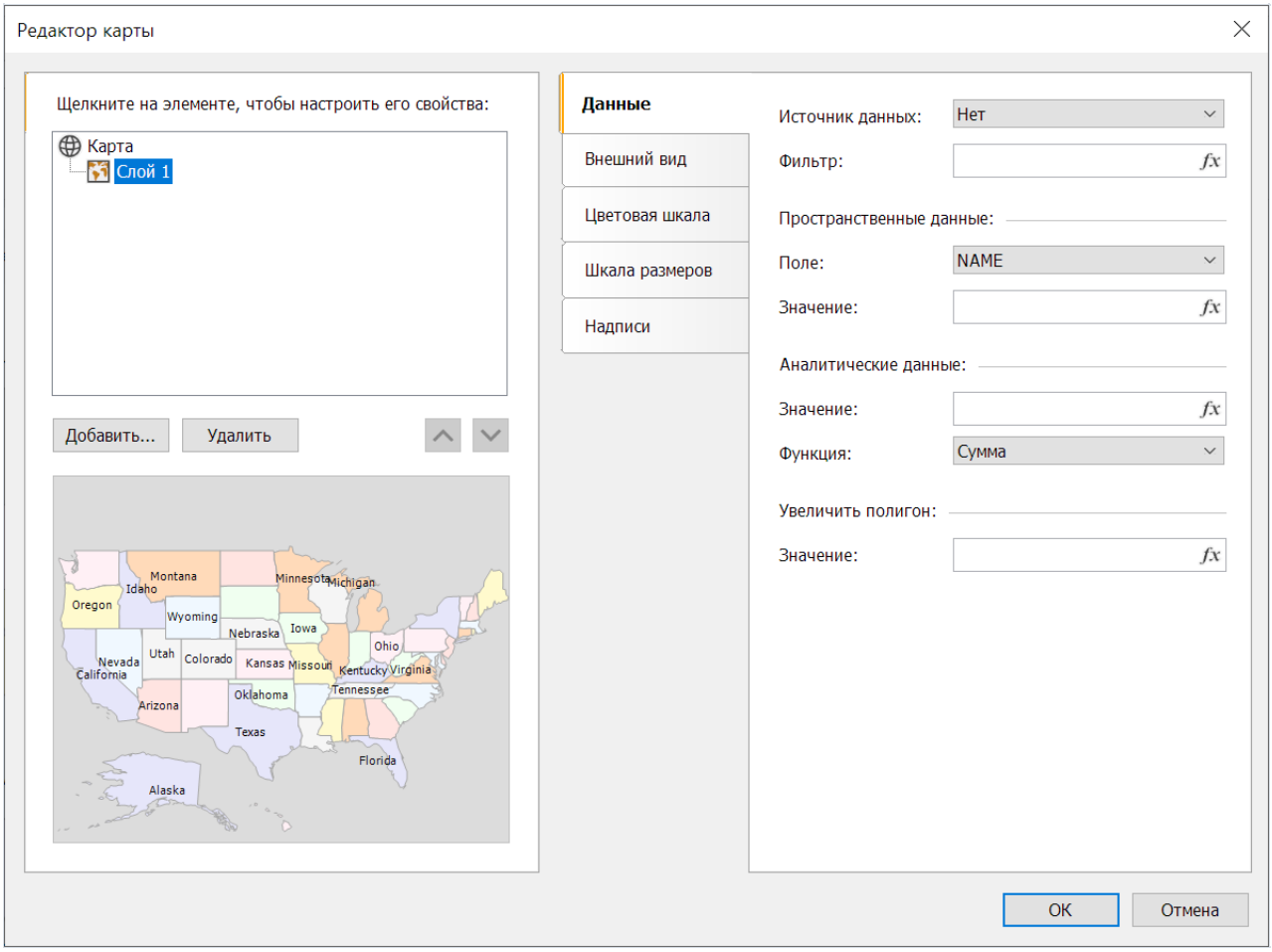
- колесо мыши меняет масштаб карты;
- нажав левую кнопку мыши, можно двигать карту;
- кликнув внутри полигона, можно настроить его свойства в окне "Свойства".

Минимальное и максимальное значения масштабирования задаются в свойствах `MinZoom`, `MaxZoom`. Эти значения можно задать в окне "Свойства".

> В ASP.NET картой управлять нельзя.

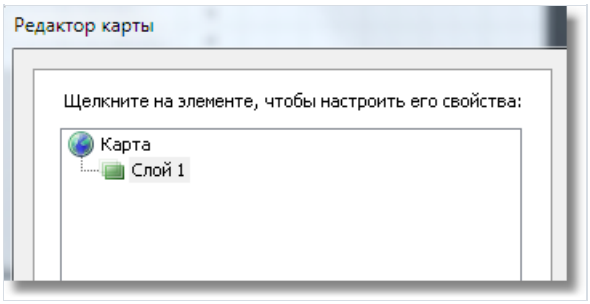
Редактор карты

Объект "Карта" содержит большое количество настроек, которыми можно управлять, вызвав редактор объекта. Для этого сделайте двойной щелчок мышью на объекте или выберите пункт "Редактировать..." в его контекстном меню:

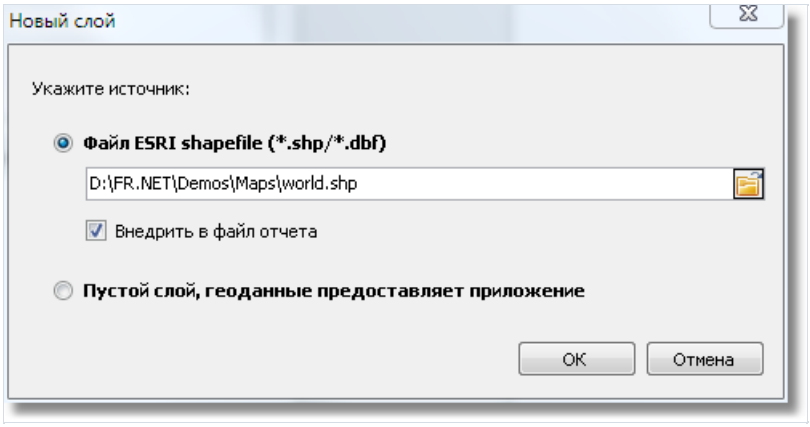


Добавление слоёв

Объект "Карта" может содержать один или несколько слоев. Список слоев отображается в левом верхнем углу редактора:



Для добавления нового слоя нажмите кнопку "Добавить...". Будет показано следующее окно:



На этом шаге нужно выбрать тип слоя:

- карта из файла shapefile (.shp/.dbf). Это наиболее часто используемый тип карт. К примеру, вы можете напечатать карту мира и выделить цветом страны, в которых продажи были больше определенного значения;
- картографические данные из приложения. Ваше приложение должно предоставлять географические координаты (пару значений - широта и долгота), которые будут отображены в виде точки на карте. Точка может иметь подпись, а также быть разного размера и/или цвета, в зависимости от некоторых данных. На практике этот тип карты используется в качестве второго слоя (первый слой, базовый, берется из файлов shapefile). Например, базовый слой отображает карту какой-либо страны, а второй слой - точки с названиями городов, в которых были продажи. Размеры и цвет точки можно настроить таким образом, чтобы был понятен уровень продаж в данном городе.

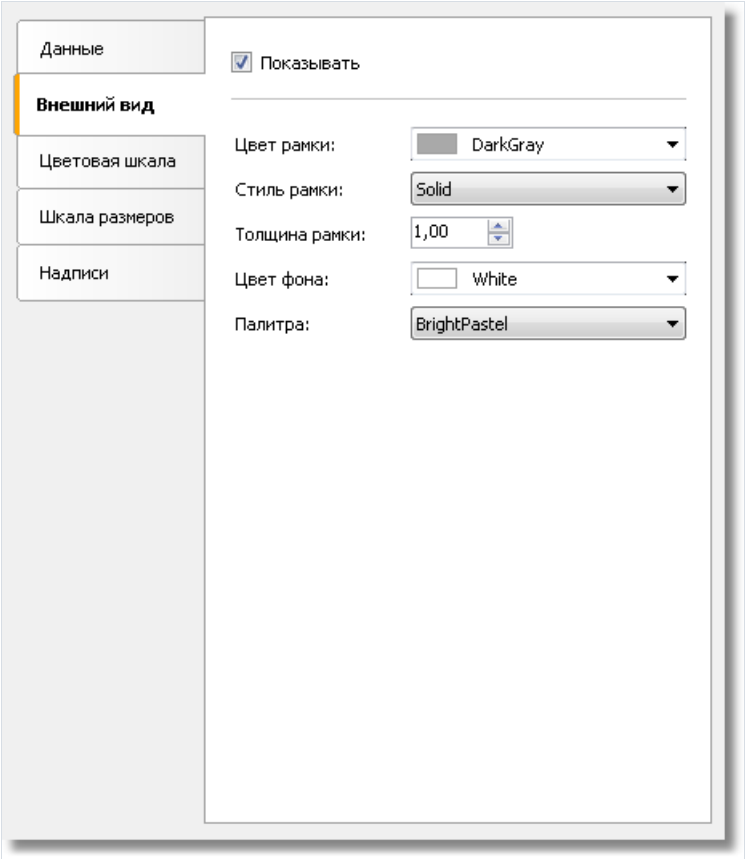
Если вы выбрали слой на основе файлов shapefile, укажите дополнительно, как хранить картографические данные:

- данные внедряются в файл отчета. При этом отчет может сильно увеличиться в размерах.
- файл отчета ссылается на файлы shapefile, внедрения не происходит. Этот режим полезен, если у вас есть несколько отчетов, использующих одни и те же карты.

> Карты большого объема (более 30Мб) или с большим количеством полигонов (более 20000) серьезно замедлят работу отчета.

Настройка внешнего вида

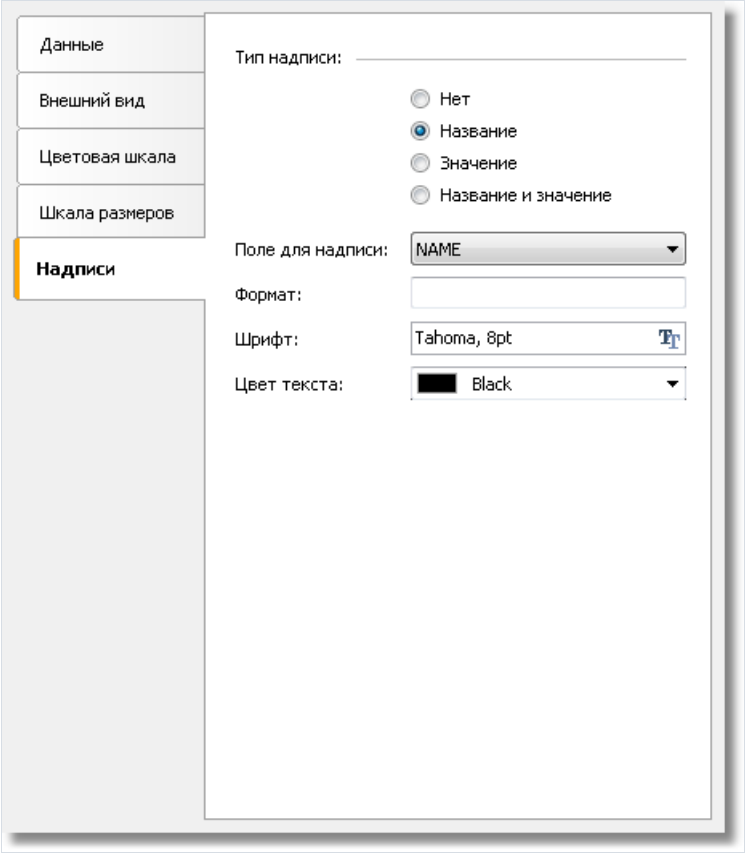
Внешний вид слоя можно настроить, выбрав слой и переключившись на закладку "Внешний вид":



Здесь можно настроить цвет и стиль рамки полигонов карты, а также выбрать цветовую палитру. Если вы настроили выделение полигонов цветом в зависимости от аналитических данных (об этом позднее), то палитра будет игнорирована.

Настройка отображаемых значений

На карте могут отображаться надписи, например, названия стран на карте мира. Вы можете настроить тип и внешний вид надписей на закладе "Надписи":



Если выбран тип слоя - карта из файла shapefile, то необходимо указать поле, из которого будет взята надпись. Как правило, это поле "NAME". Для карты мира, входящей в состав демо-программы FastReport, можно выбирать из следующих полей:

- NAME (например, Russia)
- ABBREV (например, Rus.)
- ISO_A2 (например, RU)
- ISO_A3 (например, RUS)

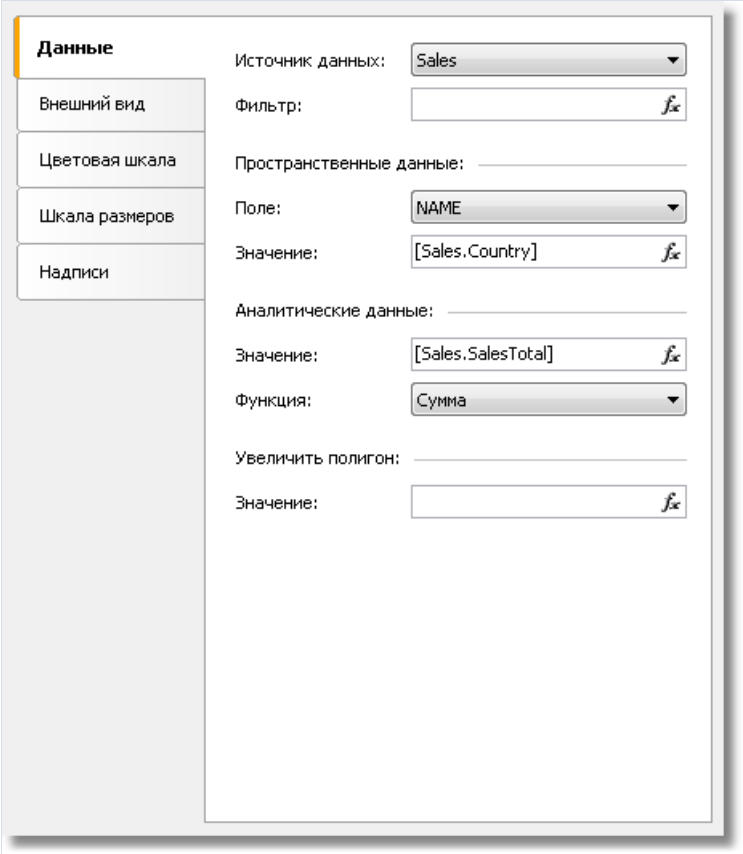
Для других карт список полей будет отличаться.

Если выбран тип слоя - картографические данные из приложения, то в этом окне можно указать минимальное значение масштаба, при котором можно показывать надписи. Значение по умолчанию - 1, что означает, что надписи будут показаны всегда.

Подключение к данным

Большинство отчетов используют объект "Карта" не сам по себе, а для отображения аналитической информации. Например, это может быть объем продаж в разных странах. Для этого слой надо подключить к данным. Сделать это можно в редакторе карты, выбрав слой и переключившись на закладку "Данные". Подключение к данным различается в зависимости от типа слоя (из файла shapefile, или геоданные из приложения):

- если тип слоя указан как карта из shapefile, закладка "Данные" выглядит следующим образом:



В этом случае приложение должно предоставить следующие данные:

- название (например, название страны);
- числовое значение (например, уровень продаж в данной стране).

Допустим, у вас имеется таблица Sales со следующими полями и данными:

Country	SalesTotal
USA	500000
Germany	1200000
Russia	300000

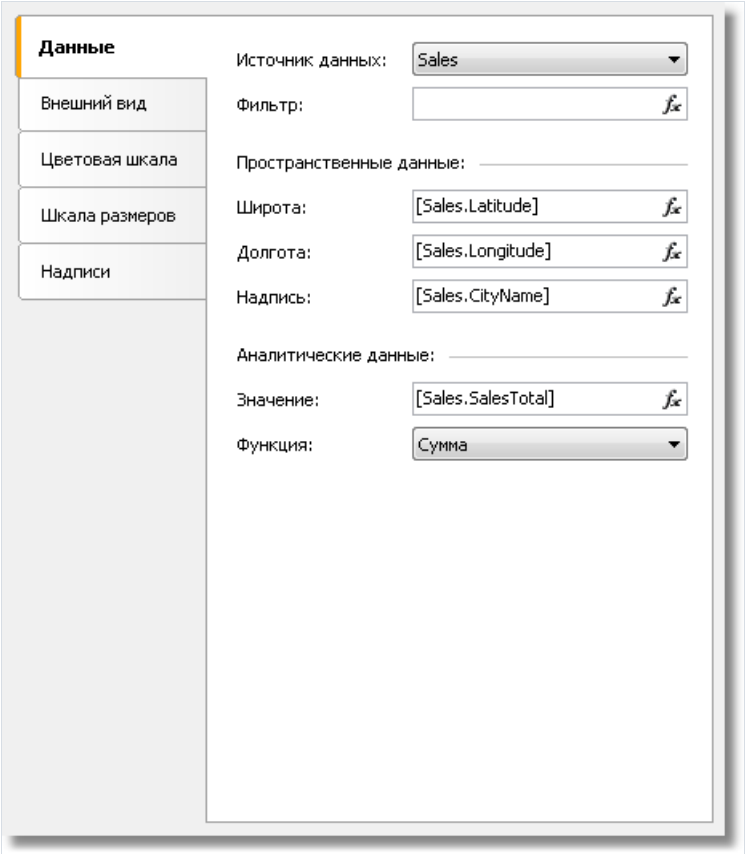
В данном случае нужно настроить данные следующим образом:

- источник данных - Sales;

- пространственные данные, поле - выберите то поле, которое в фале shapefile отвечает за название страны. Как правило, это поле `NAME`;
- пространственные данные, значение - `[Sales.Country]`;
- аналитические данные, значение - `[Sales.SalesTotal]`;
- аналитические данные, функция - "Сумма". Функция используется, если для данной страны есть несколько записей с разными значениями.

Поле "Увеличить полигон" позволяет увеличить полигон с указанным именем на весь размер объекта "Карта". Например, чтобы увеличить страну Россия на карте мира, укажите в этом поле значение "Russia" (с кавычками).

- если тип слоя указан как данные, предоставляемые приложением, закладка "Данные" выглядит следующим образом:



В этом случае приложение должно предоставить следующие данные:

- пространственные данные - широта и долгота;
- название (например, название города);
- числовое значение (например, уровень продаж в данном городе).

Допустим, у вас имеется таблица Sales со следующими полями и данными:

Latitude	Longitude	CityName	SalesTotal
48.13641	11.57753	Munchen	50000
50.94165	6.95505	Koln	36000

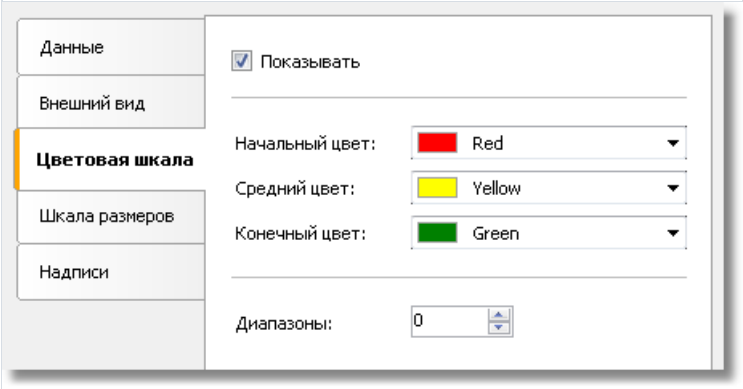
В этом случае нужно настроить данные следующим образом:

- источник данных - Sales;
- пространственные данные, широта - `[Sales.Latitude]`;

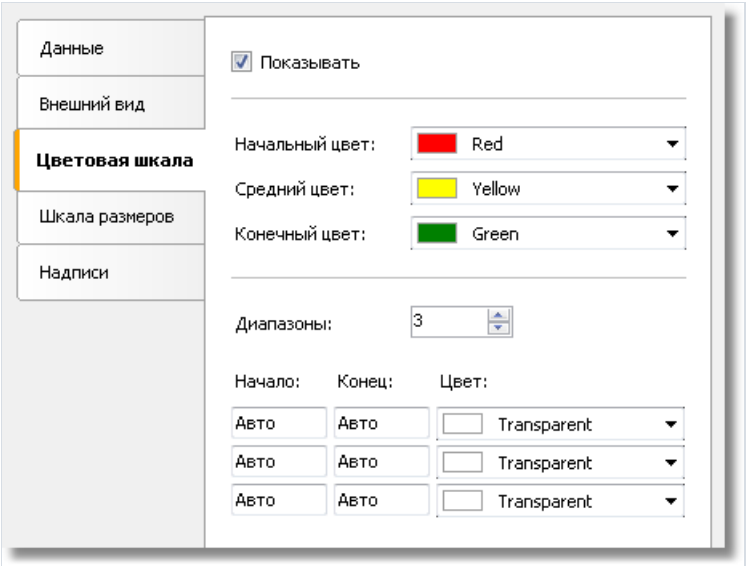
- пространственные данные, долгота - `[Sales.Longitude]`;
- пространственные данные, надпись - `[Sales.CityName]`;
- аналитические данные, значение - `[Sales.SalesTotal]`;
- аналитические данные, функция - "Сумма". Функция используется, если для данного города есть несколько записей с разными значениями.

Выделение данных цветом

После того как слой подключен к данным, возникает вопрос - в каком виде выводить аналитическую информацию (например, объемы продаж в разных странах)? Самый простой способ - настроить отображение надписей так, чтобы помимо названия страны выводились и цифры продаж (см. раздел "Настройка отображаемых значений"). Однако, гораздо более наглядный способ - это раскраска стран в определенные цвета в зависимости от объема продаж. Для этого надо настроить цветовую шкалу. Сделать это можно на закладке "Цветовая шкала":



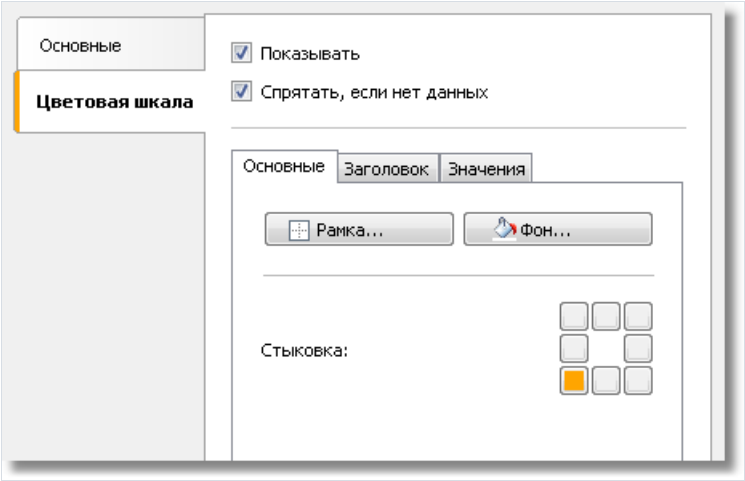
Цветовая шкала представляет собой набор значений: минимальное значение; максимальное значение; цвет. Таких наборов (диапазонов) может быть несколько. Для настройки цветовой шкалы надо указать, сколько диапазонов она содержит, после этого настроить минимальное и максимальное значение в каждом диапазоне, а также цвет:



По умолчанию все значения установлены в "Авто". В этом случае FastReport рассчитает минимальное и максимальное значения для каждого диапазона автоматически, а цвет возьмет из предустановок "Начальный цвет", "Средний цвет" и "Конечный цвет". Этот режим можно использовать в большинстве случаев.

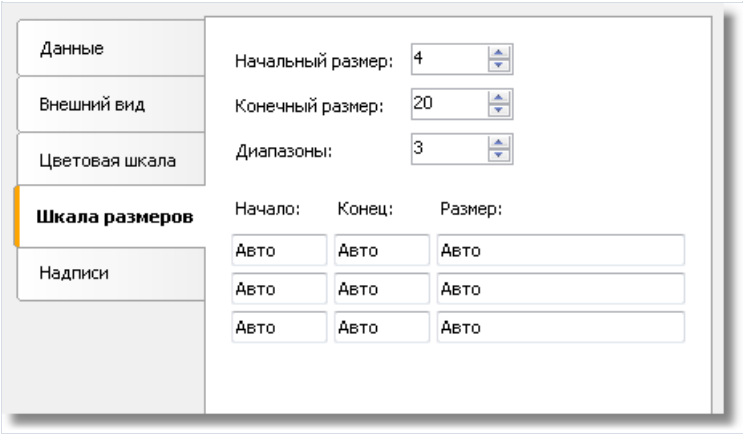
Если цветовая шкала настроена, то в нижней части карты появляется индикатор - полоска из нескольких разноцветных прямоугольников:

Внешний вид и расположение индикатора можно настроить, выбрав в списке слоев элемент "Карта" и переключившись на закладку "Цветовая шкала":



Выделение данных размером

Если тип слоя указан как данные, предоставляемые приложением, то данные будут отображаться в виде точки с надписью. Размер точки можно привязать к данным примерно таким же способом, как это делается при выделении цветом. Сделать это можно на закладке "Шкала размеров":



Шкала размеров представляет собой набор значений: минимальное значение; максимальное значение; размер в пикселах. Таких наборов (диапазонов) может быть несколько. Для настройки шкалы надо указать, сколько диапазонов она содержит, после этого настроить минимальное и максимальное значение в каждом диапазоне, а также размер.

По умолчанию все значения установлены в "Авто". В этом случае FastReport рассчитает минимальное и максимальное значения для каждого диапазона автоматически, а размер возьмет из предустановок "Начальный размер", "Конечный размер".

Работа с данными

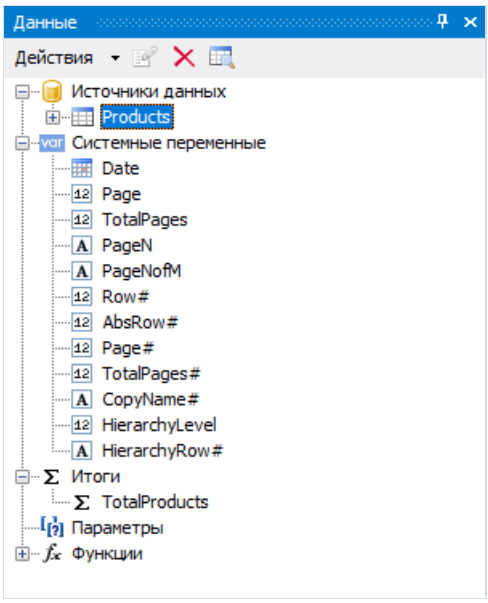
Предназначение любого отчета - это печать данных. В отчете FastReport можно использовать следующие данные:

- источники данных;
- системные переменные;
- итоговые значения;
- параметры отчета;
- функции;
- выражения, содержащие любые из вышеперечисленных данных.

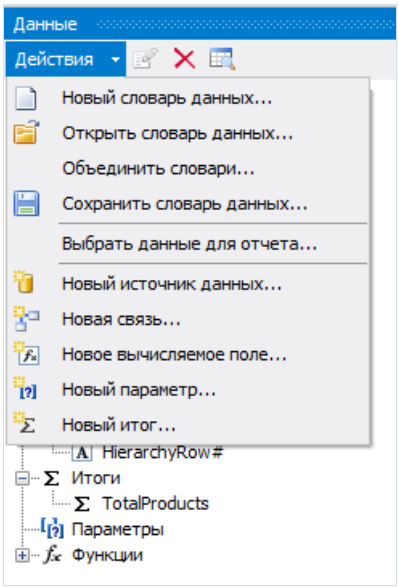
В этой главе будут рассмотрены способы работы со всеми вышеперечисленными данными.

Окно "Данные"

Все данные, которые можно использовать в отчете, доступны из одного места – из окна "Данные". Это окно можно показать, выбрав пункт меню "Данные|Показать окно данных":



Окно "Данные" позволяет управлять всеми элементами данных, а также вставлять элементы данных в отчет путем перетаскивания их на страницу отчета. Все операции выполняются с помощью панели инструментов и меню "Действия":



Часть этих операций дублируется в контекстном меню окна "Данные". Так, если выбрать в окне источник данных, то в контекстном меню будут доступны операции по созданию вычисляемого поля, удаления источника, просмотра его данных.

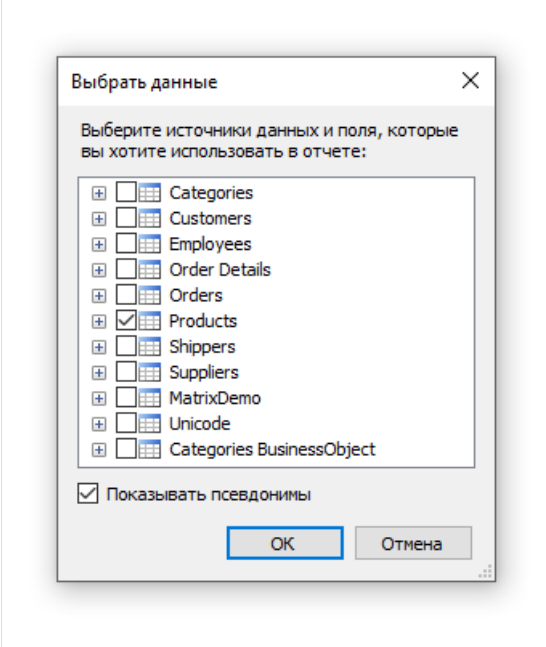
Источники данных

Под источником данных подразумевается таблица БД или запрос на языке SQL. В отчете может быть несколько источников данных. Как правило, для большинства отчетов нужен только один источник. Отчет типа master-detail нуждается в двух источниках, между которыми установлена связь (об этом далее).

Источник данных имеет одно или несколько полей. Каждое поле имеет определенный тип данных. Чтобы посмотреть тип поля, выделите поле в окне "Данные" и откройте окно "Свойства". Тип поля указан в свойстве `DataType`. Иконка рядом с названием поля также помогает определить его тип.

Источник данных для отчета может быть определен двумя способами.

Первый способ – источник данных определен в приложении и зарегистрирован в отчете. Это забота программиста, который создавал приложение (подробнее об этом см. в "Руководстве программиста"). Пользователю остается выбрать нужный источник данных из списка доступных, чтобы использовать его в отчете. Это делается в меню "Данные|Выбрать данные для отчета...":



В этом окне перечислены все данные, зарегистрированные в отчете. Просто отметьте галочками те данные, которые вам нужны. Это можно сделать в любой момент работы с отчетом.

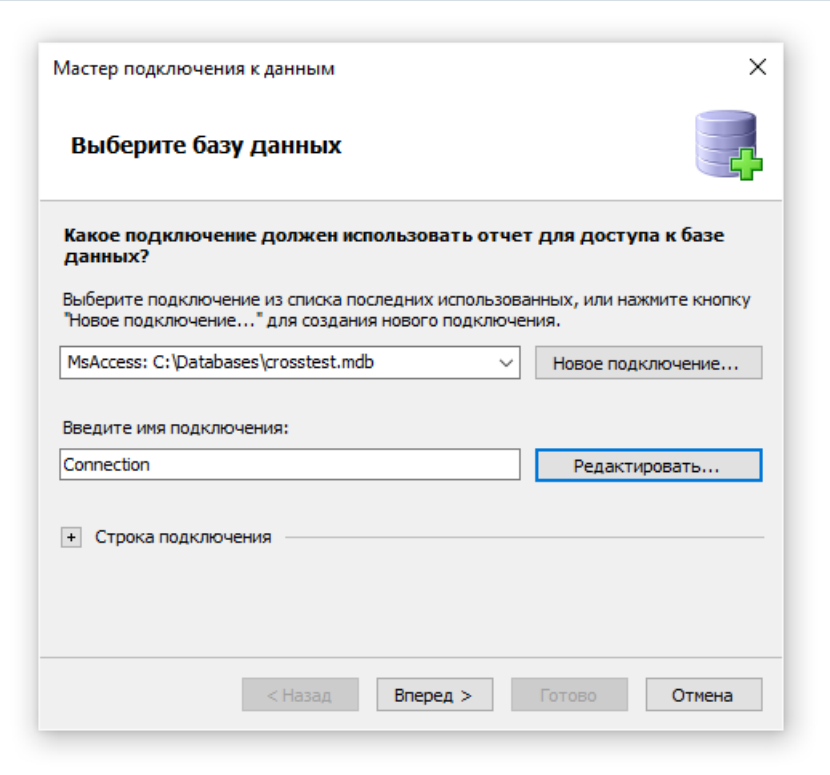
Второй способ – вы создаете новый источник данных самостоятельно. Это может быть таблица БД или запрос на языке SQL. В этом случае источник данных сохраняется в файле отчета.

FastReport позволяет подключаться к большинству популярных СУБД, таким как MS SQL, Oracle, Interbase, использовать файлы баз данных Access и файлы данных, хранящиеся в формате xml/xsd. Доступны расширения FastReport, позволяющие подключаться и к другим СУБД.

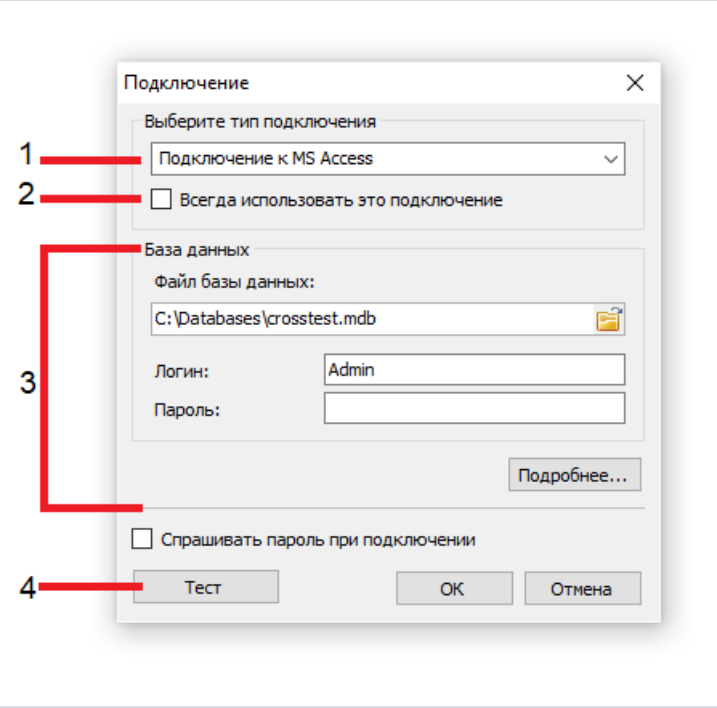
> Содержимое таблиц БД в файле отчета не сохраняется. Сохраняются только параметры подключения к БД и имена таблиц, используемых в отчете. Информация о подключении к БД может содержать такие данные, как логин и пароль, и поэтому сохраняется в файле отчета в зашифрованном виде. При необходимости степень безопасности можно увеличить, используя для шифрации данных свой собственный ключ. В этом случае файл отчета может быть корректно открыт только в вашей программе.

Создание источника данных

Чтобы создать новый источник данных, выберите пункт меню "Данные|Новый источник данных..." или в окне "Данные" нажмите кнопку "Действия" и выберите пункт "Новый источник данных...". Вы увидите окно "Мастер подключения к данным":



Первым делом вам предлагается создать подключение к базе данных. Для этого нажмите кнопку "Новое подключение...". Вы увидите окно с настройками подключения:

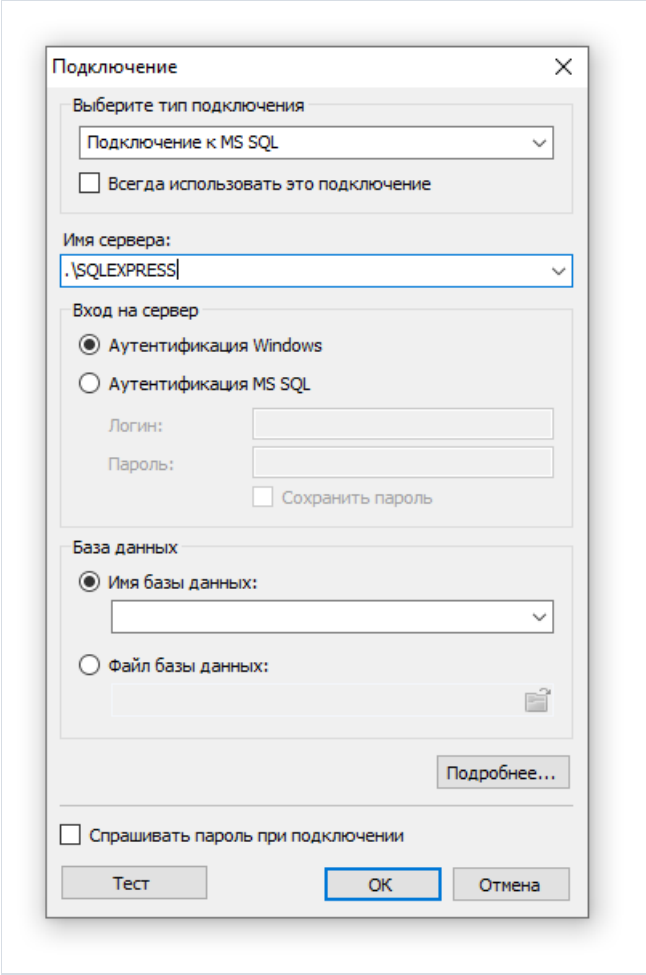


На рисунке обозначены следующие элементы управления:

1. тип подключения;

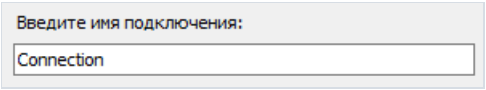
2. если включить эту галочку, в дальнейшем будет использоваться выбранный тип подключения по умолчанию;
3. настройки выбранного типа подключения;
4. кнопка проверки подключения.

На рисунке показано соединение с базой данных MS Access. Если выбрать другой тип подключения, область с настройками подключения (3) изменится. Например, подключение к базе данных MS SQL имеет следующие настройки:

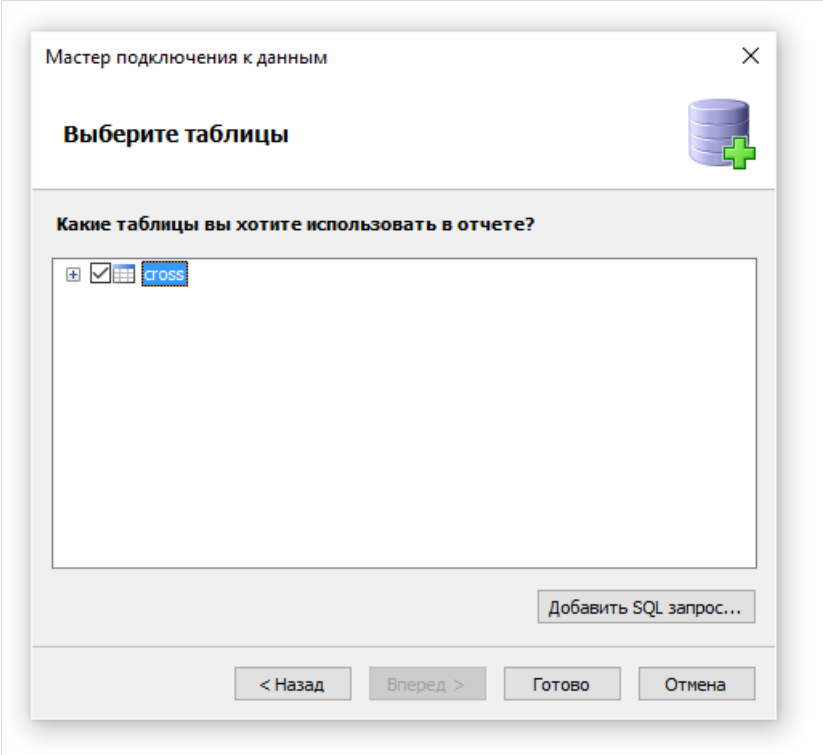


Вы должны выбрать необходимый тип подключения и настроить его параметры. После нажатия кнопки OK окно закроется, и вы вернетесь в окно мастера подключения.

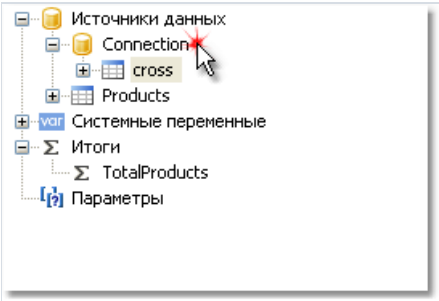
Для подключения надо указать имя. Это имя будет видно в окне "Данные". Можно оставить имя, которое предлагается по умолчанию:



Нажмите кнопку "Далее >" в окне мастера. На этом шаге предлагается выбрать таблицы, которые доступны в базе данных:



Отметьте галочками нужные таблицы и закройте мастер кнопкой "Готово". Теперь вы можете видеть в окне "Данные" созданное вами подключение, которое содержит выбранные источники данных:

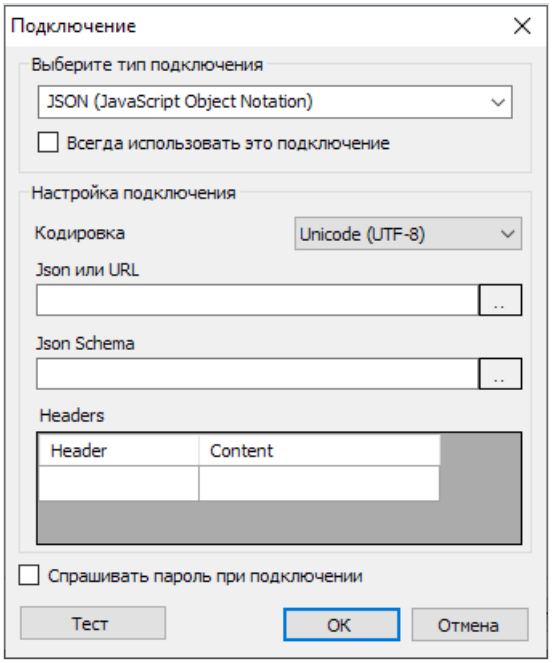


Подключение к JSON (JavaScript Object Notation)

Это подключение позволяет передать в качестве источника данных статичный JSON файл или адрес URL, по которому он будет получен перед построением отчёта.

Мастер подключения для JSON

При создании нового подключения в дизайнере FastReport оно имеет следующий вид:



Параметр	Описание
Кодировка	Задаёт кодировку, в которой будет обработан запрос к JSON, если указана ссылка на получение JSON.
JSON или URL	Задаёт статичный JSON или ссылку на получение JSON через API.
JSON Schema	Задаёт схему JSON.
Headers	Задаёт требуемые HTTP заголовки для подключения к JSON через API (необязательно).

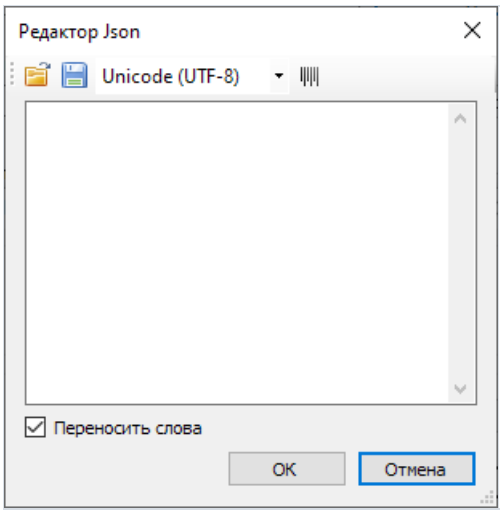
Справа от полей расположены кнопки для редактирования поля в отдельном окне.

Если значение JSON Schema пустое, то при нажатии кнопки OK или Test connection, будет автоматически построена схема по JSON.

Если изменить JSON или URL, когда схема уже указана, FastReport предложит обновить схему.

Окно редактирования JSON

Эта форма позволяет редактировать JSON, она имеет следующий вид:



Описание панели инструментов редактора слева направо:

1. Открыть файл — позволяет открыть JSON и вставить содержимое файла в редактор.
2. Сохранить файл — позволяет сохранить содержимое редактора в файл.
3. Кодировка — задаёт кодировку, в которой будет открыт JSON файл.
4. Форматирование — включает форматирование JSON, а также проводит валидацию JSON на соответствие спецификаций.

Далее идёт поле редактора, где можно изменить текст JSON.

Чекбокс "Переносить слова" позволит включить или отключить перенос слов в редакторе.

Как работает подключение

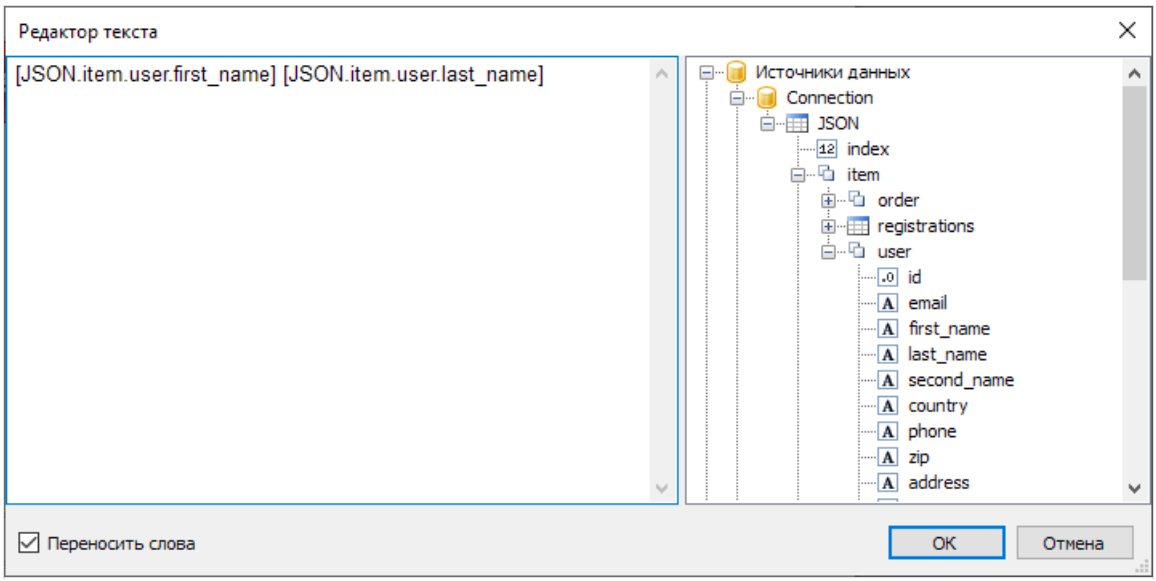
JSON не является таблицей, поэтому FastReport не воспринимает подключение к JSON, как таблицу данных.

Вместо этого FastReport воспринимает каждый массив из JSON, как иерархический источник данных с тремя полями:

Поле	Описание
index	Номер элемента.
item	Элемент.
array	Ссылка на массив элементов.

К DataBand можно подключить любой массив из JSON.

Далее можно использовать поля из JSON в объектах отчёта.


Редактор текста
[JSON.item.user.first_name] [JSON.item.user.last_name]
Источники данных
Connection
JSON
index
item
order
registrations
user
id
email
first_name
last_name
second_name
country
phone
zip
address
Переносить слова
OK
Отмена

Параметры в подключениях

При использовании параметров в подключении и передачи их с помощью мастера подключений параметры имеют строковый тип данных. Не все подключения корректно могут его конвертировать в тип, который сможет принять база данных. Поэтому для некоторых подключений были добавлены функции, конвертирующие строковое значение в другие типы данных, поддерживаемые библиотеками подключений. Ниже приведены в какие типы строка преобразуется исходя из типа параметра. Если необходимо значению параметра присвоить тип, не содержащийся в таблицах ниже, или нереализованный, то это можно сделать только программно.

PostgresDataConnection

NpgsqlDbType Type	C# Type	Пример/Формат
Bigint	long	`123456`
Money	decimal	`123456`
Numeric	decimal	`123456`
Integer	int	`123456`
Oid	uint	`123456`
Xid	uint	`123456`
Cid	uint	`123456`
Smallint	short	`123456`
InternalChar	byte	`123456`
Real	float	`123456.12`
Double	double	`123456.12`
Boolean	bool	`True` or '1'
Bit	string	`1`
Timestamp	DateTime	'12:15:12'
TimestampTZ	DateTime	'12:15:12'
Date	DateTime	'16/02/2008'
Time	TimeSpan	`6:12:14`
Interval	TimeSpan	`6:12:14`
TimeTZ	DateTimeOffset	`05/01/2008`
Uuid	Guid	`81a130d2-502f-4cf1-a376-63edeb000e9f`
Box	NpgsqlBox	`((x1,y1),(x2,y2))
Circle	NpgsqlCircle	<(x,y),r> (center point and radius)
Line	NpgsqlLine	{A,B,C}
Polygon	NpgsqlPolygon	((x1,y1),...)
Path	NpgsqlPath	((x1,y1),...)
LSeg	NpgsqlLSeg	((x1,y1),(x2,y2))
Point	NpgsqlPoint	(x,y)
Cidr	NpgsqlCidr	`192.168.100.128/25` IP address with mask
Inet	NpgsqlInet	`192.168.100.128/25` IP address
MacAddr	PhysicalAddress	'08:00:2b:01:02:03'
TsQuery	NpgsqlTsQuery	'fat & rat'
TsVector	NpgsqlTsVector	'a fat cat sat on a mat and ate a fat rat'
Char	string	`a` single char
Text	string	'string'
Varchar	string	'string'
Name	string	'string'
Citext	string	'string'
Bytea	string	`12AA` hex string
Varbit	string	`01101` bit string
Tid	NpgsqlTid	'12345, 123' uint number and ushort number
Array	---	Not implemented
Range	---	Not implemented
Hstore	---	Not implemented
Oidvector	---	Not implemented
MacAddr8	---	Not implemented
Int2Vector	---	Not implemented

MySqlDataConnection

MySqlDbType Type	C# Type	Пример
Int64	long	`123456`
UInt64	ulong	`123456`
Int32	int	`123456`
Int24	int	`123456`
UInt24	uint	`123456`
UInt32	uint	`123456`
Int16	short	`123456`
UInt16	ushort	`123456`
Decimal	decimal	`123456`
NewDecimal	decimal	`123456`
Byte	sbyte	`12345`
UByte	byte	`12345`
Year	byte	`1901`
Float	float	`123456.12`
Double	double	`123456.12`
Bit	bool	`True` or '1' bool
Bool	bool	`True` or '1' bool
DateTime	DateTime	'16/02/2008 12:15:12'
Date	DateTime	'16/02/2008'
Newdate	DateTime	'16/02/2008'
Time	DateTime	`6:12:14`
Timestamp	DateTime	'16/02/2008 12:15:12'
Timestamp	DateTime	'16/02/2008 12:15:12'
Guid	Guid	`81a130d2-502f-4cf1-a376-63edeb000e9f`
VarChar	string	'string'
String	string	'string'
TinyText	string	'string'
MediumText	string	'string'
LongText	string	'string'
Text	string	'string'
VarString	string	'string'
JSON	string	'string'
Enum	string	'string'
Binary	byte[]	`12AA` hex string
VarBinary	byte[]	`12AA` hex string
Blob	byte[]	`12AA` hex string
TinyBlob	byte[]	`12AA` hex string
MediumBlob	byte[]	`12AA` hex string
LongBlob	byte[]	`12AA` hex string
Null	DBNull	
Set	---	Not implemented
Geometry	---	Not implemented

ClickHouseDataConnection

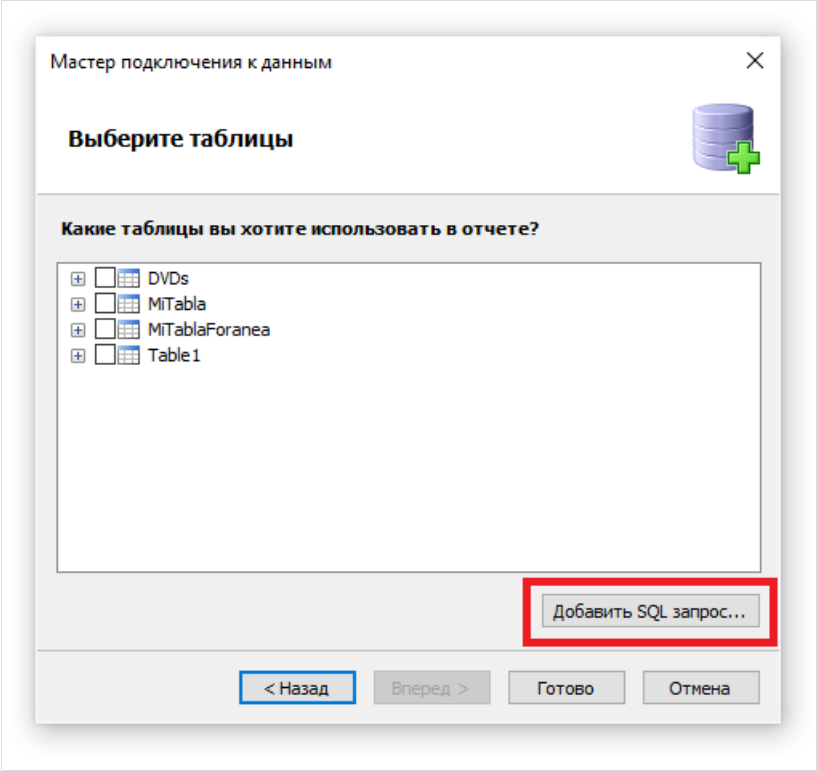
ClickHouse Type	C# Type	Пример
Enum8	sbyte	`123456`
Int8	sbyte	`123456`
Enum16	short	`123456`
Int16	short	`123456`
Int32	int	`123456`
Int64	long	`123456`
Decimal	decimal	`123456`
Float32	float	`123456.12`
Float64	double	`123456.12`
UInt8	byte	`123456`
UInt16	ushort	`123456`
UInt32	uint	`123456`
UInt64	ulong	`123456`
Date	DateTime	'16/02/2008 12:15:12'
DateTime	DateTimeOffset	`05/01/2008`
DateTime64	DateTimeOffset	`05/01/2008`
UUID	Guid	`81a130d2-502f-4cf1-a376-63edeb000e9f`
IPv6	IPAddress	'2001:0db8:85a3:08d3:1319:8a2e:0370:7344' IP address
IPv4	IPAddress	'127.0.0.1' IP address
String	string	'string'
FixedString	string	'string'
Nothing	DBNull	
Array	---	Not implemented
Nested	---	Not implemented
Tuple	---	Not implemented
Nullable	---	Not implemented
LowCardinality	---	Not implemented

MsSqlDataConnection

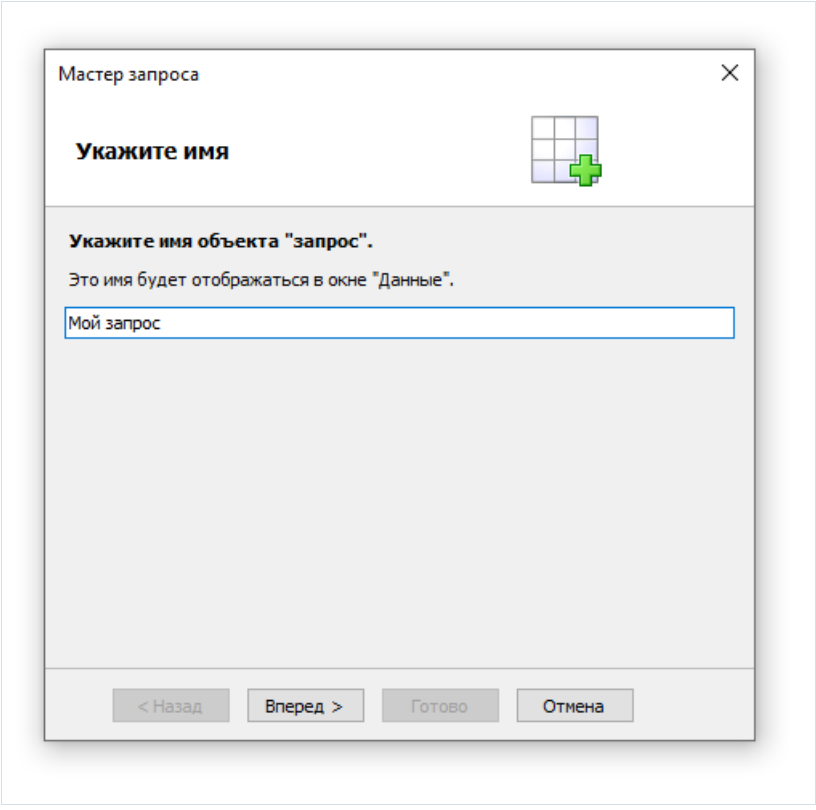
SqlDbType Type	C# Type	Пример
BigInt	long	`123456`
Bit	bool	`True` or '1'
DateTime	DateTime	'16/02/2008 12:15:12'
SmallDateTime	DateTime	'16/02/2008 12:15:12'
Date	DateTime	'16/02/2008 12:15:12'
Time	DateTime	'16/02/2008 12:15:12'
DateTime2	DateTime	'16/02/2008 12:15:12'
Char	string	`a` single char
NChar	string	'string'
NText	string	'string'
NVarChar	string	'string'
Text	string	'string'
VarChar	string	'string'
Xml	string	'<xml/>' xml string
Decimal	decimal	'string'
Money	decimal	'string'
SmallMoney	decimal	'string'
Real	float	`123456.12`
Float	float	`123456.12`
Int	int	`123456`
UniqueIdentifier	Guid	`81a130d2-502f-4cf1-a376-63edeb000e9f`
SmallInt	short	`123456`
TinyInt	byte	`123456`
DateTimeOffset	DateTimeOffset	`05/01/2008`
Variant	object	
Udt	object	
Binary	byte[]	`12AA` hex string
Image	byte[]	`12AA` hex string
Timestamp	byte[]	`12AA` hex string
VarBinary	byte[]	`12AA` hex string
Structured	---	Not implemented

Создание запроса на языке SQL

Мастер подключения к данным позволяет быстро выбрать таблицы, имеющиеся в базе данных. Создание запроса на языке SQL требует выполнения дополнительных действий. Для этого на втором шаге мастера нажмите кнопку "Добавить SQL запрос...":



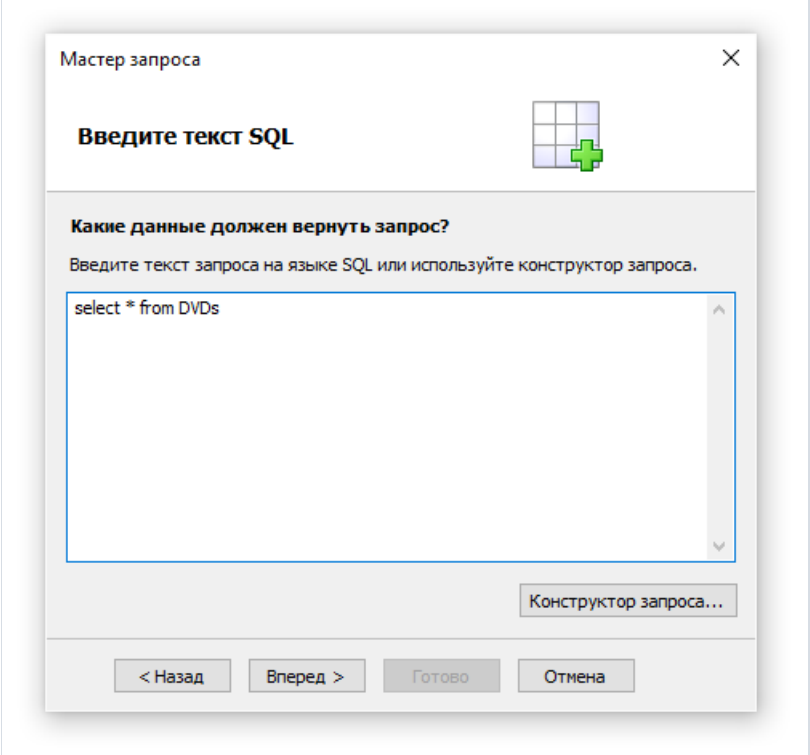
Вы увидите окно мастера запроса:



Мастер запроса содержит 4 страницы. Для переключения между страницами используйте кнопки "Далее" и

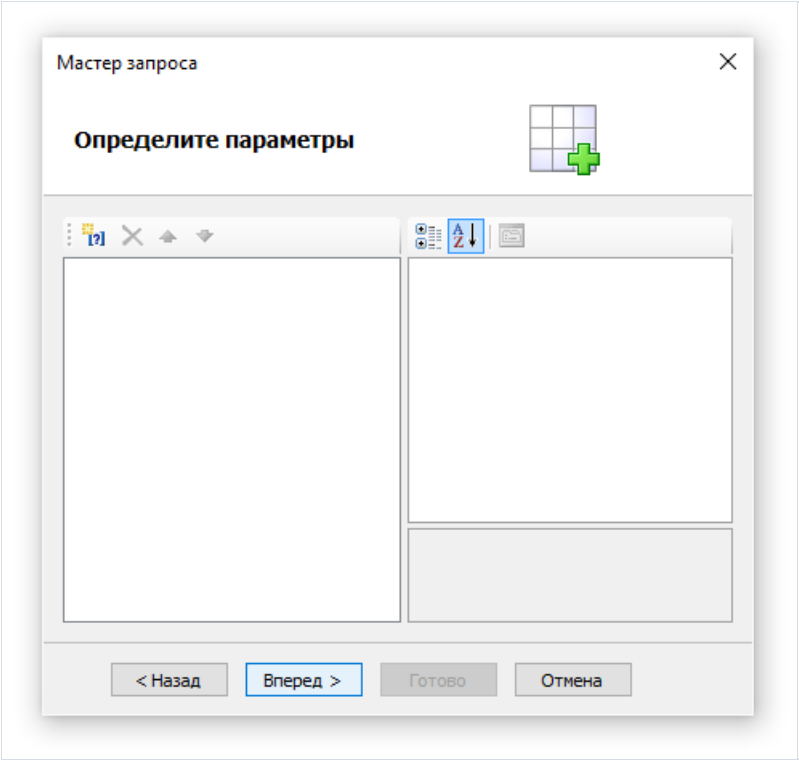
"Назад".

На первом шаге мастера вам предлагается дать имя источнику данных. Под этим именем он будет виден в окне "Данные". Введите любое имя; важно, чтобы имя объекта было уникальным. Нажмите кнопку "Далее". Вы увидите вторую страницу мастера:



На втором шаге мастера вам необходимо ввести текст запроса на языке SQL. Используйте тот диалект языка, который понимает ваша СУБД. Для построения текста запроса можно воспользоваться конструктором запроса, для этого нажмите кнопку "Конструктор запроса". Подробнее работа с построителем запроса будет рассмотрена далее.

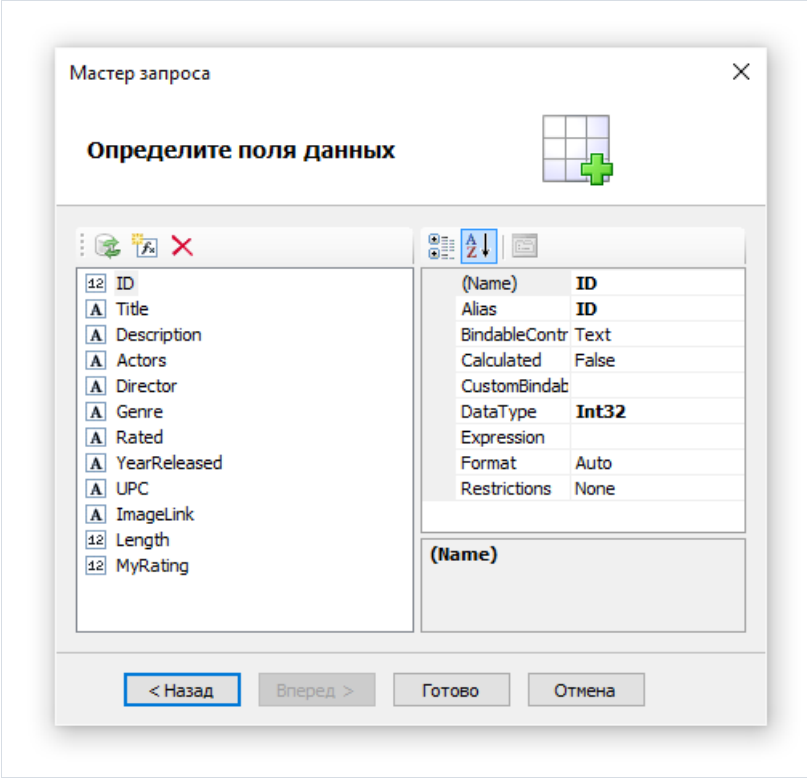
После того, как вы ввели текст запроса, нажмите кнопку "Далее". Вы увидите третью страницу мастера:



На этом шаге мастера вы можете определить параметры запроса. Это необходимо, если в тексте вашего

запроса содержатся параметры. Подробнее о настройке параметров будет рассказано далее.

Наконец, на последнем шаге запроса можно настроить поля, которые вернул запрос:



> Если вы допустили ошибку в тексте запроса или в описании параметров, вы увидите сообщение об ошибке в момент переключения на последнюю страницу мастера запроса.

Как правило, вам достаточно убедиться, что запрос вернул все необходимые поля. На этой странице мастера вы можете выполнить следующие действия:

- удалить ненужные поля из списка полей с помощью кнопки "Удалить";
- восстановить удаленные поля с помощью кнопки "Обновить";
- добавить вычисляемое поле в запрос с помощью кнопки "Добавить вычисляемое поле". Для нового поля надо настроить его свойства – имя (`Name`), тип данных (`DataType`) и выражение, которое будет возвращать значение при обращении к полю (`Expression`).

После закрытия мастера кнопкой "Готово" вы вернетесь в окно мастера подключения.

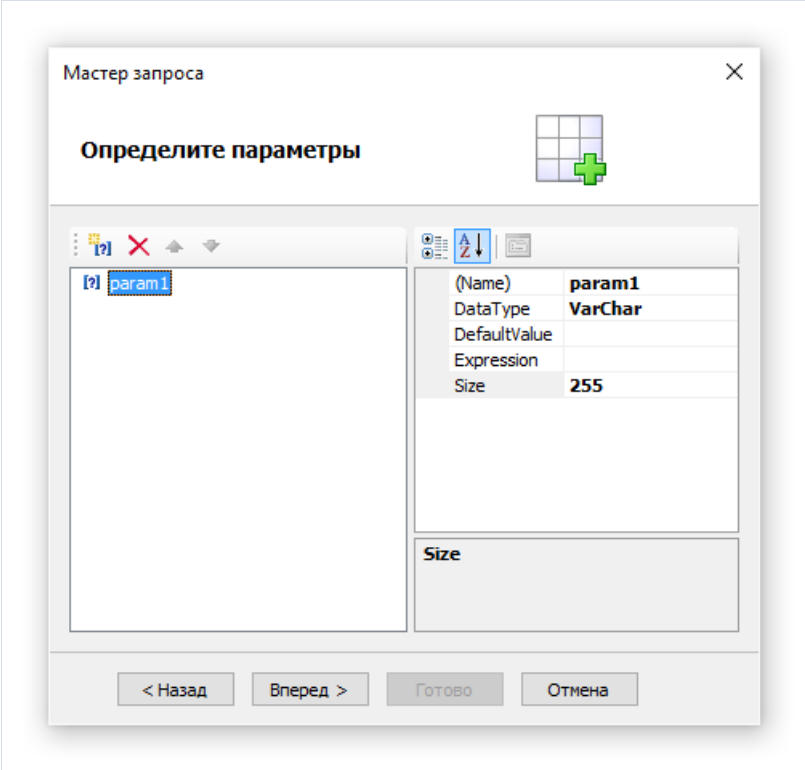
Параметры запроса

В тексте запроса могут присутствовать параметры. Рассмотрим следующий запрос:

```
select * from DVDs
where Title = @param1
```

Это запрос к демонстрационной базе MS SQL. В запросе определен параметр с именем `param1`. Здесь надо сделать важное замечание: способ определения параметров в запросе отличается для разных СУБД. Для MS SQL параметр обозначается символом `@`; для MS Access параметры не имеют имен и обозначаются символами `?`.

Для того чтобы работать с параметром, надо, как минимум, определить его значение. Это можно сделать на третьем шаге мастера "Мастер запроса", который мы рассмотрели выше. Для создания параметра нажмите кнопку "Добавить параметр". Будет создан новый параметр:



В окне справа необходимо настроить следующие свойства параметра:

Свойство	Описание
Name	Имя параметра. Здесь надо указать то же самое имя параметра, который вы использовали в тексте запроса. Некоторые СУБД (например, MS Access) не поддерживают именованные параметры. В этом случае оставьте это свойство без изменений.
DataType	Тип данных параметра.
DefaultValue	Значение по умолчанию, которое будет присвоено параметру при отсутствии значения в свойстве `Expression` или невозможности его вычислить (например, при операциях с запросом в режиме дизайна отчета).

Свойство	Описание
Expression	Выражение, которое возвращает значение параметра. Подробнее о выражениях смотрите в главе "Выражения". Это выражение будет вычислено при запуске отчета на выполнение.
Size	Размер данных параметра. Это свойство нужно указывать, если параметр имеет тип данных "строка".

Если вы неправильно настроите свойства параметра, вы получите ошибку в момент переключения на последнюю страницу мастера.

Передача значения в параметр запроса

Частый способ использования параметров – запрос у пользователя какой-либо информации, например, названия компании. Рассмотрим, как передать значение в параметр двумя способами.

Первый способ - вы передаете значение из программы. Так как нет простого и надежного способа передать значение непосредственно в параметр запроса, вам нужно использовать параметр отчета. В этом случае рекомендуется следующий порядок действий:

- Создайте параметр отчета (подробнее о параметрах смотрите в разделе "Параметры отчета"). Задайте для параметра отчета тот же тип данных, что используется в параметре запроса.
- В свойстве `Expression` параметра запроса укажите параметр отчета. Это можно сделать визуальным способом, вызвав редактор свойства `Expression`. Таким образом, свойство `Expression` будет содержать нечто вроде:

```
[ПараметрОтчета]
```

- Передайте значение параметра из программы. Для этого используйте метод `SetParameterValue` объекта Report:

```
report1.SetParameterValue("ПараметрОтчета", 10);
```

Второй способ - вы используете диалоговую форму отчета (подробнее о диалогах смотрите в главе "Диалоговые формы"). На диалоге можно разместить элементы управления, в которые пользователь будет вводить данные. Например, для запроса названия компании можно использовать элемент управления TextBoxControl. В этом случае в свойстве `Expression` можно указать следующее:

```
TextBox1.Text
```

где TextBox1 – элемент управления на диалоговой форме отчета, в котором содержится введенное пользователем значение.

Полная настройка параметра для приведенного выше примера может выглядеть так:

```
Name=param1
DataType=VarChar
DefaultValue=   (пустая строка)
Expression=TextBox1.Text
Size=255
```

Редактирование подключения

Подключение к данным, которое было создано при работе мастера подключения к данным, можно редактировать. Для этого выберите подключение в окне "Данные" и нажмите кнопку "Редактировать" на панели инструментов. Вы увидите уже знакомое окно мастера подключения. Вы можете нажать кнопку "Редактировать..." для изменения настроек подключения. Поменять тип подключения нельзя: он задается только в момент его создания.

На второй странице мастера вы можете выбрать таблицы, которые вы хотите видеть в окне "Данные". Когда вы закончили редактирование, нажмите кнопку "Готово".

Редактирование источника данных

Источник данных, который был создан с помощью мастера "Мастер подключения к данным", можно редактировать. Для этого выберите источник данных в окне "Данные" и нажмите кнопку "Редактировать" на панели инструментов. Вы увидите окно мастера запроса, которое мы уже рассматривали ранее. В этом окне можно поменять текст запроса SQL, настроить параметры запроса и поля данных.

Для удаления источника данных выберите его и нажмите кнопку "Удалить" на панели инструментов. При этом физического удаления источника не происходит, он помечается как недоступный. Вы можете зайти в меню "Данные|Выбрать данные для отчета..." и включить такой источник данных. Однако это не должно вас смущать: удаленные источники данных не сохраняются в файле отчета, и, соответственно, не восстанавливаются при следующем чтении отчета.

Добавление источника данных в существующее подключение

Добавить источник данных (таблицу или запрос) в имеющееся подключение можно двумя способами:

- вызвать редактор подключения, как описано в разделе "Редактирование подключения". Будет показано окно мастера подключения, на второй странице которого можно выбрать или создать новый источник данных;
- щелкнуть правой кнопкой мыши на элементе "Подключение" и выбрать пункт меню "Новый источник данных...". Как и в предыдущем способе, это приведет к появлению окна мастера подключения.

Псевдонимы (aliases)

Каждый элемент данных (источник данных или его поле) имеет свое имя. По умолчанию это то имя, которое определено в базе данных. В некоторых случаях бывает трудно понять, что скрывается за именем поля, например, `ProdID`.

У элементов данных есть и вторые имена – псевдонимы. Используя псевдонимы, вы можете переименовать элемент. Например, если у нас есть источник данных Categories с полем `ProdID`, можно задать следующие псевдонимы:

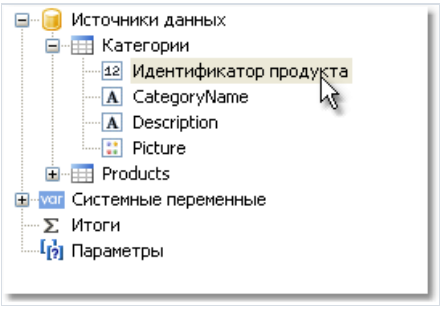
```
Categories --> Категории
ProdID --> Идентификатор продукта
```

К такому полю можно обратиться следующим образом:

```
[Категории.Идентификатор продукта]
```

> При обращении к элементу данных вы должны использовать псевдоним, если он определен. Обращаться к элементу, используя его оригинальное имя, в этом случае нельзя.

Для того чтобы переименовать элемент данных, выберите его в окне "Данные" и нажмите клавишу F2. Также можно выбрать пункт "Переименовать" в контекстном меню объекта. После этого наберите желаемое имя и нажмите `Enter`:



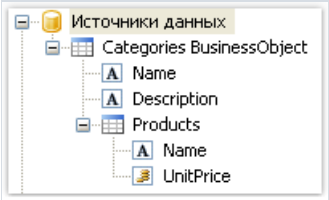
Вы также можете переименовать элемент, используя окно "Свойства". Выберите элемент в окне "Данные", а в окне "Свойства" поменяйте значение свойства `Alias`.

Чтобы удалить псевдоним (вернуть оригинальное имя), выберите элемент и в его контекстном меню выберите пункт "Удалить псевдоним".

Иерархические источники данных

Рассмотренные нами источники данных были реляционными, они представляли собой аналог таблицы реляционной СУБД. FastReport поддерживает и другой тип источников – иерархические. Такие источники создаются при регистрации в отчете данных из бизнес-объектов. Поскольку такие данные могут быть добавлены в отчет исключительно программным способом, рассмотрим работу с ними с точки зрения пользователя.

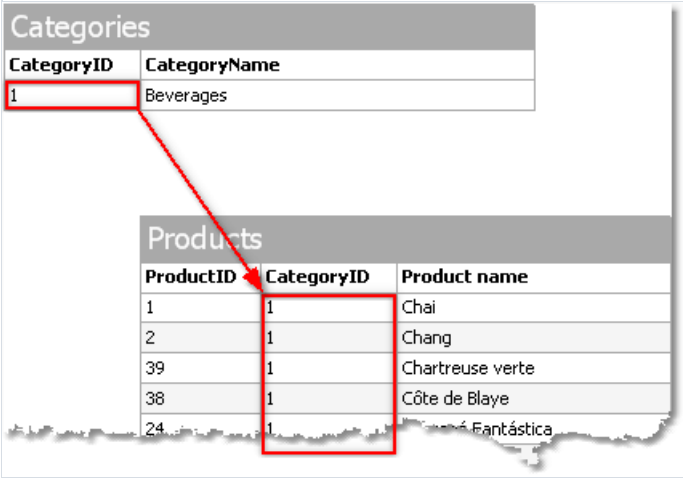
Иерархические источники данных отличаются от обычных источников только тем, что для них не нужно создавать связь. Источники уже связаны отношением "родительский-дочерний". В окне "Данные" такие источники выглядят следующим образом:



Как видно, родительский источник данных "Categories Business Object" содержит в себе дочерний источник "Products". Вы можете работать с иерархическими источниками так же, как и с остальными.

Связи (relations)

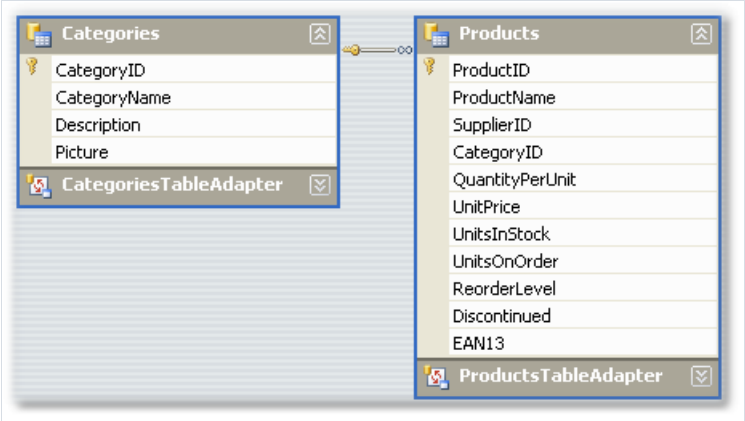
Между двумя источниками данных можно установить связь. Связь описывает отношение типа "главный-подчиненный" (или "родительский-дочерний"). Например, одна запись в таблице Categories может иметь множество подчиненных записей в таблице Products:



Для того чтобы создать связь, нужно указать следующее:

1. главная таблица;
2. подчиненная таблица;
3. набор ключевых полей в главной таблице;
4. набор ключевых полей в подчиненной таблице.

Рассмотрим в качестве примера таблицы Categories и Products из демонстрационной базы данных. Они имеют следующую структуру:

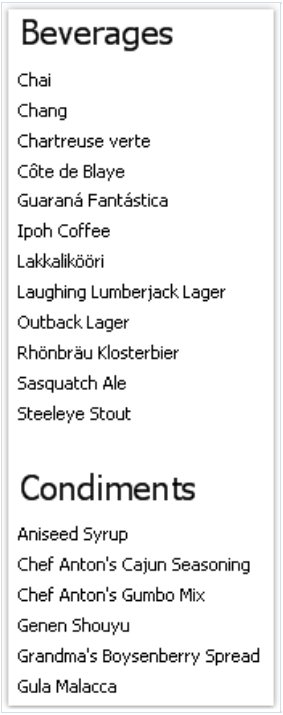


Обе таблицы имеют поле `CategoryID`, по которому можно установить связь. При этом одной категории будут соответствовать несколько продуктов.

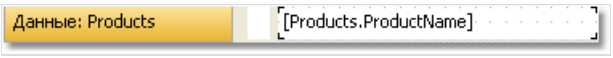
Как можно использовать связанные источники данных в FastReport? Есть два способа сделать это.

Первый способ дает возможность строить отчеты типа "главный-подчиненный" (master-detail). Для этого используются два бэнда "Данные". Главный бэнд привязывается к главному источнику данных, подчиненный бэнд – к подчиненному источнику. Наш пример может выглядеть так:

Такой отчет, если его запустить, напечатает список продуктов в каждой категории:



Второй способ позволяет из подчиненного источника обращаться к главному. Покажем это на примере. Допустим, мы хотим распечатать список всех продуктов. Для этого нам нужен один бэнд "Данные", который подключен к таблице Products:



Такой отчет, если его запустить, напечатает все продукты из всех категорий. Допустим, мы рядом с каждым продуктом хотим напечатать имя категории, к которой он относится. Без использования связи это было бы затруднительно, т.к. все, что мы знаем о категории – это ее идентификатор (поле `CategoryID` в таблице Products). Имя категории хранится в поле `CategoryName` таблицы Categories. С помощью связи мы можем обратиться к имени категории следующим образом:

```
[Products.Categories.CategoryName]
```

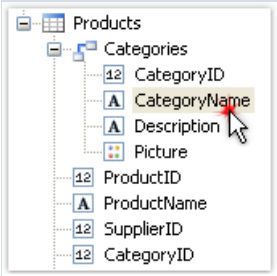
При этом FastReport выполнит следующие действия:

- возьмет текущую строку в таблице Products;
- в таблице Categories найдет строку, которая является родительской;
- для найденной строки вернет значение поля `CategoryName`.

В общем случае форма обращения к полю родительской таблицы допускает неограниченное количество предков таблицы:

```
[дочерняя_таблица.ее_родитель.родитель_родителя.и_так_далее.поле]
```

Чтобы добавить в отчет такое поле данных, откройте таблицу Products в окне "Данные". Вы увидите, что среди ее полей есть ссылка на родительскую таблицу Categories:



Если перетащить показанное на рисунке поле в отчет, мы получим объект "Текст" с текстом:

```
[Products.Categories.CategoryName]
```

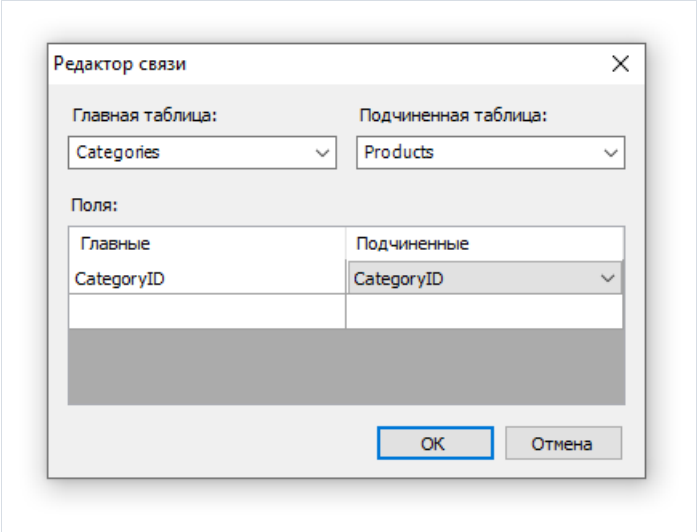
Наш отчет будет выглядеть следующим образом:

Если его запустить, мы увидим следующее:

Alice Mutton	Meat/Poultry
Aniseed Syrup	Condiments
Boston Crab Meat	Seafood
Camembert Pierrot	Dairy Products
Carnarvon Tigers	Seafood
Chai	Beverages
Chang	Beverages

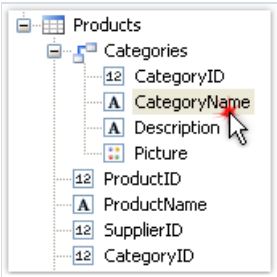
Создание связи

Для того чтобы создать связь, в окне "Данные" нажмите кнопку "Действия" и выберите пункт "Новая связь...". Вы увидите редактор связи:



В первую очередь вам нужно выбрать главную и подчиненную таблицы. После этого в нижней части окна надо выбрать пары полей – для каждой таблицы соответственно. Таблицы могут быть связаны с помощью одного или нескольких полей. После того, как поля настроены, закройте редактор связи кнопкой OK.

Созданную связь можно увидеть в окне "Данные", если выбрать подчиненный источник данных и раскрыть список его полей. Среди полей мы увидим связь с родительским источником:



Поля родительского источника можно перетаскивать на лист отчета методом drag&drop. Так, если выбрать показанное на рисунке поле и перетащить его на лист отчета, мы получим объект "Текст" со следующим содержимым:

```
[Products.Categories.CategoryName]
```

Редактирование связи

Для того чтобы отредактировать связь, откройте список полей дочернего источника данных, найдите нужную связь и нажмите кнопку "Редактировать..." на панели инструментов. При этом будет вызван редактор связи, который мы рассматривали выше.

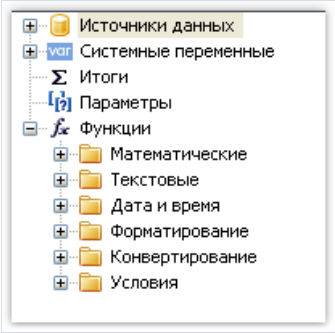
Системные переменные

В FastReport доступно несколько переменных, которые можно использовать в любом месте отчета. Ниже приведен полный список переменных:

Переменная	Описание
Date	Дата и время старта отчета.
Page	Номер текущей страницы.
TotalPages	Общее количество страниц в отчете. Чтобы использовать эту переменную, надо включить двойной проход у отчета. Это можно сделать в меню "Отчет|Свойства...".
PageN	Номер страницы в виде: "Страница N".
PageNofM	Номер страницы в виде: "Страница N из M".
Row#	Номер строки данных внутри группы. Это значение сбрасывается при старте новой группы.
AbsRow#	Абсолютный номер строки данных. Это значение не сбрасывается при старте новой группы.
Page#	Номер текущей страницы. Если вы объединяете несколько готовых отчетов в пакет, эта переменная вернет номер страницы в пакете. Эта переменная является макросом, т.е. ее значение подставляется в момент отображения компонента в окне просмотра. Использовать ее в выражениях нельзя.
TotalPages#	Общее количество страниц в отчете. Если вы объединяете несколько готовых отчетов в пакет, эта переменная вернет количество страниц в пакете. Для работы этой переменной двойной проход отчета не требуется. Эта переменная является макросом, т.е. ее значение подставляется в момент отображения компонента в окне просмотра. Использовать ее в выражениях нельзя.
HierarchyLevel	Текущий уровень иерархии в иерархическом отчете (см. "Печать иерархии"). Верхний уровень имеет значение 1.
HierarchyRow#	Полный номер строки в иерархическом отчете, имеет вид "1.2.1".

Функции

FastReport .NET предлагает пользователю большое количество встроенных функций (более 60), которые можно использовать в отчете. Функции разбиты на несколько категорий, и доступны в окне "Данные":



Функции можно использовать в любых выражениях, в коде скрипта, а также выводить их значение в объектах "Текст". Например, объект "Текст" содержит вызов функции:

```
[Sqrt(4)]
```

В результате будет напечатано значение 2 (корень квадратный из 4).

Следующее выражение вернет значение 4:

```
Sqrt(4) + 2
```

Перечислим способы визуальной вставки функций в отчет:

- функцию можно перетащить на лист отчета, при этом будет создан объект "Текст", содержащий вызов функции. Текст необходимо отредактировать, добавив в него параметры функции;
- функцию можно перетащить в код скрипта;
- в окне редактора выражения также присутствует копия окна "Данные", и так же можно перетаскивать функцию в текст выражения.

Ниже будет дано описание каждой функции.

Математические

Abs

Функция	Параметры	Возвращаемое значение
`Abs`	`sbyte value`	`sbyte`
`Abs`	`short value`	`short`
`Abs`	`int value`	`int`
`Abs`	`long value`	`long`
`Abs`	`float value`	`float`
`Abs`	`double value`	`double`
`Abs`	`decimal value`	`decimal`

Возвращает абсолютную величину значения `value`.

Пример:

```
Abs(-2.2) = 2.2
```

Acos

Функция	Параметры	Возвращаемое значение
`Acos`	`double d`	`double`

Возвращает угол (в радианах), который соответствует косинусу `d`. Значение `d` должно быть в диапазоне от -1 до 1.

> Чтобы перевести радианы в градусы, умножьте значение в радианах на `180 / Math.PI`.

Пример:

```
Acos(0) * 180 / Math.PI = 90
```

Asin

Функция	Параметры	Возвращаемое значение
`Asin`	`double d`	`double`

Возвращает угол (в радианах), который соответствует синусу `d`. Значение `d` должно быть в диапазоне от -1 до 1.

> Чтобы перевести радианы в градусы, умножьте значение в радианах на `180 / Math.PI`.

Пример:

```
Asin(0) = 0
```

Atan

Функция	Параметры	Возвращаемое значение
`Atan`	`double d`	`double`

Возвращает угол (в радианах), который соответствует тангенсу `d`.

> Чтобы перевести радианы в градусы, умножьте значение в радианах на `180 / Math.PI`.

Пример:

```
Atan(1) * 180 / Math.PI = 45
```

Ceiling

Функция	Параметры	Возвращаемое значение
`Ceiling`	`double d`	`double`
`Ceiling`	`decimal d`	`decimal`

Возвращает наименьшее целое значение, большее и равное `d`.

Пример:

```
Ceiling(1.7) = 2
```

Cos

Функция	Параметры	Возвращаемое значение
`Cos`	`double d`	`double`

Возвращает косинус значения `d`. Значение `d` должно быть в радианах.

> Чтобы перевести градусы в радианы, умножьте значение в градусах на `Math.PI / 180`.

Пример:

```
Cos(90 * Math.PI / 180) = 0
```

Exp

Функция	Параметры	Возвращаемое значение
`Exp`	`double d`	`double`

Возвращает число e (2.71828), возведенное в степень `d`.

Пример:

```
Exp(1) = 2.71828
```

Floor

Функция	Параметры	Возвращаемое значение
`Floor`	`double d`	`double`
`Floor`	`decimal d`	`decimal`

Возвращает наибольшее целое значение, меньшее и равное `d`.

Пример:

```
Floor(1.7) = 1
```

Log

Функция	Параметры	Возвращаемое значение
`Log`	`double d`	`double`

Возвращает натуральный логарифм значения `d`.

Пример:

```
Log(2.71828) = 1
```

Maximum

Функция	Параметры	Возвращаемое значение
`Maximum`	`int val1, int val2`	`int`
`Maximum`	`long val1, long val2`	`long`
`Maximum`	`float val1, float val2`	`float`
`Maximum`	`double val1, double val2`	`double`
`Maximum`	`decimal val1, decimal val2`	`decimal`

Возвращает максимальное из значений `val1`, `val2`.

Пример:

```
Maximum(1,2) = 2
```

Minimum

Функция	Параметры	Возвращаемое значение
`Minimum`	`int val1, int val2`	`int`
`Minimum`	`long val1, long val2`	`long`
`Minimum`	`float val1, float val2`	`float`
`Minimum`	`double val1, double val2`	`double`
`Minimum`	`decimal val1, decimal val2`	`decimal`

Возвращает минимальное из значений `val1`, `val2`.

Пример:

```
Minimum(1,2) = 1
```

Round

Функция	Параметры	Возвращаемое значение
`Round`	`double d`	`double`
`Round`	`decimal d`	`decimal`

Округляет значение `d` до ближайшего целого.

Пример:

```
Round(1.47) = 1
```

Функция	Параметры	Возвращаемое значение
`Round`	`double d, int digits`	`double`
`Round`	`decimal d, int digits`	`decimal`

Округляет значение `d` до числа разрядов, указанного в параметре `digits`.

Пример:

```
Round(1.478, 2) = 1.48
```

Sin

Функция	Параметры	Возвращаемое значение
`Sin`	`double d`	`double`

Возвращает синус значения `d`. Значение `d` должно быть указано в радианах.

> Чтобы перевести градусы в радианы, умножьте значение в градусах на `Math.PI / 180`.

Пример:

```
Sin(90 * Math.PI / 180) = 1
```

Sqrt

Функция	Параметры	Возвращаемое значение
`Sqrt`	`double d`	`double`

Возвращает квадратный корень от значения `d`.

Пример:

```
Sqrt(4) = 2
```

Tan

Функция	Параметры	Возвращаемое значение
`Tan`	`double d`	`double`

Возвращает тангенс значения `d`. Значение `d` должно быть указано в радианах.

> Чтобы перевести градусы в радианы, умножьте значение в градусах на `Math.PI / 180`.

Пример:

```
Tan(45 * Math.PI / 180) = 1
```

Truncate

Функция	Параметры	Возвращаемое значение
`Truncate`	`double d`	`double`
`Truncate`	`decimal d`	`decimal`

Возвращает целую часть значения `d`.

Пример:

```
Truncate(1.7) = 1
```

Текстовые

Замечание:

- эти функции не модифицируют переданное в них текстовое значение. Вместо этого они возвращают измененную строку;
- позиция первого символа в строке равна 0. Это следует учитывать при передаче параметров в некоторые функции, например, `Insert`.

Asc

Функция	Параметры	Возвращаемое значение
`Asc`	`char c`	`int`

Возвращает целое значение, представляющее код символа `c`.

Пример:

```
Asc('A') = 65
```

Chr

Функция	Параметры	Возвращаемое значение
`Chr`	`int i`	`char`

Возвращает символ с кодом `i`.

Пример:

```
Chr(65) = 'A'
```

Insert

Функция	Параметры	Возвращаемое значение
`Insert`	`string s, int startIndex, string value`	`string`

Вставляет подстроку `value` в строку `s`, начиная с позиции `startIndex`, и возвращает новую строку.

Пример:

```
Insert("ABC", 1, "12") = "A12BC"
```

Length

Функция	Параметры	Возвращаемое значение
`Length`	`string s`	`int`

Возвращает длину строки `s`.

Пример:

```
Length("ABC") = 3
```

LowerCase

Функция	Параметры	Возвращаемое значение
`LowerCase`	`string s`	`string`

Переводит все символы строки `s` в нижний регистр и возвращает результат.

Пример:

```
LowerCase("ABC") = "abc"
```

PadLeft

Функция	Параметры	Возвращаемое значение
`PadLeft`	`string s, int totalWidth`	`string`

Дополняет строку `s` пробелами слева, до длины, указанной в параметре `totalWidth`, и возвращает новую строку.

Пример:

```
PadLeft("ABC", 5) = "  ABC"
```

Функция	Параметры	Возвращаемое значение
`PadLeft`	`string s, int totalWidth, char paddingChar`	`string`

Дополняет строку `s` символами `paddingChar` слева, до длины, указанной в параметре `totalWidth`, и возвращает новую строку.

Пример:

```
PadLeft("ABC", 5, '0') = "00ABC"
```

PadRight

Функция	Параметры	Возвращаемое значение
`PadRight`	`string s, int totalWidth`	`string`

Дополняет строку `s` пробелами справа, до длины, указанной в параметре `totalWidth`, и возвращает новую строку.

Пример:

```
PadRight("ABC", 5) = "ABC  "
```

Функция	Параметры	Возвращаемое значение
`PadRight`	`string s, int totalWidth, char paddingChar`	`string`

Дополняет строку `s` символами `paddingChar` справа, до длины, указанной в параметре `totalWidth`, и возвращает новую строку.

Пример:

```
PadRight("ABC", 5, '0') = "ABC00"
```

Remove

Функция	Параметры	Возвращаемое значение
`Remove`	`string s, int startIndex`	`string`

Удаляет все символы из строки `s`, начиная с позиции `startIndex`, и возвращает новую строку.

Пример:

```
Remove("ABCD", 3) = "ABC"
```

Функция	Параметры	Возвращаемое значение
`Remove`	`string s, int startIndex, int count`	`string`

Удаляет указанное в `count` количество символов из строки `s`, начиная с позиции `startIndex`, и возвращает новую строку.

Пример:

```
Remove("A00BC", 1, 2) = "ABC"
```

Replace

Функция	Параметры	Возвращаемое значение
`Replace`	`string s, string oldValue, string newValue`	`string`

Заменяет в строке `s` все вхождения подстроки `oldValue` на подстроку `newValue`, и возвращает новую строку.

Пример:

```
Replace("A00", "00", "BC") = "ABC"
```

Substring

Функция	Параметры	Возвращаемое значение
`Substring`	`string s, int startIndex`	`string`

Возвращает все символы строки `s`, начиная с позиции `startIndex` и до конца строки.

Пример:

```
Substring("ABCDEF", 4) = "EF"
```

Функция	Параметры	Возвращаемое значение
`Substring`	`string s, int startIndex, int length`	`string`

Возвращает количество `length` символов строки `s`, начиная с позиции `startIndex`.

Пример:

```
Substring("ABCDEF", 1, 3) = "BCD"
```

TitleCase

Функция	Параметры	Возвращаемое значение
`TitleCase`	`string s`	`string`

Переводит символы строки `s` в "титульный" регистр. При этом первый символ каждого слова делается заглавным, остальные символы - строчными.

Пример:

```
TitleCase("john smith") = "John Smith"
```

Trim

Функция	Параметры	Возвращаемое значение
`Trim`	`string s`	`string`

Обрезает незначащие пробелы в начале и конце строки `s`.

Пример:

```
Trim("  ABC  ") = "ABC"
```

UpperCase

Функция	Параметры	Возвращаемое значение
`UpperCase`	`string s`	`string`

Переводит все символы строки `s` в верхний регистр и возвращает результат.

Пример:

```
UpperCase("abc") = "ABC"
```

Дата и время

Для установки конкретной даты можно использовать конструкторы структуры `DateTime` либо функции `ToDateTime` (подробнее в разделе "Конвертирование") или `DateSerial` (рассматривается ниже).

Например, чтобы создать дату и время через конструкторы, можно использовать следующий код:

```
new DateTime(год, месяц, день, часы, минуты, секунды);
```

При использовании функций для преобразования даты из данного раздела по умолчанию сохраняется и информация о времени. Для удаления времени из даты можно воспользоваться функциями `Format` или `FormatDateTime`. Более подробная информация об этих функциях представлена в разделе "Форматирование".

В качестве примера использования поля из базы данных в функциях будет использоваться поле `[Employees.BirthDate]` из демонстрационной базы данных.

```
[Employees.BirthDate] = 27.01.1986
```

AddDays

Функция	Параметры	Возвращаемое значение
`AddDays`	`DateTime date, double value`	`DateTime`

Добавляет к дате `date` количество дней `value` и возвращает новую дату.

Пример:

```
AddDays(new DateTime(2024,4,2), 2) = 04.04.2024 0:00:00
AddDays(ToDateTime("2.4.2024"), 2) = 04.04.2024 0:00:00
AddDays([Employees.BirthDate], 2) = 29.01.1986 0:00:00
```

AddHours

Функция	Параметры	Возвращаемое значение
`AddHours`	`DateTime date, double value`	`DateTime`

Добавляет к дате `date` количество часов `value` и возвращает новую дату.

Пример:

```
AddHours(new DateTime(2024,4,2,5,30,5), 1) = 02.04.2024 6:30:05
AddHours(ToDateTime("2.4.2024 5:30:05"), 1) = 02.04.2024 6:30:05
AddHours([Employees.BirthDate], 1) = 27.01.1986 1:00:00
```

AddMinutes

Функция	Параметры	Возвращаемое значение
`AddMinutes`	`DateTime date, double value`	`DateTime`

Добавляет к дате `date` количество минут `value` и возвращает новую дату.

Пример:

```
AddMinutes(new DateTime(2024,4,2,5,30,5), 10) = 02.04.2024 5:40:05
AddMinutes(ToDateTime("2.4.2024 5:30:05"), 10) = 02.04.2024 5:40:05
AddMinutes([Employees.BirthDate], 10) = 27.01.1986 0:10:00
```

AddMonths

Функция	Параметры	Возвращаемое значение
`AddMonths`	`DateTime date, int value`	`DateTime`

Добавляет к дате `date` количество месяцев `value` и возвращает новую дату.

Пример:

```
AddMonths(new DateTime(2024,4,2,5,30,5), 2) = 02.06.2024 5:30:05
AddMonths(ToDateTime("2.4.2024 5:30:05"), 2) = 02.06.2024 5:30:05
AddMonths([Employees.BirthDate], 2) = 27.03.1986 0:00:00
```

AddSeconds

Функция	Параметры	Возвращаемое значение
`AddSeconds`	`DateTime date, double value`	`DateTime`

Добавляет к дате `date` количество секунд `value` и возвращает новую дату.

Пример:

```
AddSeconds(new DateTime(2024,4,2,5,30,5), 10) = 02.04.2024 5:30:15
AddSeconds(ToDateTime("2.4.2024 5:30:05"), 10) = 02.04.2024 5:30:15
AddSeconds([Employees.BirthDate], 10) = 27.01.1986 0:00:10
```

AddYears

Функция	Параметры	Возвращаемое значение
`AddYears`	`DateTime date, int value`	`DateTime`

Добавляет к дате `date` количество лет `value` и возвращает новую дату.

Пример:

```
AddYears(new DateTime(2024,4,2,5,30,5), 3) = 02.04.2027 5:30:05
AddYears(ToDateTime("2.4.2024 5:30:05"), 3) = 02.04.2027 5:30:05
AddYears([Employees.BirthDate], 3) = 27.01.1989 0:00:00
```

DateDiff

Функция	Параметры	Возвращаемое значение
`DateDiff`	`DateTime date1, DateTime date2`	`TimeSpan`

Возвращает интервал - количество дней, часов, минут, секунд между двумя датами.

Пример:

```
DateDiff(new DateTime(2024,4,2,5,0,0), new DateTime(2025,1,2,5,30,5)) = -275.00:30:05
DateDiff(ToDateTime("2.1.2025 5:30:05"), ToDateTime("2.4.2024 5:00:00")) = 275.00:30:05
DateDiff(ToDateTime("2.4.2024 5:00:00"), [Employees.BirthDate]) = 13945.05:00:00
```

DateSerial

Функция	Параметры	Возвращаемое значение
`DateSerial`	`int year, int month, int day`	`DateTime`

Создает новое значение `DateTime` из указанных года (`year`), месяца (`month`) и дня (`day`).

Еще один доступный способ для установки конкретной даты.

Пример:

```
DateSerial(2024,4,2) = 02.04.2024 0:00:00
```

Day

Функция	Параметры	Возвращаемое значение
`Day`	`DateTime date`	`int`

Извлекает день месяца (1-31) из указанной даты.

Пример:

```
Day(new DateTime(2024,4,2)) = 2
Day(ToDateTime("2.4.2024")) = 2
Day([Employees.BirthDate]) = 27
```

DayOfWeek

Функция	Параметры	Возвращаемое значение
`DayOfWeek`	`DateTime date`	`string`

Возвращает название дня недели (понедельник..воскресенье) указанной даты.

Пример:

```
DayOfWeek(new DateTime(2024,4,2)) = "вторник"
DayOfWeek(ToDateTime("2.4.2024")) = "вторник"
DayOfWeek([Employees.BirthDate]) = "понедельник"
```

DayOfYear

Функция	Параметры	Возвращаемое значение
`DayOfYear`	`DateTime date`	`int`

Возвращает порядковый номер дня в году (1-365) в указанной дате.

Пример:

```
DayOfYear(new DateTime(2024,4,2)) = 93
DayOfYear(ToDateTime("2.4.2024")) = 93
DayOfYear([Employees.BirthDate]) = 27
```

DaysInMonth

Функция	Параметры	Возвращаемое значение
`DaysInMonth`	`int year, int month`	`int`

Возвращает количество дней в месяце `month` указанного года `year`.

Пример:

```
DaysInMonth(2024, 4) = 30
```

Hour

Функция	Параметры	Возвращаемое значение
`Hour`	`DateTime date`	`int`

Извлекает час (0-23) из указанной даты.

Пример:

```
Hour(new DateTime(2024,4,2,5,30,5)) = 5
Hour(ToDateTime("2.4.2024 5:30:05")) = 5
Hour([Employees.BirthDate]) = 0
```

Minute

Функция	Параметры	Возвращаемое значение
`Minute`	`DateTime date`	`int`

Извлекает минуты (0-59) из указанной даты.

Пример:

```
Minute(new DateTime(2024,4,2,5,30,5)) = 30
Minute(ToDateTime("2.4.2024 5:30:05")) = 30
Minute([Employees.BirthDate]) = 0
```

Month

Функция	Параметры	Возвращаемое значение
`Month`	`DateTime date`	`int`

Извлекает месяц (1-12) из указанной даты.

Пример:

```
Month(new DateTime(2024,4,2)) = 4
Month(ToDateTime("2.4.2024")) = 4
Month([Employees.BirthDate]) = 1
```

MonthName

Функция	Параметры	Возвращаемое значение
`MonthName`	`int month`	`string`

Возвращает локализованное название месяца (Январь..Декабрь) с номером `month`.

Пример:

```
MonthName(1) = "Январь"
```

Second

Функция	Параметры	Возвращаемое значение
`Second`	`DateTime date`	`int`

Извлекает секунды (0-59) из указанной даты.

Пример:

```
Second(new DateTime(2024,4,2,5,30,5)) = 5
Second(ToDateTime("2.4.2024 5:30:05")) = 5
Second([Employees.BirthDate]) = 0
```

Year

Функция	Параметры	Возвращаемое значение
`Year`	`DateTime date`	`int`

Извлекает год из указанной даты.

Пример:

```
Year(new DateTime(2024,4,2)) = 2024
Year(ToDateTime("2.4.2024")) = 2024
Year([Employees.BirthDate]) = 1986
```

Форматирование

Format

Функция	Параметры	Возвращаемое значение
`Format`	`string format, params object[] args`	`string`

Заменяет каждый элемент формата в указанной строке `format` значением соответствующего параметра в массиве `args`.

Например, следующий вызов функции:

```
Format("Name = {0}, hours = {1:hh}", myName, DateTime.Now)
```

содержит элементы формата `{0}` и `{1:hh}`. Они заменяются значениями `myName` и `DateTime.Now`. Результат может выглядеть так:

```
Name = Alex, hours = 12
```

Элемент формата имеет следующий синтаксис:

```
{index[,alignment][:formatString]}
```

где

- `index` - обязательный номер значения, которое будет подставлено в это место строки;
- `alignment` - необязательная ширина форматированного значения;
- `formatString` - необязательный спецификатор формата.

Для форматирования числовых значений используются следующие стандартные символы-спецификаторы:

Символ	Тип	Описание
C или c	Валюта	Число преобразуется в строку, представляющую денежные единицы. `Format("{0:C}", 10) = "10,00 ₽"`
D или d	Десятичное число	Этот формат доступен только для целых типов. Число преобразуется в строку, состоящую из десятичных цифр (0-9). `Format("{0:D}", 10) = "10"`
E или e	Научный	Число преобразуется в строку вида `-d.ddd…E+ddd` или `-d.ddd…e+ddd`, где знак `d` представляет цифру (0-9). `Format("{0:E}", 10) = "1,000000E+001"`

Символ	Тип	Описание
F или f	Фиксированная запятая	Число преобразуется в строку вида `-ddd.ddd…`, где знак `d` представляет цифру (0-9). `Format("{0:F}", 10) = "10,00"`
G или g	Общий	Число преобразуется в наиболее короткую запись. `Format("{0:G}", 10) = "10"`
N или n	Число	Число преобразуется в строку вида `-d,ddd,ddd.ddd…`, где знак `d` - цифра (0-9), знак `,` - разделитель тысяч, а знак `.` - разделитель целой и дробной части. `Format("{0:N}", 1234.56) = "1 234,56"`
P или p	Процент	Число преобразуется в строку, представляющую проценты. Преобразуемое число умножается на 100, чтобы соответствовать процентам. `Format("{0:P}", 0.15) = "15,00%"`
X или x	Шестнадцатеричный	Этот формат доступен только для целых типов. Число преобразуется в строку шестнадцатеричных знаков. Регистр шестнадцатеричных знаков, превосходящих 9, совпадает с регистром указателя формата. `Format("{0:X}", 26) = "1A"`

После символа-спецификатора формата можно указать число. Для чисел с плавающей запятой это будет означать количество знаков после запятой в форматированном значении:

```
Format("{0:C1}", 12.23) = "12,2 ₽"
```

Если стандартных средств форматирования чисел недостаточно, можно использовать символы настраиваемого числового формата:

Символ	Описание
0	Знак-заменитель нуля. Если форматируемое значение содержит цифру в этой позиции, она копируется в выходную строку. В противном случае 0 отображается в выходной строке.
#	Заместитель цифры. Если форматируемое значение содержит цифру в этой позиции, она копируется в выходную строку. В противном случае в выходной строке ничего не записывается.
.	Разделитель дроби. Первый знак `.` в строке формата определяет положение разделителя дроби.
,	Разделитель числовых разрядов. Знак `,` в строке формата определяет положение разделителя групп разрядов.
%	Заместитель процентов. При использовании знака `%` в строке форматирования число будет умножено на 100, и в соответствующую позицию в выходной строке будет вставлен символ `%`.
;	Разделитель секций. Знак `;` служит для разделения секций положительных, отрицательных и нулевых чисел в строке формата.

Примеры использования:

```
Format("{0:#,##0.00 ₽}", 1024.25) = "1 024,25 ₽"
Format("{0:00%}", 0.25) = "25%"
Format("{0:#,##0.00 ₽;(#,##0.00 ₽);Zero}", 1024.25) = "1 024,25 ₽"
Format("{0:#,##0.00 ₽;(#,##0.00 ₽);Zero}", -1024.25) = "(1 024,25 ₽)"
Format("{0:#,##0.00 ₽;(#,##0.00 ₽);Zero}", 0) = "Zero"
```

Для форматирования даты/времени используются следующие стандартные символы-спецификаторы:

Символ	Тип	Пример
d	Короткий шаблон даты	"02.04.2024"
D	Полный шаблон даты	"2 апреля 2024 г."
f	Полный шаблон даты и времени (короткий шаблон времени)	"2 апреля 2024 г. 20:41"
F	Полный шаблон даты и времени (полный шаблон времени)	"2 апреля 2024 г. 20:41:33"
g	Общий шаблон даты и времени (короткий шаблон времени)	"02.04.2024 20:41"
G	Общий шаблон даты и времени (полный шаблон времени)	"02.04.2024 20:41:33"
t	Короткий шаблон времени	"20:41"
T	Полный шаблон времени	"20:41:33"

Так же можно использовать символы настраиваемого формата даты/времени:

Символ	Описание
d	Представляет день месяца в виде числа от 1 до 31. Одноразрядные числа форматируются без нуля в начале.
dd	Представляет день месяца в виде числа от 01 до 31.
ddd	Представляет сокращенное название дня недели.
dddd	Представляет полное название дня недели.
f или F	Представляет миллисекунды.
h	Представляет часы числом от 1 до 12. Одноразрядные числа форматируются без нуля в начале.
hh	Представляет часы числом от 01 до 12.
H	Представляет часы числом от 0 до 23. Одноразрядные числа форматируются без нуля в начале.
HH	Представляет часы числом от 00 до 23.
m	Представляет минуты как число от 0 до 59. Одноразрядные числа форматируются без нуля в начале.

Символ	Описание
mm	Представляет минуты как число от 00 до 59.
M	Представляет месяц как число от 1 до 12. Одноразрядные числа форматируются без нуля в начале.
MM	Представляет месяц как число от 01 до 12.
MMM	Представляет сокращенное название месяца.
MMMM	Представляет полное название месяца.
s	Представляет секунды как число от 0 до 59. Одноразрядные числа форматируются без нуля в начале.
ss	Представляет секунды как число от 00 до 59.
y	Представляет год как число из одной или двух цифр.
yy	Представляет год как число из двух цифр.
yyyy	Представляет год как число из четырех цифр.
:	Представляет стандартный разделитель времени.
/	Представляет стандартный разделитель даты.

Примеры использования:

```
Format("{0:d MMM yyyy}", DateTime.Now) = "2 апр 2024"
Format("{0:dd/MM/yyyy}", DateTime.Now) = "02.04.2024"
Format("{0:d MMMM}", DateTime.Now) = "2 апреля"
Format("{0:HH:mm}", DateTime.Now) = "20:41"
Format("{0:dd/MM/yyyy HH:mm}", DateTime.Now) = "02.04.2024 20:41"
```

FormatCurrency

Функция	Параметры	Возвращаемое значение
`FormatCurrency`	`object value`	`string`

Форматирует значение `value` как валюту, используя региональные установки Windows.

Пример:

```
FormatCurrency(1.25) = "1,25 ₽"
```

Функция	Параметры	Возвращаемое значение
`FormatCurrency`	`object value, int decimalDigits`	`string`

Форматирует значение `value` как валюту, округляя результат до количества знаков после запятой `decimalDigits`.

Пример:

```
FormatCurrency(1.25, 1) = "1,3 ₽"
```

FormatDateTime

Функция	Параметры	Возвращаемое значение
`FormatDateTime`	`DateTime value`	`string`

Форматирует значение `value` как дату/время. Данная функция не возвращает дату или время, если соответствующие части имеют пустые значения. Пустой датой считается дата 1/1/1, пустым временем - 0:00:00.

Пример:

```
FormatDateTime(new DateTime(2024,4,2)) = "02.04.2024"
FormatDateTime(new DateTime(2024,4,2,1,30,0)) = "02.04.2024 1:30:00"
FormatDateTime(new DateTime(1,1,1,1,30,1)) = "1:30:01"
```

Функция	Параметры	Возвращаемое значение
`FormatDateTime`	`DateTime value, string format`	`string`

Форматирует значение `value` как дату/время, используя именованный формат `format`. Допустимые значения для параметра `format`:

- Long Date;
- Short Date;
- Long Time;
- Short Time.

Пример:

```
FormatDateTime(new DateTime(2024,4,2,1,30,0), "Long Date") = "2 апреля 2024 г."
FormatDateTime(new DateTime(2024,4,2), "Short Date") = "02.04.2024"
FormatDateTime(new DateTime(1,1,1,1,30,1), "Long Time") = "1:30:01"
FormatDateTime(new DateTime(1,1,1,1,30,1), "Short Time") = "01:30"
```

FormatNumber

Функция	Параметры	Возвращаемое значение
`FormatNumber`	`object value`	`string`

Форматирует значение `value` как число, используя региональные установки Windows.

Пример:

```
FormatNumber(1234.56) = "1 234,56"
```

Функция	Параметры	Возвращаемое значение
`FormatNumber`	`object value, int decimalDigits`	`string`

Форматирует значение `value` как число, округляя результат до количества знаков после запятой `decimalDigits`.

Пример:

```
FormatNumber(1234.56, 1) = "1 234,6"
```

FormatPercent

Функция	Параметры	Возвращаемое значение
`FormatPercent`	`object value`	`string`

Форматирует значение `value` как процент, используя региональные установки Windows.

Пример:

```
FormatPercent(0.15) = "15,00%"
```

Функция	Параметры	Возвращаемое значение
`FormatPercent`	`object value, int decimalDigits`	`string`

Форматирует значение `value` как процент, округляя результат до количества знаков после запятой `decimalDigits`.

Пример:

```
FormatPercent(0.15, 0) = "15%"
```

Конвертирование

ToBoolean

Функция	Параметры	Возвращаемое значение
`ToBoolean`	`object value`	`bool`

Конвертирует значение `value` в логический тип.

Пример:

```
ToBoolean(1) = true
ToBoolean(0) = false
```

ToByte

Функция	Параметры	Возвращаемое значение
`ToByte`	`object value`	`byte`

Конвертирует значение `value` в тип `byte`.

Пример:

```
ToByte("55") = 55
```

ToChar

Функция	Параметры	Возвращаемое значение
`ToChar`	`object value`	`char`

Конвертирует значение `value` в символ.

Пример:

```
ToChar(65) = 'A'
```

ToDateTime

Функция	Параметры	Возвращаемое значение
`ToDateTime`	`object value`	`DateTime`

Конвертирует значение `value` в тип `DateTime`.

Допускается использование разделителей `.`, `,` и `/`.

Пример:

```
ToDateTime("2.4.2024") = 02.04.2024 0:00:00
ToDateTime("2,4,2024") = 02.04.2024 0:00:00
ToDateTime("2/4/2024") = 02.04.2024 0:00:00
```

ToDecimal

Функция	Параметры	Возвращаемое значение
`ToDecimal`	`object value`	`decimal`

Конвертирует значение `value` в тип `decimal`.

Пример:

```
ToDecimal(1) = 1m
ToDecimal("1") = 1m
```

ToDouble

Функция	Параметры	Возвращаемое значение
`ToDouble`	`object value`	`double`

Конвертирует значение `value` в тип `double`.

Пример:

```
ToDouble(1) = 1
ToDouble("1") = 1
```

ToInt32

Функция	Параметры	Возвращаемое значение
`ToInt32`	`object value`	`int`

Конвертирует значение `value` в тип `int`.

Пример:

```
ToInt32(1f) = 1
ToInt32("1") = 1
```

ToRoman

Функция	Параметры	Возвращаемое значение
`ToRoman`	`object value`	`string`

Конвертирует числовое значение `value` в римские цифры. `value` не должно превышать значения 3998.

Пример:

```
ToRoman(9) = "IX"
```

ToSingle

Функция	Параметры	Возвращаемое значение
`ToSingle`	`object value`	`float`

Конвертирует значение `value` в тип `float`.

Пример:

```
ToSingle(1m) = 1f
ToSingle("1") = 1f
```

ToString

Функция	Параметры	Возвращаемое значение
`ToString`	`object value`	`string`

Конвертирует значение `value` в тип `string`.

Пример:

```
ToString(false) = "False"
ToString(DateTime.Now) = "02.04.2024 20:41:08"
```

ToWords

Функция	Параметры	Возвращаемое значение
`ToWords`	`object value`	`string`

Конвертирует числовое значение `value` в сумму прописью на английском.

Пример:

```
ToWords(1024.25) = "One thousand and twenty-four dollars and 25 cents"
```

Функция	Параметры	Возвращаемое значение
`ToWords`	`object value, string currencyName`	`string`

Конвертирует числовое значение `value` в сумму прописью на английском. Используется валюта, заданная в параметре `currencyName`. Возможные значения этого параметра:

- USD;
- EUR;
- GBP.

Пример:

```
ToWords(1024.25, "EUR") = "One thousand and twenty-four euros and 25 cents"
```

Функция	Параметры	Возвращаемое значение
`ToWords`	`object value, string one, string many`	`string`

Конвертирует целочисленное значение `value` в число прописью на английском. В параметрах `one` и `many` задаются словоформы для единственного и множественного числа.

Пример:

```
ToWords(124, "page", "pages") = "One hundred and twenty-four pages"
ToWords(1, "page", "pages") = "One page"
```

ToWordsEnGb

Функция	Параметры	Возвращаемое значение
`ToWordsEnGb`	`object value`	`string`

Конвертирует числовое значение `value` в сумму прописью на британском английском. Отличие от функции `ToWords` в следующем:

- по умолчанию используется валюта GBP;

- по-разному переводятся суммы "миллиард" и "триллион".

Пример:

```
ToWordsEnGb(121) = "One hundred and twenty-one pounds and 00 pence"
```

Функция	Параметры	Возвращаемое значение
`ToWordsEnGb`	`object value, string currencyName`	`string`

Конвертирует числовое значение `value` в сумму прописью на британском английском. Используется валюта, заданная в параметре `currencyName`. Возможные значения этого параметра:

- USD;
- EUR;
- GBP.

Пример:

```
ToWordsEnGb(1024.25, "EUR") = "One thousand and twenty-four euros and 25 cents"
```

Функция	Параметры	Возвращаемое значение
`ToWordsEnGb`	`object value, string one, string many`	`string`

Конвертирует целочисленное значение `value` в число прописью на британском английском. В параметрах `one` и `many` задаются словоформы для единственного и множественного числа.

Пример:

```
ToWordsEnGb(124, "page", "pages") = "One hundred and twenty-four pages"
ToWordsEnGb(1, "page", "pages") = "One page"
```

ToWordsRu

Функция	Параметры	Возвращаемое значение
`ToWordsRu`	`object value`	`string`

Конвертирует числовое значение `value` в сумму прописью на русском.

Пример:

```
ToWordsRu(1024.25) = "Одна тысяча двадцать четыре рубля 25 копеек"
```

Функция	Параметры	Возвращаемое значение
`ToWordsRu`	`object value, string currencyName`	`string`

Конвертирует числовое значение `value` в сумму прописью на русском. Используется валюта, заданная в параметре `currencyName`. Возможные значения этого параметра:

- RUR;
- UAH;
- USD;
- EUR.

Пример:

```
ToWordsRu(1024.25, "EUR") = "Одна тысяча двадцать четыре евро 25 евроцентов"
```

Функция	Параметры	Возвращаемое значение
`ToWordsRu`	`object value, bool male, string one, string two, string many`	`string`

Конвертирует целочисленное значение `value` в число прописью на русском. В параметре `male` надо указать `true`, если существительное - мужского рода. В параметрах `one`, `two` и `many` задаются словоформы для чисел 1, 2 и 5 (1 "лист", 2 "листа", 5 "листов").

Пример:

```
// слово "страница" женского рода - параметр male = false
ToWordsRu(124, false, "страница", "страницы", "страниц") = "Сто двадцать четыре страницы"
// слово "лист" мужского рода - параметр male = true
ToWordsRu(124, true, "лист", "листа", "листов") = "Сто двадцать четыре листа"
```

Условия

Choose

Функция	Параметры	Возвращаемое значение
`Choose`	`double index, params object[] choice`	`object`

Возвращает элемент массива `choice` с индексом `index`. Первый элемент массива имеет индекс = 1.

Пример:

```
Choose(2, "one", "two", "three") = "two"
```

IIf

Функция	Параметры	Возвращаемое значение
`IIf`	`bool expression, object truePart, object falsePart`	`object`

Если `expression` равно `true`, возвращает значение `truePart`, иначе возвращает `falsePart`.

Пример:

```
IIf(2 > 5, "true", "false") = "false"
```

Switch

Функция	Параметры	Возвращаемое значение
`Switch`	`params object[] expressions`	`object`

В параметр `expressions` передаются пары вида "условие-значение". Возвращает первое значение, условие для которого равно `true`.

Пример:

```
// вернет одну из строк "a больше 0","a меньше 0","a равно 0" в зависимости от a
Switch(
  a > 0, "a больше 0",
  a < 0, "a меньше 0",
  a == 0, "a равно 0")
```

IsNull

Функция	Параметры	Возвращаемое значение
`IsNull`	`string name`	`bool`

Параметр `name` может представлять собой имя столбца базы данных, имя параметра или имя итога, и должен быть заключен в двойные кавычки.

Если значение объекта с именем `name` равно `null`, метод возвращает `true`, в противном случае - `false`.

Пример:

```
IsNull("Parameter")
```

Важно! Если включено свойство отчета ConvertNulls, и если значение объекта с именем `name` равно `null`, то оно будет преобразовано и функция вернет `false` в любом случае.

Итоговые значения

В большинстве отчетов надо выводить некую итоговую информацию: сумма по группе, количество строк в списке и т.п. В FastReport для этих целей существуют так называемые итоговые значения. С их помощью можно подсчитать функцию от определенного значения по диапазону данных.

Для итога нужно указать следующие параметры:

1. тип итоговой функции;
2. диапазон данных, для которого будет вычисляться функция;
3. выражение, которое надо вычислять. Для функции "Количество" выражение не указывается;
4. выражение – условие. Функция будет вычисляться, если это условие выполняется. Этот параметр задавать не обязательно.

Диапазон данных задается следующим образом: указывается бэнд "Данные", для каждой строки которого будет вычисляться функция. Также указывается бэнд, на котором будет печататься итог. После печати этого бэнда значение итога сбрасывается.

Ниже приведен список итоговых функций:

Функция	Описание
Сумма	Вычисляет сумму указанного выражения для диапазона данных.
Минимум	Вычисляет минимальное значение указанного выражения для диапазона данных.
Максимум	Вычисляет максимальное значение указанного выражения для диапазона данных.
Среднее	Находит среднее значение указанного выражения для диапазона данных.
Количество	Возвращает количество строк в диапазоне.
Количество разных	Возвращает количество разных строк в диапазоне.

Создание итога

Рассмотрим использование итоговой функции на примере. Допустим, у нас есть две таблицы – Categories (Категории) и Products (Продукты). Таблицы связаны таким образом, что мы можем создать отчет типа "главный-подчиненный", который печатает список продуктов в каждой категории, а также количество товара на складе:

Готовый отчет выглядит следующим образом:

Beverages

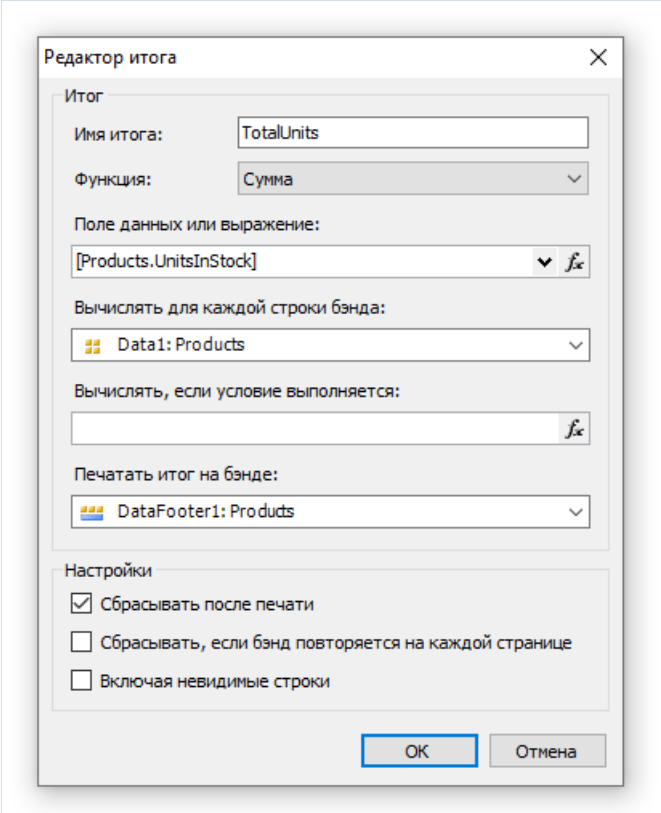
Product Name	Units In Stock
Chai	39
Chang	17
Chartreuse verte	69
Côte de Blaye	17
Guaraná Fantástica	20
Ipoh Coffee	17
Lakkalikööri	57
Laughing Lumberjack Lager	52
Outback Lager	15
Rhönbräu Klosterbier	125
Sasquatch Ale	111
Steeleye Stout	20

Condiments

Product Name	Units In Stock
Aniseed Syrup	13
Chef Anton's Cajun Seasoning	53
Chef Anton's Gumbo Mix	0
Genen Shouyu	39
Grandma's Boysenberry Spread	120
Gula Malacca	27
Louisiana Fiery Hot Pepper Sauce	76
Louisiana Hot Spiced Okra	4
Northwoods Cranberry Sauce	6
Original Frankfurter grüne Soße	32
Sirop d'érable	113
Vegie-spread	24

Добавим итог в этот отчет, который будет печатать общее количество товара на складе для каждой категории - сумму поля `UnitsInStock`. Итог будет печататься в бэнде - подвале данных.

Для того чтобы напечатать итоговое значение, его сначала надо создать. Для этого в окне "Данные" нажмите кнопку "Действия" и выберите пункт меню "Новый итог...". Другой способ – щелкните правой кнопкой мыши на элементе "Итоги" в дереве данных и выберите пункт меню "Новый итог...". Вы увидите окно редактора итогового значения:

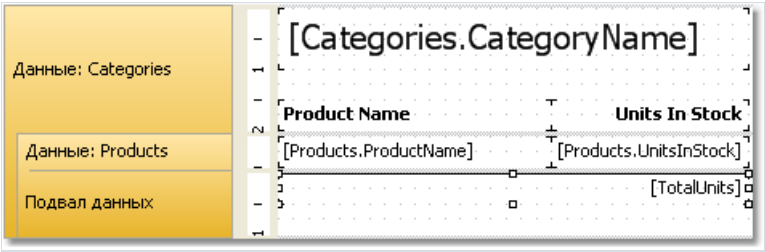


Первое, что вам предлагается сделать, это указать имя итога. Вы будете обращаться к итогу по его имени, поэтому назовите итог так, чтобы было понятно, что он вычисляет. Назовем наш итог `TotalUnits`.

Далее выбираем итоговую функцию. Нас интересует функция "Сумма".

Теперь мы должны указать диапазон данных, для которого будет вычисляться итог. Для этого в поле "Вычислять для каждой строки бэнда" выбираем бэнд "Данные", в котором печатается список продуктов. В поле "Печатать итог на бэнде" выбираем бэнд, в котором будет печататься итог – подвал бэнда "Данные".

Закройте редактор итога кнопкой OK. Вы увидите, что новый итог появился в окне "Данные". Теперь его можно перетащить в отчет:



Если запустить отчет на выполнение, мы увидим следующее:

Beverages

Product Name	Units In Stock
Chai	39
Chang	17
Chartreuse verte	69
Côte de Blaye	17
Guaraná Fantástica	20
Ipoh Coffee	17
Lakkalikööri	57
Laughing Lumberjack Lager	52
Outback Lager	15
Rhönbräu Klosterbier	125
Sasquatch Ale	111
Steeleye Stout	20
	559

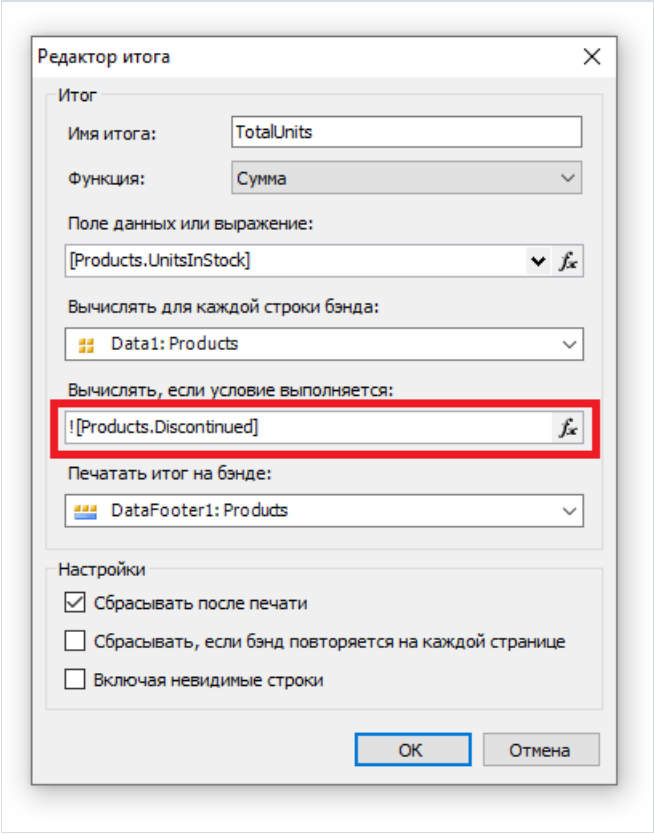
Condiments

Product Name	Units In Stock
Aniseed Syrup	13
Chef Anton's Cajun Seasoning	53
Chef Anton's Gumbo Mix	0
Genen Shouyu	39
Grandma's Boysenberry Spread	120
Gula Malacca	27
Louisiana Fiery Hot Pepper Sauce	76
Louisiana Hot Spiced Okra	4
Northwoods Cranberry Sauce	6
Original Frankfurter grüne Soße	32
Sirop d'érable	113
Vegie-spread	24
	507

Вычисление итога по условию

В предыдущем примере итог вычислялся для всего диапазона данных. Мы можем ограничить диапазон, указав условие в редакторе итога. Итог будет вычисляться только для тех строк диапазона, для которых условие вернет истину (`true`).

Например, мы можем указать следующее условие:

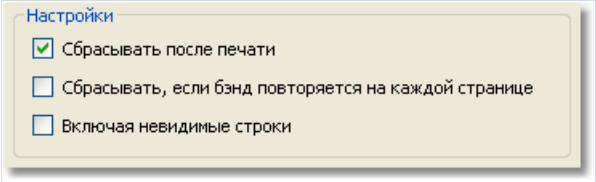


Это будет означать, что итог надо вычислять для тех продуктов, у которых флаг `Discontinued` не установлен.

Печать нарастающего итога

В нашем примере итоги сбрасывались после печати бэнда "Подвал данных". Это происходило потому, что мы указали в редакторе, что итог необходимо сбрасывать после печати этого бэнда. Таким образом, каждая категория продуктов печатала свои собственные значения итогов.

Если выключить флажок "Сбрасывать после печати", итог не будет сбрасываться после печати. Это так называемый нарастающий итог.



Если вам необходимо печатать два вида итогов одновременно - обычные итоги и нарастающие, создайте еще один итог с аналогичными настройками, и выключите флажок "Сбрасывать после печати".

Итоги по странице

Для того чтобы создать итог, который будет печататься в подвале страницы, укажите бэнд - подвал страницы - в поле "Печатать на бэнде".

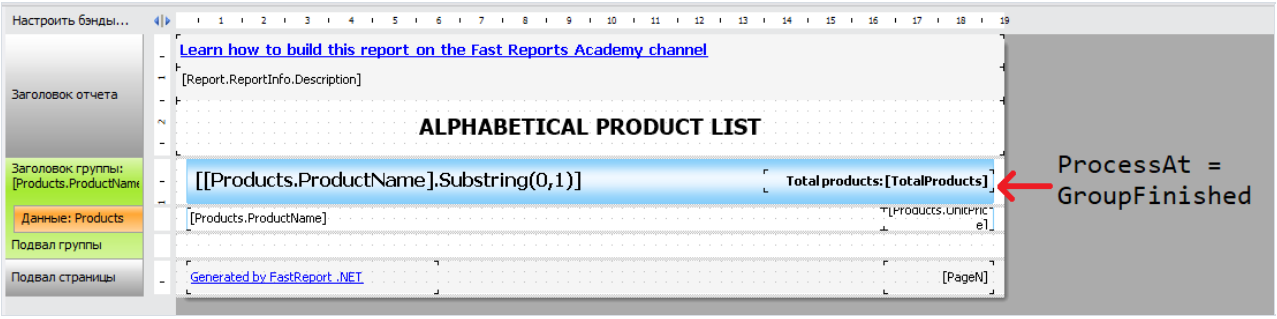
Печать итога в заголовке

Обычно итоговые значения печатаются на бэндах-подвалах (подвал данных, подвал группы, и т.п.). Это естественный порядок печати: в этот момент итог уже вычислен и готов к применению. Однако часто необходимо напечатать итог, например, в заголовке группы. Если попытаться это сделать, мы увидим нулевое значение. В этот момент итог еще не начинал вычисляться, его значение будет готово в подвале группы.

Для решения этой проблемы FastReport предлагает механизм "отложенной" печати. Объект "Текст" имеет свойство `ProcessAt`, которое может принимать одно из следующих значений:

Значение	Описание
Default	Обычный режим, значение по умолчанию.
ReportFinished	Значение объекта обновляется при завершении отчета.
ReportPageFinished	Значение объекта обновляется при завершении всех бэндов на странице отчета.
PageFinished	Значение объекта обновляется печати подвала страницы.
ColumnFinished	Значение объекта обновляется при печати подвала колонки.
DataFinished	Значение объекта обновляется при печати подвала данных.
GroupFinished	Значение объекта обновляется при печати подвала группы.

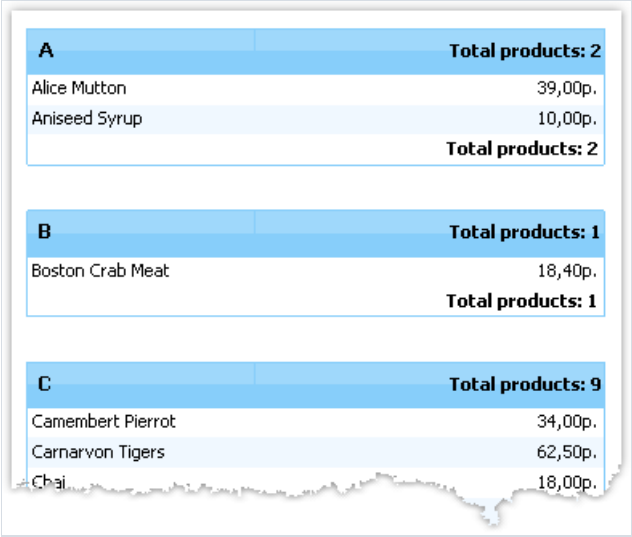
Поясним принцип работы на примере. Положим на заголовок группы объект "Текст", который печатает итог по группе. Свойство `ProcessAt` установим в `GroupFinished`:



При запуске отчета произойдет следующее:

- будет напечатан заголовок группы. При этом FastReport напечатает нулевой (неправильный) итог и запомнит это место в готовом отчете;
- будут напечатаны все строки данных группы;
- будет напечатан подвал группы. В этот момент FastReport возьмет ранее запомненный объект и обработает его еще раз, на этот раз с правильным значением итога.

Готовый отчет выглядит так:



Аналогичным образом можно печатать итог по всему отчету в его заголовке (установите `ProcessAt = ReportFinished`) или итог по странице в заголовке страницы (`ProcessAt = PageFinished`).

> Механизм отложенной печати не работает, если включен файловый кэш (меню "Отчет|Свойства...", флажок "Использовать файловый кэш").

Параметры отчета

В отчете можно определить параметры. Параметр – это переменная, значение которой может быть определено как в самом отчете, так и вне его (программа, вызывающая отчет, может передавать в него значения параметров – подробнее об этом см. "Руководство программиста"). Параметр может использоваться в выражениях, выводиться в объектах типа "Текст".

Наиболее частые способы использования параметров:

- фильтрация данных по условию, заданному в параметре;
- печать значения параметра в отчете.

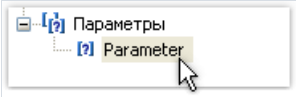
Параметр имеет следующие свойства:

Свойство	Описание
Name	Имя параметра может содержать любые символы, кроме точки `.`.
DataType	Тип данных параметра.
Expression	Выражение, которое возвращает значение параметра. Подробнее о выражениях смотрите в главе "Выражения". Это выражение будет вычислено при обращении к параметру.
Value	Значение параметра. Это свойство недоступно в дизайнере и может быть заполнено программным способом.

Вы должны настроить свойства `Name` и `DataType`. Свойство `Expression` можно оставить пустым. В этом случае значение параметра надо передавать программным способом.

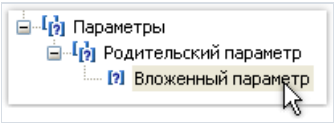
Создание параметра

Чтобы создать параметр, выберите элемент "Параметры отчета" в окне "Данные" и выберите пункт "Новый параметр" в контекстном меню:



Нажмите клавишу F2 и задайте желаемое имя параметра, затем переключитесь в окно "Свойства" и укажите тип данных параметра (свойство `DataType`).

Параметры могут быть вложенными. Чтобы создать вложенный параметр, выберите родительский параметр и выберите пункт "Новый параметр" в контекстном меню:



При этом можно обращаться как к родительскому параметру, так и к вложенному. Количество уровней вложенности не ограничено.

Использование параметров в отчете

К параметру можно обращаться из выражения, используя квадратные скобки:

```
[Имя параметра]
```

К вложенному параметру надо обращаться следующим образом:

```
[Родительский параметр.Дочерний параметр]
```

Так как параметр имеет определенный тип (он задан в свойстве `DataType`), с параметром можно выполнять действия, разрешенные для типа данных. Так, параметры строкового типа можно использовать в выражении следующим образом:

```
[Строковый параметр].Substring(0, 2)
```

Рассмотрим пример использования параметра. Допустим, у нас есть отчет, печатающий таблицу Employees (Сотрудники). Мы хотим распечатать информацию о сотруднике с указанным номером (поле `EmployeeID`). Для этого создадим параметр с именем "Номер сотрудника". Укажем тип параметра – `Int32`, так как именно этот тип имеет поле `EmployeeID`. Чтобы отфильтровать сотрудника с указанным номером, зайдем в редактор бэнда "Данные" и на закладке "Фильтр" укажем следующее выражение:

```
[Employees.EmployeeID] == [Номер сотрудника]
```

Чтобы передать значение параметра из вашей программы в отчет, используйте следующий код:

```
report1.SetParameterValue("Номер сотрудника", 2);
```

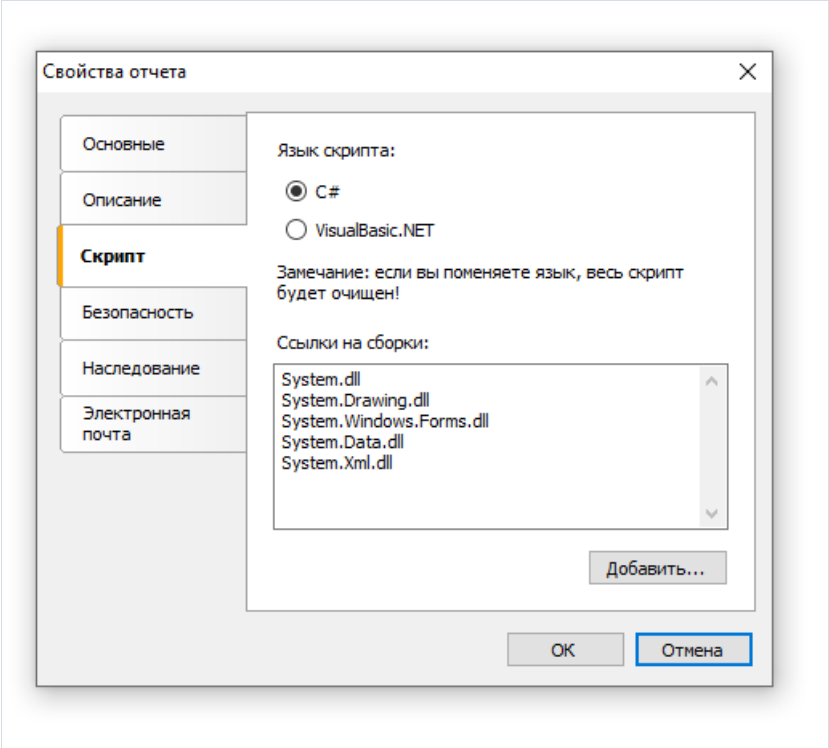
Выражения

Во многих местах в FastReport используются выражения. Так, объект "Текст" может содержать выражения, обрамленные квадратными скобками.

Выражение – это строка кода на языке C# или VB.NET, которая возвращает какое-либо значение. Например:

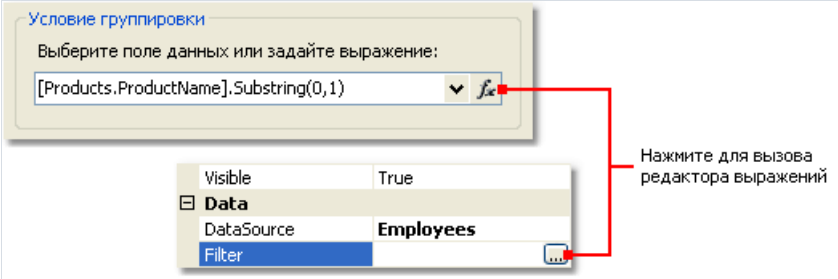
```
2 + 2
```

Выражение должно быть написано на том языке, который выбран в качестве скрипта в данном отчете. По умолчанию это C#. Сменить язык можно в меню "Отчет|Свойства...", выбрав в окне элемент "Скрипт":

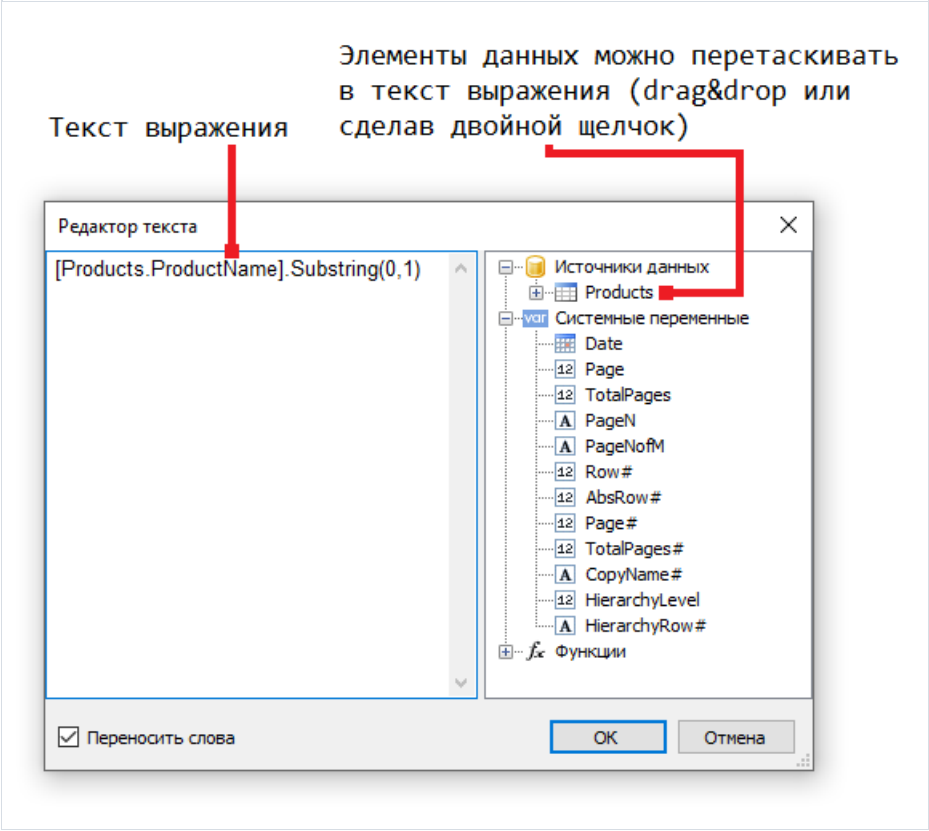


Редактор выражений

Для удобства написания выражения используйте редактор выражений. Его можно вызывать в тех местах интерфейса FastReport, где поддерживается ввод выражений:



Редактор выражений представляет собой окно, в котором можно ввести текст выражения и вставить в него элементы из окна "Данные":



Обращение к объектам отчета

Для обращения к объектам отчета (например, объекту "Текст") используйте имя объекта. Следующий пример вернет высоту объекта Text1:

```
Text1.Height
```

Для обращения к свойствам отчета используйте переменную `Report`. Следующий пример вернет имя файла, из которого был загружен отчет:

```
Report.FileName
```

Кроме этого, вы можете обращаться к вложенным свойствам объекта. Следующий пример вернет название отчета:

```
Report.ReportInfo.Name
```

Использование функций .NET

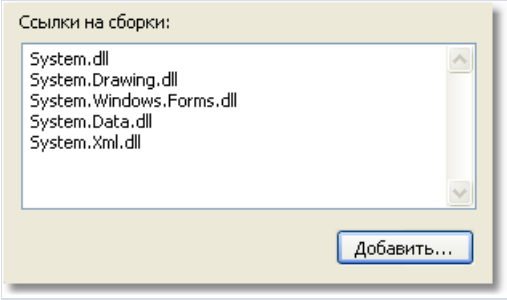
Вы можете использовать любые объекты .NET в выражениях. Следующий пример демонстрирует использование функции `Max`:

```
Math.Max(5, 10)
```

По умолчанию отчет использует следующие сборки .NET:

```
System.dll
System.Drawing.dll
System.Windows.Forms.dll
System.Data.dll
System.Xml.dll
```

Вам доступны все объекты .NET, объявленные в этих сборках. Если вам нужно получить доступ к другой сборке, добавьте ее название в список сборок отчета. Это можно сделать в меню "Отчет|Свойства...", выбрав в окне элемент "Скрипт".



Например, если вы хотите использовать в отчете функцию, объявленную в вашем приложении, добавьте ссылку на сборку приложения (.exe или .dll) в список сборок отчета. После этого можно обращаться к функции, используя пространство имени (namespace) вашего приложения. Например, если в приложении определена следующая функция:

```
namespace Demo
{
  public static class MyFunctions
  {
    public static string Func1()
    {
      return "Hello!";
    }
  }
}
```

Обратиться к ней в отчете можно так:

```
Demo.MyFunctions.Func1()
```

Если добавить в скрипт директиву `using Demo`, это позволит укоротить форму обращения к функции:

```
MyFunctions.Func1()
```

Для обращения к переменным или функциям, которые были вами определены в скрипте отчета, используйте имя переменной или функции:

```
myPrivateVariableThatIHaveDeclaredInScript
MyScriptFunction()
```

> Использовать в выражении можно функции, которые возвращают значение.

Обращение к данным отчета

Помимо стандартных языковых элементов, в выражениях можно использовать следующие элементы отчета:

- поля источников данных;
- системные переменные;
- итоговые значения;
- параметры отчета.

Все эти элементы содержатся в окне "Данные". Подробнее о работе с ними читайте в главе "Данные".

Любой из этих элементов можно использовать в выражении, заключив его в квадратные скобки. Например:

```
[Page] + 1
```

Это выражение возвращает номер следующей печатаемой страницы. В выражении используется системная переменная `Page`, которая возвращает номер текущей страницы отчета. Она заключена в квадратные скобки.

Обращение к источникам данных

Для обращения к полям источников данных используется следующая форма записи:

```
[Имя источника.Имя поля]
```

Имя источника отделяется от имени поля точкой, например:

```
[Employees.FirstName]
```

Имя источника может быть составным в случае, если мы обращаемся к источнику данных, используя связь (relation). Подробнее о связях рассказано в главе "Данные". Например, так можно обратиться к полю связанного источника данных:

```
[Products.Categories.CategoryName]
```

Рассмотрим следующий пример использования полей в выражении:

```
[Employees.FirstName] + " " + [Employees.LastName]
```

Здесь надо сделать важное замечание. Каждое поле имеет определенный тип данных - он задается в свойстве `DataType` поля (его можно увидеть в окне "Свойства", если предварительно выбрать поле данных в окне "Данные"). От того, какой тип имеет поле, зависит, каким образом его можно использовать в выражении. Так, в примере выше, оба поля (имя и фамилия) имеют строковый тип и поэтому их допустимо использовать таким образом. В следующем примере мы попробуем использовать поле `Employees.Age` числового типа, что приведет к ошибке:

```
[Employees.FirstName] + " " + [Employees.Age]
```

Ошибка происходит потому, что нельзя напрямую складывать строку и число. Для этого число надо явным образом преобразовать в строку:

```
[Employees.FirstName] + " " + [Employees.Age].ToString()
```

В данном случае мы обращаемся с полем `Employees.Age` так, будто это целочисленная переменная. Так оно и есть. Мы уже знаем, что все выражения компилируются в исполняемый код. Все нестандартные с точки зрения компилятора вещи (вроде обращения к системным переменным и полям данных) конвертируются в другой вид, понятный компилятору. Так, последнее выражение будет преобразовано в следующий вид:

```
(string)(Report.GetColumnValue("Employees.FirstName")) + " " +
(int)(Report.GetColumnValue("Employees.Age")).ToString()
```

Как видно, FastReport при компиляции выражений заменяет обращения к полям данных следующим образом:

`[Employees.FirstName]` --> `(string)(Report.GetColumnValue("Employees.FirstName"))`

`[Employees.Age]` --> `(int)(Report.GetColumnValue("Employees.Age"))`

То есть, мы можем использовать поле БД в выражениях, как будто это переменная, имеющая определенный тип. Например, следующее выражение вернет первый символ имени сотрудника:

```
[Employees.FirstName].Substring(0, 1)
```

Обращение к системным переменным

В выражениях можно использовать следующие системные переменные (они доступны в окне "Данные"):

Переменная	Тип данных .NET	Описание
Date	DateTime	Дата и время старта отчета.
Page	int	Номер текущей страницы.
TotalPages	int	Общее количество страниц в отчете. Чтобы использовать эту переменную, надо включить двойной проход у отчета. Это можно сделать в меню "Отчет|Свойства...".
PageN	string	Номер страницы в виде: "Страница N".
PageNofM	string	Номер страницы в виде: "Страница N из M".
Row#	int	Номер строки данных внутри группы. Это значение сбрасывается при старте новой группы.
AbsRow#	int	Абсолютный номер строки данных. Это значение не сбрасывается при старте новой группы.

Каждая переменная имеет определенный тип данных, от этого зависит, как ее можно использовать в выражении. Вот пример выражения, в котором используется дата:

```
[Date].Year
```

Это выражение возвращает текущий год. Так как переменная `Date` имеет тип `DateTime`, мы можем обратиться к ее свойству `Year`. Аналогичным образом можно получить текущий месяц (`[Date].Month`).

FastReport преобразует обращение к системной переменной в следующий вид (на примере переменной `Date`):

```
((DateTime)Report.GetVariableValue("Date"))
```

Обращение к итоговым значениям

Для обращения к итоговому значению используйте его имя:

```
[TotalSales]
```

При этом FastReport преобразует обращение к итогу в следующую форму:

```
Report.GetTotalValue("TotalSales")
```

Как видно, здесь тип значения не используется. Это потому, что итоговое значение имеет тип `FastReport.Variant`. Оно может быть напрямую использовано в любых выражениях, потому что тип `FastReport.Variant` автоматически приводится к любому типу. Например:

```
[TotalSales] * 0.2f
```

Обращение к параметрам отчета

Для обращения к параметру отчета используйте его имя:

```
[Parameter1]
```

Параметры могут быть вложенными. В этом случае указывается имя родительского параметра и через точку имя дочернего параметра:

```
[ParentParameter.ChildParameter]
```

Параметры, как и поля данных, и системные переменные, имеют определенный тип данных. Он задается в свойстве `DataType` параметра. От того, какой тип данных имеет параметр, зависит, как его можно использовать в выражении.

FastReport преобразует обращение к параметру отчета в следующий вид (на примере строкового параметра):

```
((string)Report.GetParameterValue("Parameter1"))
```

Скрипт

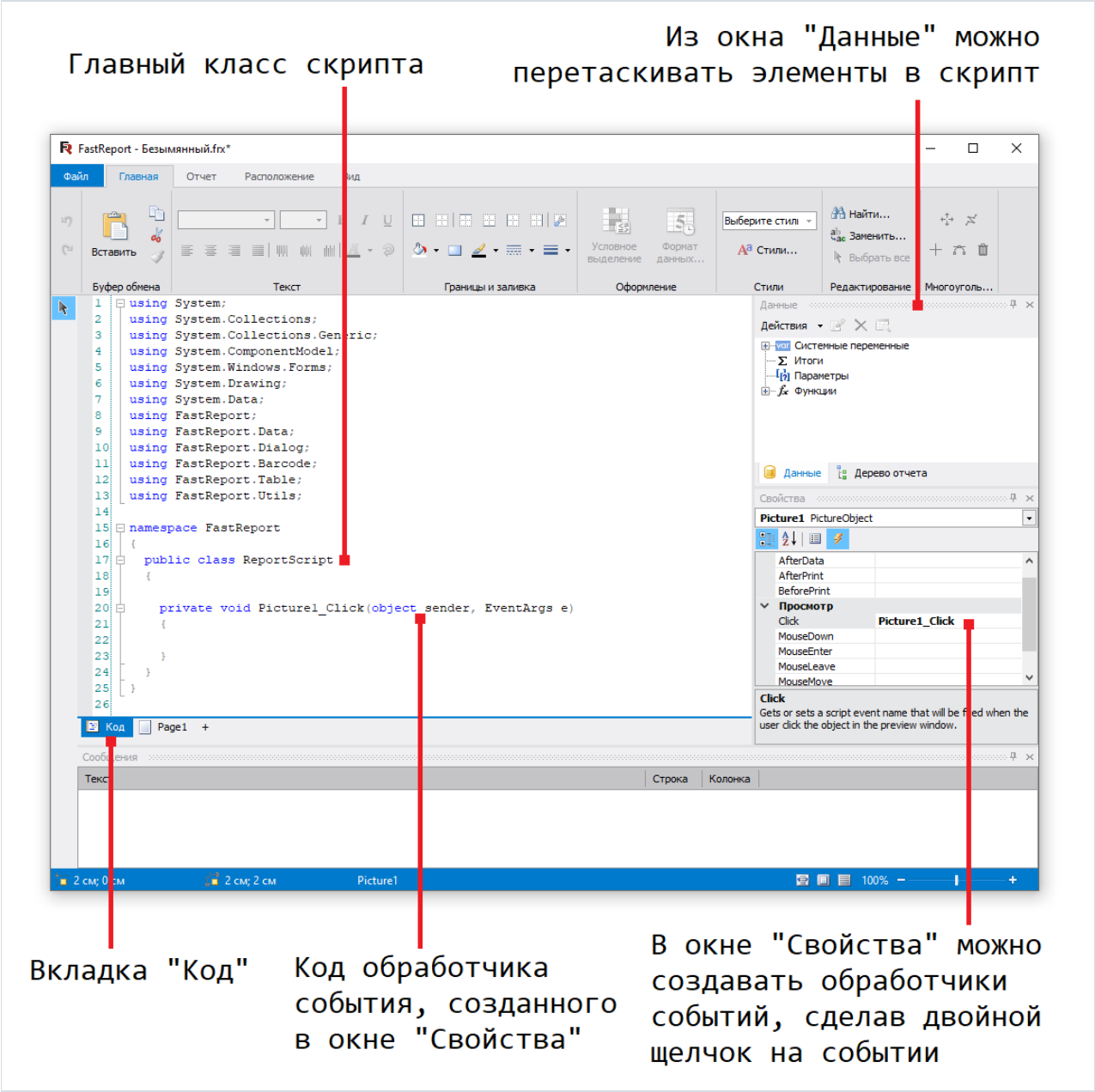
Скрипт – это программа на языке высокого уровня, которая является частью отчета. Скрипт может быть написан на одном из языков .NET:

- C#
- VisualBasic.NET

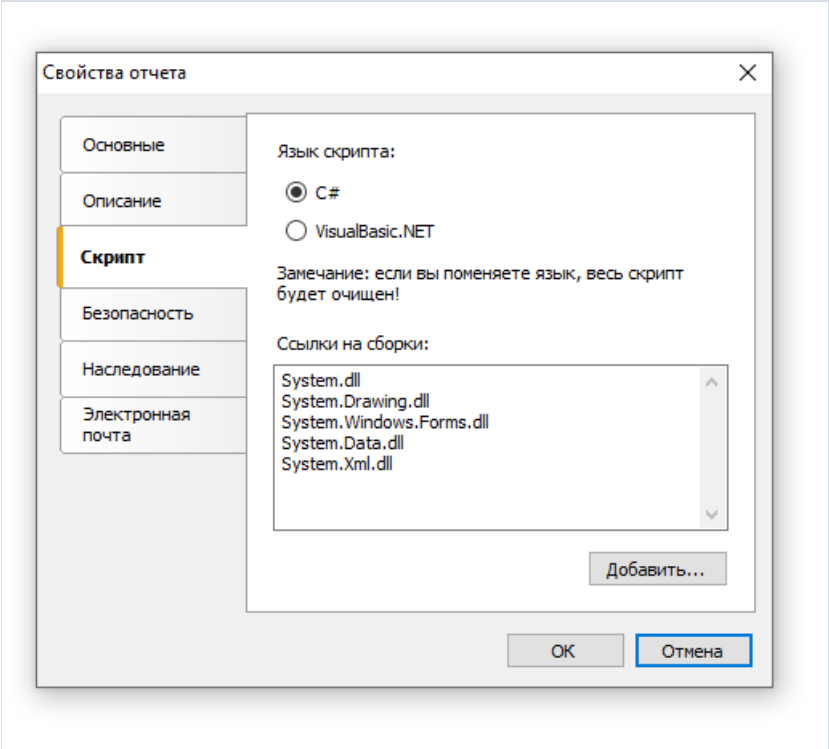
Область применения скрипта довольно обширна. Используя скрипт, вы можете сделать следующее:

- выполнить обработку данных, которую невозможно сделать штатными средствами ядра FastReport;
- управлять печатью страниц отчета и бэндов на странице;
- управлять взаимодействием элементов управления на диалоговых формах;
- управлять формированием динамических объектов "Таблица";
- и многое другое.

Чтобы увидеть скрипт отчета, переключитесь на закладку "Код" в дизайнере:



Язык скрипта настраивается в меню "Отчет|Свойства...". Делать это нужно сразу после того как вы создали новый отчет, так как при смене языка существующий скрипт очищается.



Общая информация

В отличие от других генераторов отчетов, скрипт в FastReport содержит только то, что написано вами. В скрипте вы можете:

- добавлять в главный класс скрипта свои переменные, методы, свойства;
- создавать обработчики событий объектов отчета;
- добавлять новые классы в скрипт, если это необходимо. Класс может быть добавлен как перед главным классом `ReportScript`, так и после него.

Вы не можете:

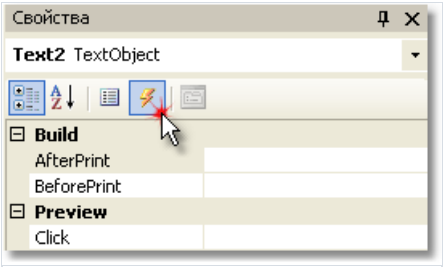
- удалять, переименовывать или изменять область видимости главного класса `ReportScript`;
- переименовывать пространство имен, в котором находится главный класс.

При запуске отчета происходит следующее:

- FastReport добавляет в скрипт список переменных, имена которых совпадают с именами объектов отчета. Это делается перед компиляцией скрипта и позволяет вам обращаться к объектам отчета по их имени;
- в скрипт добавляются выражения, имеющиеся в отчете, в виде функций;
- выполняется компиляция скрипта, если он не пустой;
- инициализируются переменные, которые были неявно добавлены в скрипт;
- обработчики событий, определенные в скрипте, привязываются к объектам отчета;
- запускается отчет.

Обработчики событий

Скрипт главным образом используется для создания обработчиков событий объектов. Для создания обработчика события выделите нужный объект. В окне "Свойства" нажмите кнопку  , чтобы переключиться на список событий:



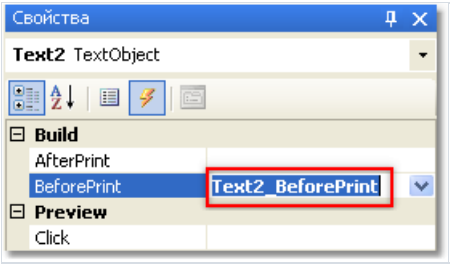
Выберите интересующее вас событие и сделайте на нем двойной щелчок мышью. FastReport добавит в код отчета пустой обработчик события:

```
private void Text2_BeforePrint(object sender, EventArgs e)
{

}
```

Объект "Отчет" также имеет события. Выбрать этот объект можно следующими способами:

- выберите "Report" в окне "Дерево отчета";
- выберите "Report" в выпадающем списке объектов окна "Свойства".

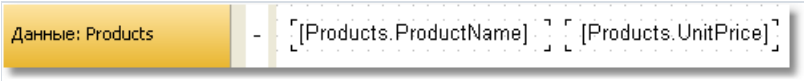
Чтобы удалить обработчик события, выберите событие в окне "Свойства", выделите текст имени события и нажмите клавишу `Delete`:



События

Чтобы максимально гибко управлять отчетом, каждый объект отчета имеет несколько событий, которым можно назначить обработчик – метод из скрипта. Например, в обработчике, привязанном к бэнду "Данные", можно выполнять фильтрацию записей, т.е. скрывать или показывать бэнд в зависимости от каких-либо условий.

Рассмотрим процесс формирования отчета и события, которые при этом генерируются. В качестве примера возьмем простой отчет, содержащий одну страницу, один бэнд "Данные" и два объекта "Текст" на бэнде:



В начале отчета вызывается событие StartReport объекта "Отчет". Перед формированием страницы вызывается событие страницы StartPage. Это событие вызывается один раз для каждой страницы шаблона отчета (не путать со страницами готового отчета!). В нашем случае, сколько бы ни было страниц в готовом отчете – событие вызовется один раз, т.к. шаблон отчета состоит из одной страницы. Затем вызывается событие CreatePage, которое происходит как раз в тот момент, когда страница создается в подготовленном отчете.

Далее начинается печать строк бэнда "Данные". Происходит это следующим образом:

1. вызывается событие бэнда `BeforePrint`;
2. вызываются события `BeforePrint` всех объектов, лежащих на бэнде;
3. все объекты заполняются данными;
4. вызываются события `AfterData` всех объектов, лежащих на бэнде;
5. вызывается событие бэнда `BeforeLayout`;
6. происходит размещение объектов на бэнде (если среди них есть растягиваемые объекты) и подсчет высоты бэнда и его растягивание (если бэнд растягиваемый);
7. вызывается событие бэнда `AfterLayout`;
8. если бэнд не помещается на свободном месте страницы, формируется новая страница;
9. бэнд и все его объекты выводятся на страницу готового отчета;
10. вызывается событие `AfterPrint` бэнда;
11. вызывается событие `AfterPrint` всех объектов бэнда.

Печать строк бэнда происходит до тех пор, пока есть данные в источнике. После этого формирование отчета в нашем случае завершается и вызываются события `FinishPage` страницы отчета и наконец – событие `FinishReport` объекта "Отчет".

Таким образом, используя события разных объектов, можно контролировать практически каждый момент формирования отчета. Ключ к правильному использованию событий – полное понимание процесса печати бэндов, изложенного выше в одиннадцати пунктах.

Так, большинство действий можно выполнить, используя только событие бэнда `BeforePrint` – любые изменения, внесенные в объект, будут тут же отображены. Но в этом событии невозможно анализировать, на какой странице будет напечатан бэнд, если он растягиваемый – ведь подсчет высоты бэнда будет выполнен в пункте 6. Это можно сделать с помощью событий `AfterLayout` в пункте 7 или `AfterPrint` в пункте 10, но в последнем случае бэнд уже будет напечатан и действия над объектами ничего не дадут.

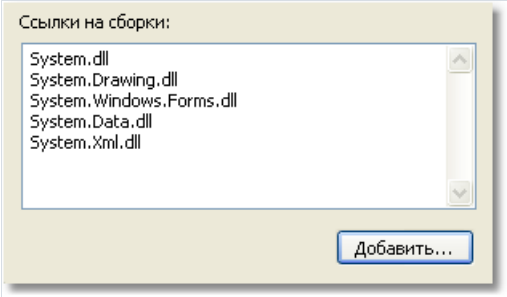
Одним словом, вы должны четко представлять, в какой момент времени вызывается каждое из событий и использовать те, которые соответствуют поставленной задаче.

Обращение к объектам .NET

Из скрипта вы можете обращаться к любым объектам .NET, которые определены в следующих сборках:

```
System.dll
System.Drawing.dll
System.Windows.Forms.dll
System.Data.dll
System.Xml.dll
```

Кроме того, вы можете использовать любые объекты, определенные в сборках FastReport. Если вам нужно получить доступ к другой сборке, добавьте ее название в список сборок отчета. Это можно сделать в меню "Отчет|Свойства...", выбрав в окне элемент "Скрипт".



Например, если вы хотите использовать в скрипте функцию, объявленную в вашем приложении, добавьте ссылку на сборку приложения (.exe или .dll) в список сборок отчета. После этого можно обращаться к функции, используя пространство имени (namespace) вашего приложения. Например, в приложении определена следующая функция:

```
namespace Demo
{
  public static class MyFunctions
  {
    public static string Func1()
    {
      return "Hello!";
    }
  }
}
```

Обратиться к ней в скрипте можно так:

```
string hello = Demo.MyFunctions.Func1();
```

Если добавить в скрипт директиву `using Demo`, это позволит укоротить форму обращения к функции:

```
string hello = MyFunctions.Func1();
```

Обращение к объектам отчета

Для обращения к объектам отчета (например, объекту "Текст") используйте имя объекта. Следующий пример вернет высоту объекта Text1:

```
float height = Text1.Height;
```

Учтите, что "родными" единицами измерения отчета являются экранные пиксели. Вы должны это помнить при обращении к таким свойствам объектов, как `Left`, `Top`, `Width`, `Height`. Для перевода пикселей в сантиметры и обратно используйте константы, определенные в классе `Units`:

```
float heightInPixels = Text1.Height;
float heightInCM = heightInPixels / Units.Centimeters;
Text1.Height = Units.Centimeters * 5; // 5см
```

Объекты Report и Engine

Кроме объектов, которые содержатся в отчете, в скрипте определены две переменные: `Report` и `Engine`.

Переменная `Report` возвращает ссылку на текущий отчет. В таблице ниже приведен список методов объекта Report:

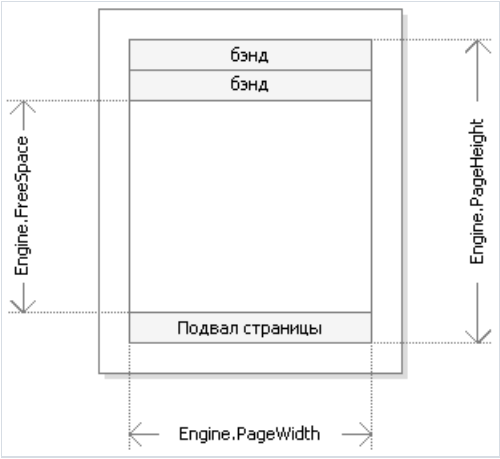
Метод	Описание
`object Calc(string expression)`	Вычисляет выражение и возвращает значение. При вызове этого метода первый раз выражение компилируется, что требует некоторого времени.
`object GetColumnValue(string complexName)`	Возвращает значение поля источника данных. Имя должно быть представлено в форме "Источник.Поле". Если поле имеет значение null, оно преобразуется в значение по умолчанию (0, пустая строка, `false` и др).
`object GetColumnValueNullable(string complexName)`	Возвращает значение поля источника данных. В отличие от предыдущего метода, не выполняет преобразование null-значений.
`Parameter GetParameter(string complexName)`	Возвращает параметр отчета с указанным именем. Имя может быть составным при обращении к вложенному параметру: `MainParam.NestedParam`.
`object GetParameterValue(string complexName)`	Возвращает значение параметра отчета с указанным именем.
`void SetParameterValue(string complexName, object value)`	Устанавливает значение параметра отчета с указанным именем.
`object GetVariableValue(string complexName)`	Возвращает значение системной переменной, например, `Date`.
`object GetTotalValue(string name)`	Возвращает значение итога, определенного в окне "Данные", по его имени.
`DataSourceBase GetDataSource(string alias)`	Возвращает источник данных, определенный в отчете, по его имени.

Объект Engine представляет собой движок, управляющий построением отчета. Используя свойства и методы движка, можно управлять процессом размещения бэндов на странице. Вы можете использовать следующие свойства объекта Engine:

Свойство	Описание
`float CurX`	Текущее смещение координат по оси X. Этому свойству можно присваивать значение, чтобы смещать печатаемые объекты.
`float CurY`	Текущая позиция печати по оси Y. Этому свойству можно присваивать значение, чтобы смещать печатаемые объекты.
`int CurColumn`	Номер текущей колонки в многоколоночном отчете. Первая колонка имеет номер 0.

Свойство	Описание
`int CurPage`	Номер текущей печатаемой страницы. Это значение можно получить из системной переменной `Page`.
`float PageWidth`	Ширина страницы минус размер левого и правого полей.
`float PageHeight`	Высота страницы минус размер верхнего и нижнего полей.
`float PageFooterHeight`	Высота подвала страницы (и всех его дочерних бэндов).
`float ColumnFooterHeight`	Высота подвала колонки (и всех ее дочерних бэндов).
`float FreeSpace`	Размер свободного места на странице.
`bool FirstPass`	Возвращает `true`, если выполняется первый (или единственный) проход отчета. Количество проходов можно получить из свойства `Report.DoublePass`.
`bool FinalPass`	Возвращает `true`, если выполняется последний (или единственный) проход отчета.

На рисунке ниже представлено изображение страницы отчета и название свойств, которые возвращают то или иное измерение страницы.



Свойства `Engine.PageWidth` и `Engine.PageHeight` определяют размер области печати, которая почти всегда меньше физических размеров страницы. Размер области печати определяют поля страницы, которые задаются свойствами страницы отчета `LeftMargin`, `TopMargin`, `RightMargin`, `BottomMargin`.

Свойство `Engine.FreeSpace` определяет высоту свободного места на странице. Если на странице есть бэнд "Подвал страницы", его высота учитывается при вычислении `FreeSpace`. Следует учесть, что после вывода очередного бэнда свободное место на странице уменьшается, что учитывается при вычислении `FreeSpace`.

Как происходит формирование страниц готового отчета? Ядро FastReport выводит бэнды на страницу до тех пор, пока на ней остается свободное место, достаточное для вывода бэнда. Когда свободного места не остается, печатается бэнд "Подвал страницы" (если он есть) и формируется новая пустая страница. Как уже говорилось, после вывода очередного бэнда высота свободного места уменьшается. Кроме того, вывод очередного бэнда начинается с текущей позиции, которая определяется координатами по оси X и Y. Эта позиция возвращается в свойствах `Engine.CurX`, `Engine.CurY`. После печати очередного бэнда позиция `CurY` автоматически увеличивается на высоту напечатанного бэнда. После формирования новой страницы позиция `CurY = 0`. Позиция `CurX` изменяется при печати многоколоночных отчетов.

Свойства `Engine.CurX` и `Engine.CurY` доступны не только для чтения, но и для записи. Это значит, что можно смещать бэнды вручную, используя одно из подходящих событий. Пример использования этих свойств смотрите в разделе "Примеры".

> При работе со свойствами, которые возвращают размер, учтите, что в FastReport используются единицы измерения - экранные пиксели.

В объекте Engine определены следующие методы:

Метод	Описание
`void AddOutline(string text)`	Добавляет элемент в структуру отчета (см. главу "Интерактивные отчеты") и смещает текущую позицию на добавленный элемент.
`void OutlineRoot()`	Смещает текущую позицию на корень структуры.
`void OutlineUp()`	Смещает текущую позицию на уровень выше.
`void AddBookmark(string name)`	Добавляет закладку (см. главу "Интерактивные отчеты").
`int GetBookmarkPage(string name)`	Возвращает номер страницы, на которой расположена закладка с указанным именем.
`void StartNewPage()`	Добавляет в отчет разрыв страницы. Если отчет многоколоночный, добавляется новая колонка.

Используя методы `AddOutline`, `OutlineRoot` и `OutlineUp`, можно формировать структуру отчета программным способом. Обычно это делается автоматически с помощью свойства `OutlineExpression`, которое имеется у каждого бэнда и у страницы отчета.

Метод `AddOutline` добавляет к текущему узлу структуры дочерний узел и делает его текущим. С элементом ассоциируется текущая страница отчета и текущая позиция на странице. Таким образом, если несколько раз подряд вызвать `AddOutline`, то получится "лесенка" типа

```
Item1
    Item2
        Item3
```

Для управления текущим элементом служат методы `OutlineUp` и `OutlineRoot`. Первый метод перемещает указатель на элемент, расположенный уровнем выше. Так, скрипт

```
Engine.AddOutline("Item1");
Engine.AddOutline("Item2");
Engine.AddOutline("Item3");
Engine.OutlineUp();
Engine.AddOutline("Item4");
```

построит структуру вида

```
Item1
    Item2
        Item3
        Item4
```

Метод `OutlineRoot` передвигает текущий элемент в корень структуры. Например, скрипт:

```
Engine.AddOutline("Item1");
Engine.AddOutline("Item2");
Engine.AddOutline("Item3");
Engine.OutlineRoot();
Engine.AddOutline("Item4");
```

построит структуру следующего вида:

```
Item1
    Item2
        Item3
Item4
```

Для работы с закладками используются методы `AddBookmark` и `GetBookmarkPage` объекта Engine. Обычно закладки добавляются автоматически при использовании свойства `Bookmark`, которое имеется у всех объектов отчета.

Используя метод `AddBookmark`, можно добавлять закладку программно. Этот метод создаст закладку на текущей странице, в текущей позиции печати.

Метод `GetBookmarkPage` возвращает номер страницы, на которой расположена закладка. Этот метод часто применяется при создании оглавлений, для отображении номеров страниц. В этом случае отчет должен быть двухпроходным.

Обращение к источникам данных

В отличие от выражений FastReport (они рассмотрены в главе "Выражения"), в скрипте нельзя использовать квадратные скобки для обращения к данным отчета. Вместо этого используется метод `GetColumnValue` объекта Report, возвращающий значение поля:

```
string productName = (string)Report.GetColumnValue("Products.Name");
```

Как видно, надо указать имя источника данных и его поля через точку. Имя источника может быть составным в случае, если мы обращаемся к источнику данных, используя связь (relation). Подробнее о связях рассказано в главе "Данные". Например, так можно обратиться к полю связанного источника данных:

```
string categoryName = (string)Report.GetColumnValue("Products.Categories.CategoryName");
```

Для облегчения работы используйте окно "Данные". Из него можно перетаскивать элементы данных в скрипт, при этом FastReport автоматически создает код для обращения к элементу.

Для обращения к самому источнику данных используйте метод `GetDataSource` объекта Report:

```
DataSourceBase ds = Report.GetDataSource("Products");
```

Справку по свойствам и методам класса `DataSourceBase` можно получить в справочной системе FastReport .NET Class Reference. Как правило, этот объект используется в скрипте следующим образом:

```
// получаем ссылку на источник данных
DataSourceBase ds = Report.GetDataSource("Products");
// инициализируем его
ds.Init();
// перебираем все записи в источнике
while (ds.HasMoreRows)
{
  // получаем значение поля для текущей записи источника
  string productName = (string)Report.GetColumnValue("Products.Name");
  // выполняем с ним какие-то действия...
  // ...
  // переходим на следующую запись
  ds.Next();
}
```

Обращение к системным переменным

Для обращения к системной переменной используйте метод `GetVariableValue` объекта Report:

```
DateTime date = (DateTime)Report.GetVariableValue("Date");
```

Список системных переменных можно увидеть в окне "Данные". Из него можно перетаскивать переменные в скрипт, при этом FastReport автоматически создает код для обращения к переменной.

Обращение к итоговым значениям

Для обращения к итоговому значению используйте метод `GetTotalValue` объекта Report:

```
float sales = Report.GetTotalValue("TotalSales");
```

Список итогов можно увидеть в окне "Данные". Из него можно перетаскивать итоги в скрипт, при этом FastReport автоматически создает код для обращения к итогу.

Итоговое значение имеет тип `FastReport.Variant`. Оно может быть напрямую использовано в любых выражениях, потому что тип `FastReport.Variant` автоматически приводится к любому типу. Например:

```
float tax = Report.GetTotalValue("TotalSales") * 0.2f;
```

Обращаться к итоговому значению можно в тот момент, когда оно вычислено. Обычно итог "готов к употреблению" в момент печати бэнда, на котором он располагается в отчете.

Обращение к параметрам отчета

Для обращения к параметру отчета используйте метод `GetParameterValue` объекта Report:

```
int myParam = (int)Report.GetParameterValue("MyParameter");
```

Параметры могут быть вложенными. В этом случае укажите имя родительского параметра и через точку имя дочернего параметра:

```
Report.GetParameterValue("ParentParameter.ChildParameter")
```

Параметры имеют определенный тип данных. Он задается в свойстве `DataType` параметра. Вы должны учитывать это при обращении к параметру. Список параметров можно увидеть в окне "Данные". Из него можно перетаскивать параметры в скрипт, при этом FastReport автоматически создает код для обращения к параметру.

Для изменения значения параметра используйте метод `SetParameterValue` объекта Report:

```
Report.SetParameterValue("MyParameter", 10);
```

Примеры использования

Пример 1. Изменение внешнего вида объекта

В этом примере мы покажем, как изменить цвет текста у объекта в зависимости от значения, которое печатается в объекте. Мы будем использовать:

- событие `BeforePrint`;
- обращение к полю БД из скрипта.

Создайте простой отчет следующего вида:

У объекта, который печатает стоимость продукта, создайте обработчик события `BeforePrint`:

```
private void Text2_BeforePrint(object sender, EventArgs e)
{
  if (((Decimal)Report.GetColumnValue("Products.UnitPrice")) > 20)
    Text2.TextColor = Color.Red;
}
```

Чтобы вставить в скрипт значение поля `Products.UnitPrice`, перетащите его из окна "Данные". При этом в скрипт будет вставлена строка:

```
((Decimal)Report.GetColumnValue("Products.UnitPrice"))
```

Если запустить отчет, мы увидим, что все продукты, имеющие стоимость > 20, выделены красным цветом:

> Такого же эффекта можно достичь с помощью условного выделения (подробнее об этом читайте в разделе "Условное выделение").

Пример 2. Выделение четных строк бэнда

В этом примере мы покажем, как изменить цвет заливки у четных строк бэнда "Данные". Мы будем использовать:

- событие `BeforePrint` бэнда;
- обращение к системной переменной `Row#` из скрипта.

Создайте простой отчет следующего вида:

У бэнда создайте обработчик события `BeforePrint`:

```
private void Data1_BeforePrint(object sender, EventArgs e)
{
  if (((Int32)Report.GetVariableValue("Row#")) % 2 == 0)
    Data1.FillColor = Color.Gainsboro;
}
```

Системная переменная `Row#` возвращает номер строки печатаемого бэнда. Чтобы вставить в скрипт обращение к переменной, перетащите его из окна "Данные". При этом в скрипт будет вставлена строка:

```
((Int32)Report.GetVariableValue("Row#"))
```

Если запустить отчет, мы увидим, что четные строки бэнда выделены серым цветом:

> Такого же эффекта можно достичь, используя свойство бэнда `EvenStyle`. Подробнее об этом читайте в разделе "Выделение строк данных через одну".

Пример 3. Фильтрация данных

В этом примере мы покажем, как спрятать строки бэнда "Данные" в зависимости от условия. Мы будем использовать:

- событие `BeforePrint` бэнда;
- обращение к полю БД из скрипта.

Создайте простой отчет следующего вида:

У бэнда создайте обработчик события `BeforePrint`:

```
private void Data1_BeforePrint(object sender, EventArgs e)
{
  if (((Decimal)Report.GetColumnValue("Products.UnitPrice")) > 20)
    Data1.Visible = false;
}
```

В данном случае будут спрятаны строки бэнда, для которых стоимость продукта > 20:

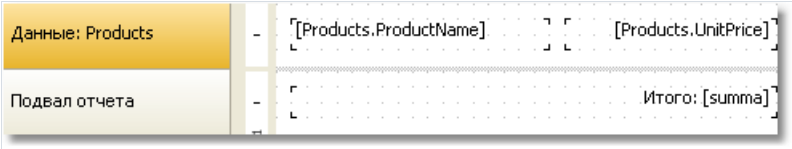
> Такого же эффекта можно достичь, используя условие фильтрации данных, которое можно настроить в редакторе бэнда "Данные".

Пример 4. Вычисление итогов

В этом примере мы покажем, как программным способом вычислить суммарное значение. Мы будем использовать:

- событие `BeforePrint` бэнда;
- обращение к полю БД из скрипта;
- локальную переменную, значение которой будет печататься в отчете.

Создайте отчет следующего вида:

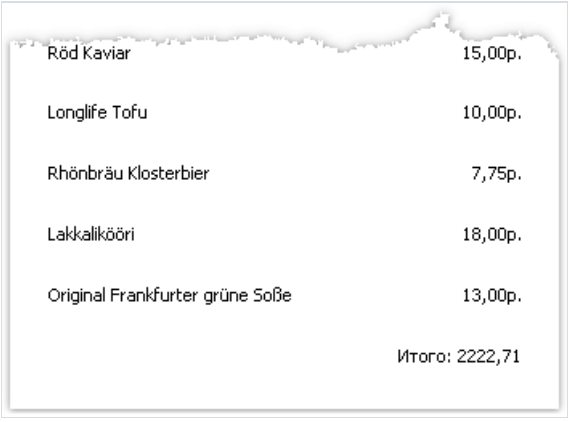


В скрипте объявите переменную `summa` и создайте у бэнда обработчик события `BeforePrint`:

```
public class ReportScript
{
  private decimal summa;

  private void Data1_BeforePrint(object sender, EventArgs e)
  {
    summa += (Decimal)Report.GetColumnValue("Products.UnitPrice");
  }
}
```

Поле таблицы `Products.UnitPrice` можно вставить в скрипт, перетащив его из окна "Данные".

Если запустить отчет, мы увидим следующее:



> Такого же эффекта можно достичь, используя итоги. Смотрите подробнее в главе "Данные".

Пример 5. Смещение позиции печати

В этом примере мы покажем, как можно смещать бэнды вручную, используя свойства объекта Engine. Мы будем использовать:

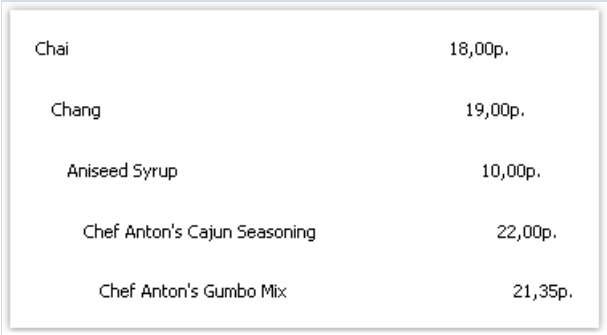
- событие `BeforePrint` бэнда;
- объект Engine.

Создайте простой отчет следующего вида:

У бэнда создайте обработчик события `BeforePrint`:

```
private void Data1_BeforePrint(object sender, EventArgs e)
{
  Engine.CurX = ((Int32)Report.GetVariableValue("Row#")) * 10;
}
```

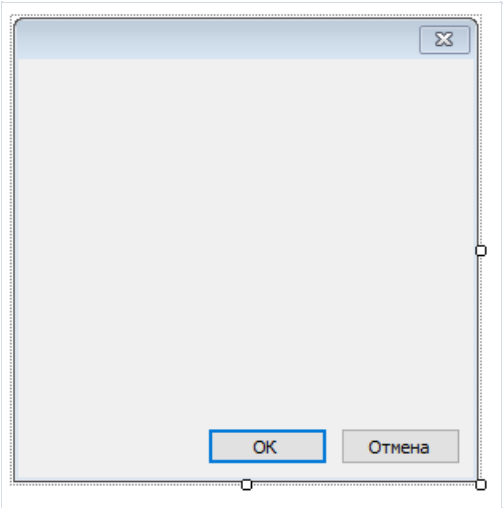
Запустите отчет на выполнение, и вы увидите следующее:



Диалоговые формы

Кроме обычных страниц, отчет может содержать диалоги. Диалог - это окно с элементами управления, которое показывается в момент запуска отчета. В диалоге можно ввести какую-либо информацию, необходимую для построения отчета. Также диалог удобно использовать для фильтрации данных, выводимых в отчете.

Для добавления диалога в отчет нажмите кнопку  на панели инструментов дизайнера. Новый диалог выглядит следующим образом:



Отчет, который содержит один или несколько диалогов, работает так:

- при запуске отчета показывается первый диалог;
- если диалог закрывается с помощью кнопки "OK", показывается следующий диалог;
- если диалог закрывается с помощью кнопки "Отмена" или крестиком на заголовке окна, работа отчета завершается;
- после того как показаны все диалоги, выполняется построение отчета.

Элементы управления

На диалоговой форме можно использовать следующие элементы управления:

Иконка	Название	Описание
	ButtonControl	Представляет собой кнопку.
	CheckBoxControl	Представляет собой флажок, имеющий два состояния (включен/выключен).
	CheckedListBoxControl	Представляет собой список строк с флажками.
	ComboBoxControl	Представляет собой выпадающий список.
	DataGridViewControl	Отображает данные в табличном виде.
	DataSelectorControl	Показывает два списка и позволяет переносить строки из одного списка в другой.
	DateTimePickerControl	Позволяет ввести дату или время.
	GroupBoxControl	Позволяет сгруппировать другие элементы управления.
	LabelControl	Представляет собой поясняющую надпись.
	ListBoxControl	Представляет собой список строк.
	ListViewControl	Представляет собой список строк с иконками.
	MaskedTextBoxControl	Позволяет ввести значение, используя маску ввода.
	MonthCalendarControl	Представляет собой календарь.
	NumericUpDownControl	Позволяет ввести числовое значение.
	PanelControl	Позволяет сгруппировать другие элементы управления.
	PictureBoxControl	Отображает рисунок.
	RadioButtonControl	Представляет собой зависимый переключатель, имеющий два состояния (включен/выключен).

Иконка	Название	Описание
	RichTextBoxControl	Отображает форматированный текст в формате RTF.
	TextBoxControl	Представляет собой поле для редактирования однострочного или многострочного текста.
	TreeViewControl	Показывает данные в иерархическом виде.

Все элементы управления, за исключением DataSelectorControl, являются полными аналогами стандартных элементов управления, доступных в .NET Framework. Названия элементов имеют приставку Control, чтобы избежать совпадения имен. Так, элементу управления ButtonControl соответствует элемент Button из .NET Framework.

Обращение к элементу из кода

Обратиться к элементу можно, используя его имя:

```
TextBoxControl1.Text = "my text";
```

По сути, элемент управления FastReport является оберткой над стандартным элементом управления .NET Framework. Он реализует многие, но не все, свойства стандартного элемента. Если реализованных свойств вам не достаточно, вы можете обратиться к стандартному элементу следующим образом:

- используя свойство `Control`, которое возвращает тип `System.Windows.Forms.Control`:

```
(TextBox1.Control as TextBox).ShortcutsEnabled = false;
```

- используя свойство, имя которого совпадает с именем элемента без приставки "Control". Например, для элемента TextBoxControl свойство `TextBox` возвращает стандартный элемент типа `System.Windows.Forms.TextBox`:

```
TextBox1.TextBox.ShortcutsEnabled = false;
```

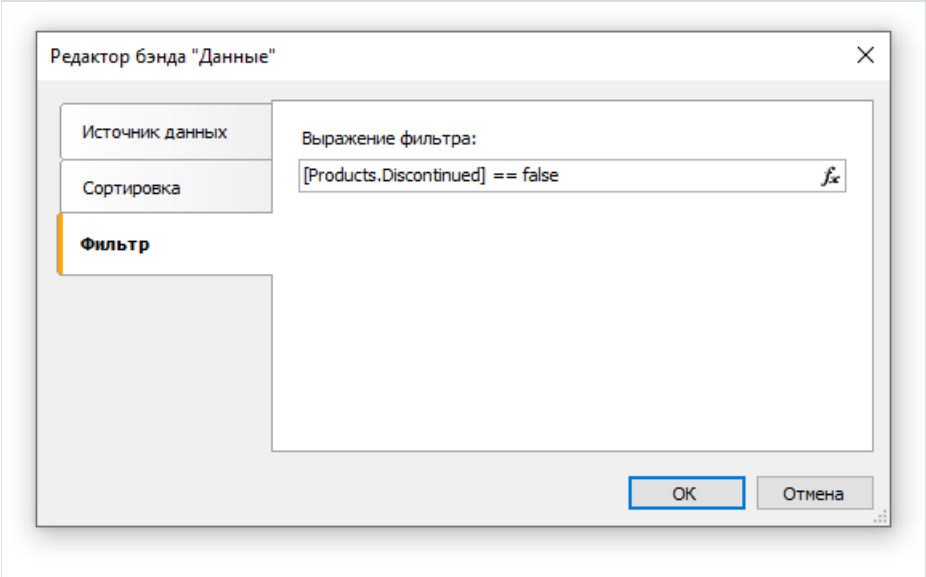
Справку по свойствам и методам элементов управления можно получить в MSDN.

Фильтрация данных

Один из способов использования диалогов - это фильтрация данных, выводимых в отчете. Например, имеется отчет, печатающий список всех сотрудников. Используя диалог, можно дать пользователю возможность выбора одного или нескольких из них, затем при построении отчета отфильтровать данные, чтобы показать выбранных сотрудников.

> Для использования фильтрации необходимо, чтобы исходный отчет печатал все данные. Само название "фильтрация" предполагает, что ненужные данные будут отброшены в процессе построения отчета.

Самый простой способ организовать фильтрацию данных - использовать свойство `Filter` бэнда "Данные". В редакторе бэнда можно указать условие фильтрации, например:



Используя диалог, можно запросить у пользователя значение и использовать его в выражении фильтра. Смотрите пример "Простой фильтр" в разделе "Примеры".

Этот способ можно использовать, если нужно запросить одно простое значение. Если же стоит задача показать пользователю список значений и запросить одно или несколько из них, реализовать это становится довольно тяжело. Казалось бы, простая задача - показать пользователю список сотрудников в элементе управления ListBoxControl и выбрать одно или несколько значений. Для реализации этого надо использовать скрипт, который выполнит следующее:

- получит источник данных по его имени;
- инициализирует данные;
- заполнит список ListBoxControl имеющимися в источнике значениями;
- после выбора сотрудников сформирует условие для фильтрации.

FastReport позволяет выполнить все эти действия автоматически, без написания какого-либо кода. Для этого используется автоматическая фильтрация, которую мы сейчас рассмотрим.

Автоматическая фильтрация - как это работает

Элемент управления подключается к полю данных с помощью свойства `DataColumn`. Если элемент может отображать список значений (например, ListBoxControl), он заполняется всеми значениями из указанного поля данных. Это происходит автоматически при запуске диалога. Далее пользователь работает с диалогом, выбирает одно или несколько значений в элементе и закрывает диалог. В этот момент к источнику данных, который указан в свойстве `DataColumn`, применяется фильтр, содержащий выбранные значения.

> Преимущество данного способа заключается в том, что его можно использовать в любом отчете, не написав для этого ни строки кода.

Автоматическая фильтрация поддерживается следующими элементами управления:

Иконка	Название
	CheckBoxControl
	CheckedListBoxControl
	ComboBoxControl
	DataSelectorControl
	DateTimePickerControl
	ListBoxControl
	MaskedTextBoxControl
	MonthCalendarControl
	NumericUpDownControl
	RadioButtonControl
	TextBoxControl

Операция фильтра

По умолчанию FastReport фильтрует строки данных, которые содержат значения, равные значению элемента управления. Это настраивается с помощью свойства `FilterOperation` элемента управления. В этом свойстве указывается операция которая будет использована при фильтрации данных. Вы можете использовать следующие операции:

Операция	Эквивалент	Действие
Equal	=	Выбрать значения, равные значению элемента управления.
NotEqual	<>	Выбрать значения, не равные значению элемента.
LessThan	<	Выбрать значения, меньшие чем значение элемента.
LessThanOrEqual	<=	Выбрать значения, меньшие чем или равные значению элемента.
GreaterThan	>	Выбрать значения, бОльшие чем значение элемента.
GreaterThanOrEqual	>=	Выбрать значения, бОльшие чем или равные значению элемента.

Например, если свойство `FilterOperation` элемента управления установлено в `LessThanOrEqual` и вы ввели в элемент значение `5`, то будут выбраны все строки данных, для которых значение фильтруемого поля `меньше или равно 5`.

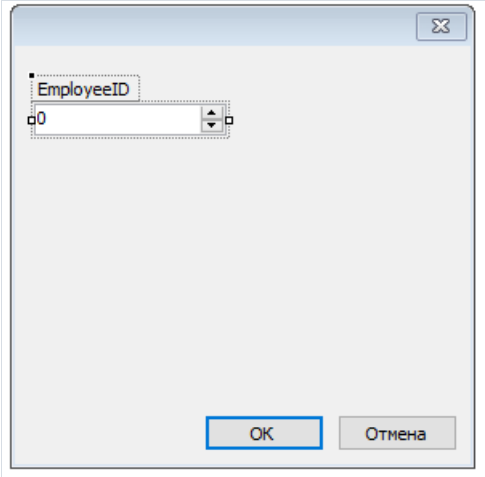
Для данных строкового типа можно использовать дополнительные операции, которые проверяют вхождение одной строки в другую:

Операция	Действие
Contains	Выбрать строки, содержащие значение из элемента управления.
NotContains	Выбрать строки, не содержащие значение из элемента.
StartsWith	Выбрать строки, начинающиеся со значения из элемента.
NotStartsWith	Выбрать строки, не начинающиеся со значения из элемента.
EndsWith	Выбрать строки, заканчивающиеся значением из элемента.
NotEndsWith	Выбрать строки, не заканчивающиеся значением из элемента.

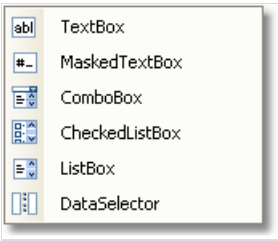
Например, если свойство `FilterOperation` элемента управления установлено в `StartsWith` и вы ввели в элемент значение `A`, то будут выбраны все строки данных, для которых значение фильтруемого поля начинается на `A`.

Добавление фильтра в отчет

Дизайнер диалогов FastReport имеет удобные средства для добавления элементов управления, которые поддерживают фильтрацию данных, на диалог. Для этого перетащите (drag&drop) поле данных, которое вас интересует, из окна "Данные" на диалоговую форму. При этом FastReport создаст заголовок (элемент управления LabelControl) и собственно элемент, фильтрующий данные:



Тип элемента в данном случае зависит от того, какой тип имеет поле данных. Если поле строкового типа, то после его вставки будет предложено выбрать тип элемента:



Если перетащить на диалог два одинаковых элемента, связанные с одним и тем же полем данных, FastReport автоматически настроит диапазон данных с помощью свойства `FilterOperation`. У первого добавленного элемента установится `FilterOperation = GreaterThanOrEqual`, у второго - `LessThanOrEqual`. Это будет сделано в случае, если вы добавляете поле не строкового типа.

Таким образом, для добавления фильтрации в любой отчет вам нужно сделать следующее:

- добавить в отчет новый диалог;
- перетащить на диалог поле данных, по которому вы хотите отфильтровать отчет.

Фильтрация по диапазону значений

Этот способ фильтрации удобно использовать при работе со значениями, имеющими количественную характеристику, например, стоимость. Таким образом можно отфильтровать товар, имеющий стоимость ниже или выше заданной. Для того чтобы указать, как следует трактовать введенное в элемент значение, используется свойство `FilterOperation`, рассмотренное выше.

Используя два элемента управления, которые подключены к одному и тому же полю данных и имеют разные настройки свойства `FilterOperation`, можно указать начало и конец диапазона данных. Для первого элемента надо указать свойство `FilterOperation = GreaterThanOrEqual`, для второго - `LessThanOrEqual`.

Фильтрация по связанному полю

Как нам известно, между двумя источниками данных можно установить связь. Подробнее об этом читайте в главе "Данные". С помощью связи можно фильтровать данные в источнике, используя для этого поле из другого, связанного источника.

Допустим, вы положили на диалог элемент управления ListBoxControl и указали в свойстве `DataColumn` следующее поле данных:

```
Products.Categories.CategoryName
```

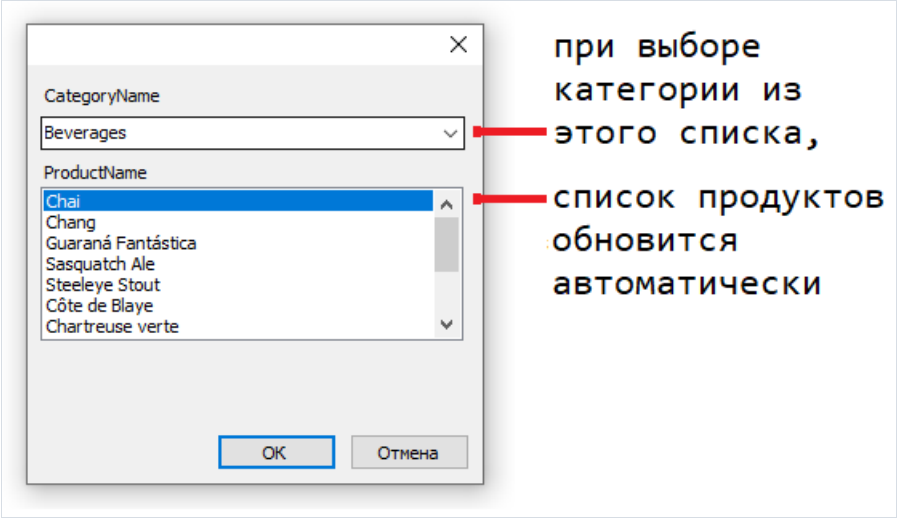
Как будет работать фильтрация?

- при заполнении элемента управления будет использовано поле `CategoryName` источника данных Categories;
- при фильтрации данных будут отфильтрованы те строки источника данных Products, для которых верно условие:

```
поле [Products.Categories.CategoryName] содержит одно из выбранных пользователем значений.
```

Фильтрация с каскадными списками

Каскадный список - это список, набор значений в котором определяется выбором, сделанным пользователем в другом списке. Допустим, на форме есть два списка - список категорий и список продуктов. При выборе категории из первого списка, во втором списке будут отображены только те продукты, которые входят в выбранную категорию.



Для создания каскадного списка нужно два источника данных, связанных отношением типа "главный-подчиненный" (подробнее об источниках и связях читайте в главе "Данные"). Первый список (главный) прикрепляется к полю из главного источника данных, второй список (подчиненный) - к полю подчиненного источника. Кроме того, надо настроить свойство `DetailControl` главного списка, указав в нем подчиненный список.

> Каскадную фильтрацию удобно применять к отчету типа master-detail. Если отчет другого типа (например, с одним бэндом "Данные"), для корректной работы фильтрации нужно будет использовать скрипт.

Управление фильтрацией из кода

Хотя возможностей автоматической фильтрации достаточно для большинства случаев, у вас есть возможность управлять фильтрацией вручную. Для этого используются следующие свойства и методы.

Свойство `AutoFill` управляет заполнением элемента. Оно используется элементами, которые могут показывать список данных, например, ListBoxControl. Перед тем как показать диалог, FastReport заполняет данными такие элементы. По умолчанию свойство равно `true`. Если его отключить, элемент заполняться не будет, и вы должны сделать это сами, вызвав метод `FillData`:

```
ListBox1.FillData();
```

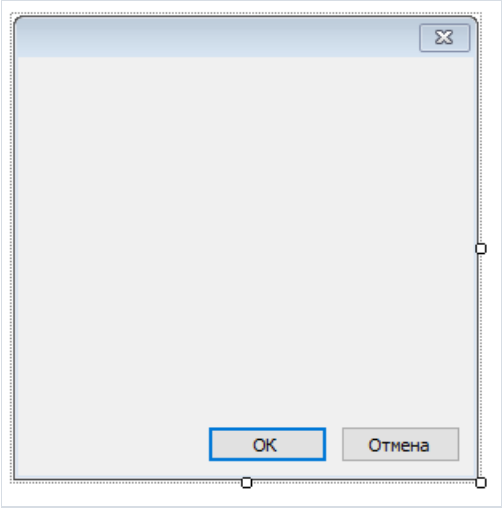
Свойство `AutoFilter` управляет фильтрацией данных. Оно используется всеми элементами управления. После того как диалог закрыт кнопкой "OK", FastReport выполняет фильтрацию данных автоматически. По умолчанию свойство равно `true`. Если его отключить, фильтрация выполняться не будет, и вы должны сделать это сами, вызвав метод `FilterData`:

```
ListBox1.FilterData();
```

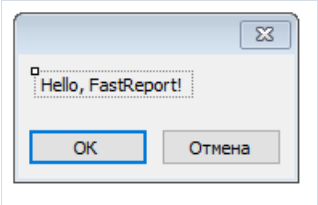
Примеры использования

Пример 1. Диалог "Hello, FastReport!"

В этом примере все, что мы сделаем - создадим диалог, который будет показывать приветствие. Создайте новый отчет и добавьте в него диалог. Для этого нажмите кнопку  на панели инструментов:



На диалог положите элемент управления LabelControl и настройте его свойство `Text` в окне "Свойства":

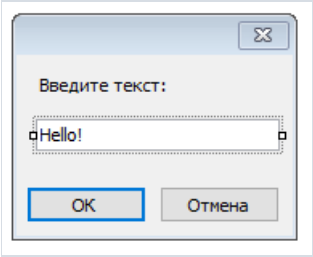


Если запустить отчет, вы увидите диалог. Закройте его кнопкой "OK", и отчет будет построен. Если закрыть диалог кнопкой "Отмена" или крестиком на заголовке, отчет прекратит работу, и вы вернетесь в дизайнер.

Пример 2. Запрос строки у пользователя

В этом примере мы создадим диалог, который будет запрашивать произвольную строку текста у пользователя, а затем печатать введенное значение в отчете.

Создайте новый отчет и добавьте в него диалог. На диалог положите элементы управления LabelControl и TextBoxControl:



В данном случае введенное нами значение содержится в свойстве `Text` элемента управления TextBoxControl. Чтобы напечатать это значение в отчете, добавьте новый объект "Текст" на бэнд "Заголовок отчета" и напишите в нем следующее:

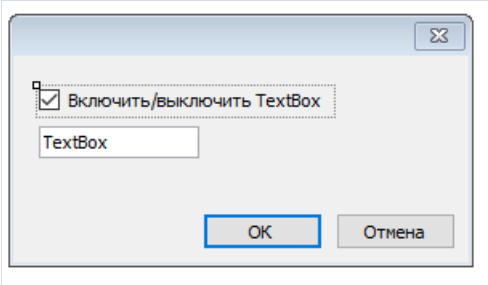
```
Вы ввели: [TextBox1.Text]
```

где TextBox1 - имя элемента управления TextBoxControl.

Пример 3. Управление элементами диалога

Используя скрипт и события элементов управления, можно управлять элементами так же, как это делается в Visual Studio. Покажем на примере, как с помощью элемента CheckBoxControl управлять доступностью элемента TextBoxControl.

Создайте новый отчет и добавьте в него диалог. На диалог положите элементы CheckBoxControl и TextBoxControl, как показано на рисунке:



Теперь выберите элемент CheckBoxControl, откройте окно "Свойства" и нажмите кнопку  . Сделайте двойной щелчок на событии `CheckedChanged`, которое возникает при изменении состояния флажка. FastReport создаст пустой обработчик этого события. Напишите в нем следующий код:

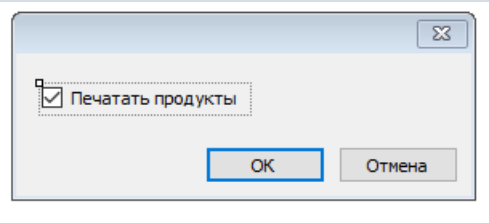
```
private void CheckBox1_CheckedChanged(object sender, EventArgs e)
{
  TextBox1.Enabled = CheckBox1.Checked;
}
```

Если запустить отчет, мы увидим, что элемент TextBoxControl реагирует на состояние флажка.

Пример 4. Управление объектами отчета

Рассмотрим пример отчета, который печатает список категорий и продуктов в каждой категории:

Покажем, как с помощью диалога запретить печать продуктов и напечатать только категории. Для этого добавьте в отчет диалог:



Сделайте двойной щелчок на кнопке "OK". FastReport создаст пустой обработчик события `Click` кнопки. Напишите в нем следующий код:

```
private void btnOk_Click(object sender, EventArgs e)
{
  Data2.Visible = CheckBox1.Checked;
}
```

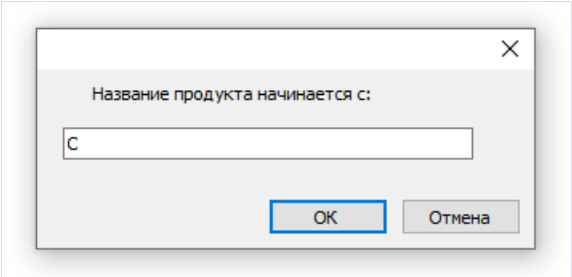
Мы будем управлять видимостью бэнда, который печатает список продуктов. В нашем примере это бэнд с именем Data2. Если запустить отчет и выключить флажок, мы увидим следующий результат:


Beverages
Soft drinks, coffees, teas, beers, and ales
Condiments
Sweet and savory sauces, relishes, spreads, and seasonings
Confections
Desserts, candies, and sweet breads

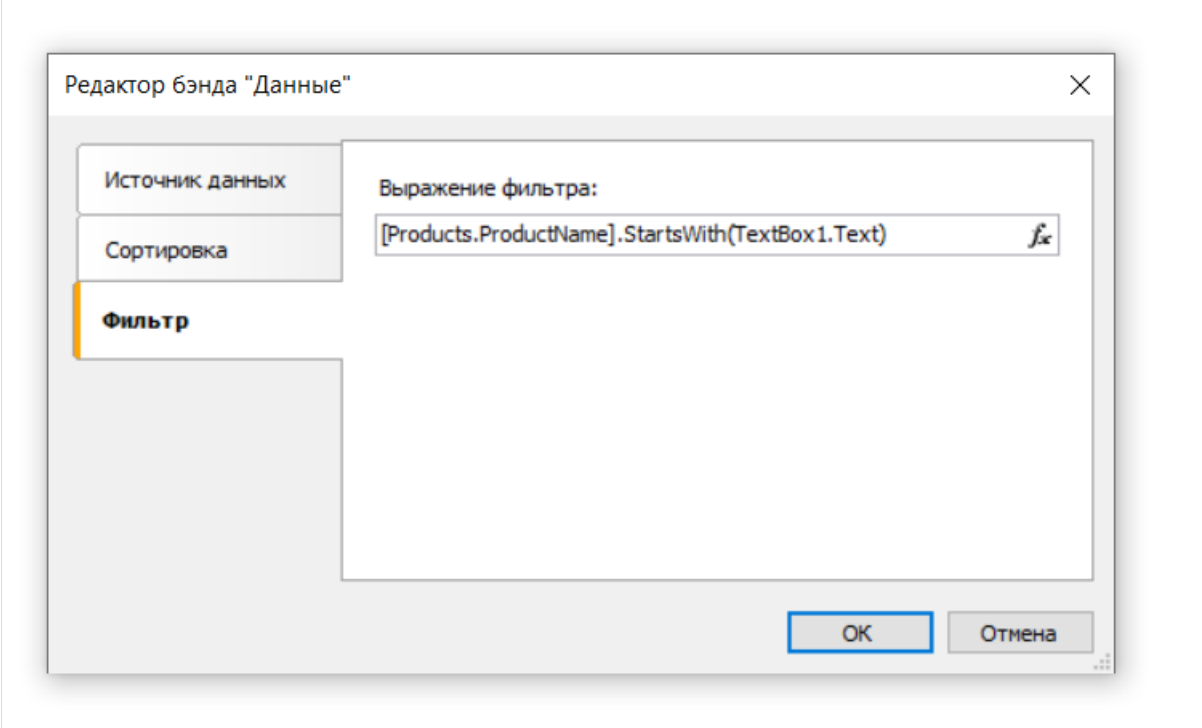
Пример 5. Простой фильтр

Рассмотрим следующий отчет, который печатает список продуктов:

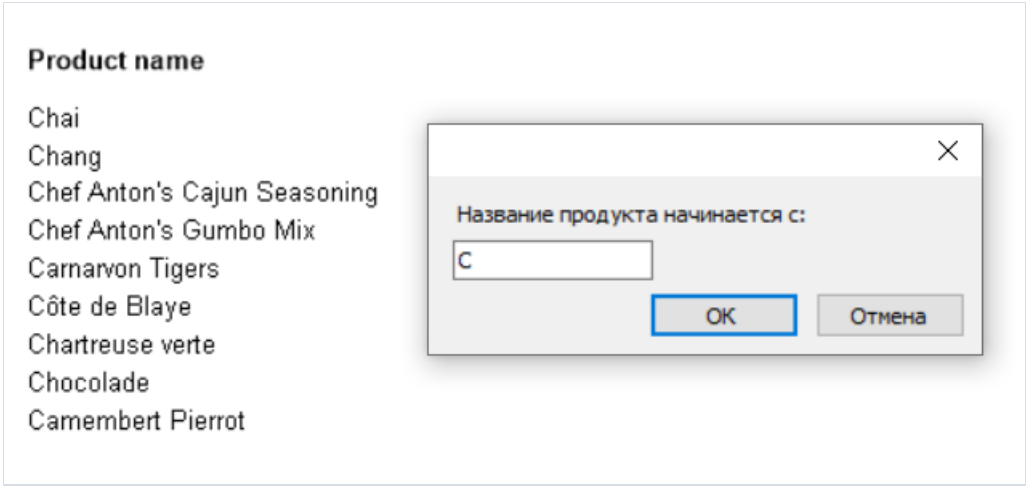
Покажем в этом примере, как отфильтровать список продуктов по первой букве названия продукта. При этом мы не будем использовать средства автоматической фильтрации данных. Для этого добавьте в отчет новый диалог и положите на него два элемента управления - LabelControl и TextBoxControl:



Теперь откроем редактор бэнда "Данные" и укажем следующее условие фильтрации:



После чего запустим отчет и убедимся, что все работает:

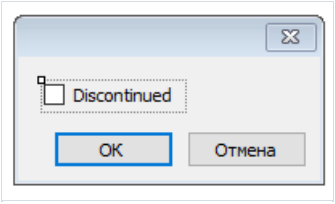

Product name
Chai
Chang
Chef Anton's Cajun Seasoning
Chef Anton's Gumbo Mix
Carnarvon Tigers
Côte de Blaye
Chartreuse verte
Chocolade
Camembert Pierrot
Название продукта начинается с:
C
OK
Отмена

Пример 6. Автоматическая фильтрация

В этом примере покажем, как добавить фильтр в отчет, печатающий список продуктов из таблицы Products. Фильтрация будет выполняться по полю `Products.Discontinued`.

Сам отчет имеет следующий вид:

Добавьте в отчет новый диалог, нажав кнопку  на панели инструментов, и перетащите на него поле `Products.Discontinued` из окна "Данные":



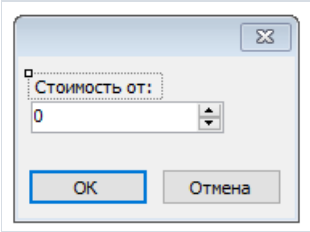
Это все, что нам нужно сделать - мы уложились буквально в 2 клика мыши. FastReport автоматически привязал элемент управления к полю данных.

Запустите отчет и включите флажок `Discontinued`. После этого нажмите кнопку "OK", и вы увидите отчет, который содержит только продукты с флагом `Discontinued`:

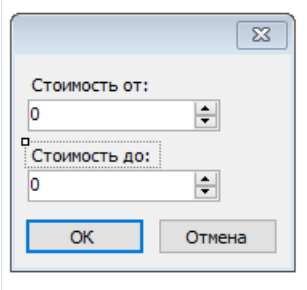
Product name	Discontinued
Alice Mutton	✓
Chef Anton's Gumbo Mix	✓
Guaraná Fantástica	✓
Mishi Kobe Niku	✓
Perth Pasties	✓
Rössle Sauerkraut	✓
Singaporean Hokkien Fried Mee	✓
Thüringer Rostbratwurst	✓

Пример 7. Автоматическая фильтрация по диапазону

Покажем на отчете из предыдущего примера, как напечатать продукты, имеющие стоимость в указанном диапазоне. Для этого добавим в отчет диалог и перетащим на него поле `Products.UnitPrice`. После этого поправим текст элемента:



Теперь аналогичным образом добавьте еще одно поле `Products.UnitPrice` и поправьте его заголовок:



Это все, что нам нужно сделать. Всю остальную работу сделал FastReport: привязал элементы к полю данных и настроил их свойства `FilterOperation`. Первый элемент имеет `FilterOperation = GreaterThanOrEqual`, второй - `LessThanOrEqual`.

Запустите отчет и укажите значения, например, от 20 до 30. При нажатии на кнопку "OK" будет построен отчет. Он содержит продукты, имеющие стоимость в указанном нами диапазоне:

Product name	UnitPrice
Chef Anton's Cajun Seasoning	22,00р.
Chef Anton's Gumbo Mix	21,35р.
Fløtemysost	21,50р.
Grandma's Boysenberry Spread	25,00р.
Gravad lax	26,00р.
Gustaf's Knäckebröd	21,00р.
Louisiana Fiery Hot Pepper Sauce	21,05р.
Maxilaku	20,00р.
Nord-Ost Matjeshering	25,89р.
Pâté chinois	24,00р.
Queso Cabrales	21,00р.
Sirop d'érable	28,50р.
Tofu	23,25р.
Uncle Bob's Organic Dried Pears	30,00р.

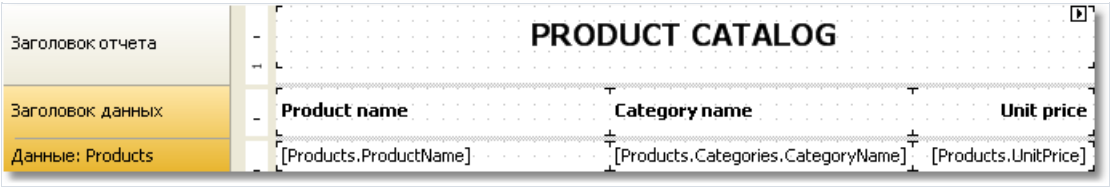
Пример 8. Фильтрация по полю связанного источника данных

В этом примере мы будем использовать для фильтрации источника данных поле, которое содержится в связанном источнике.

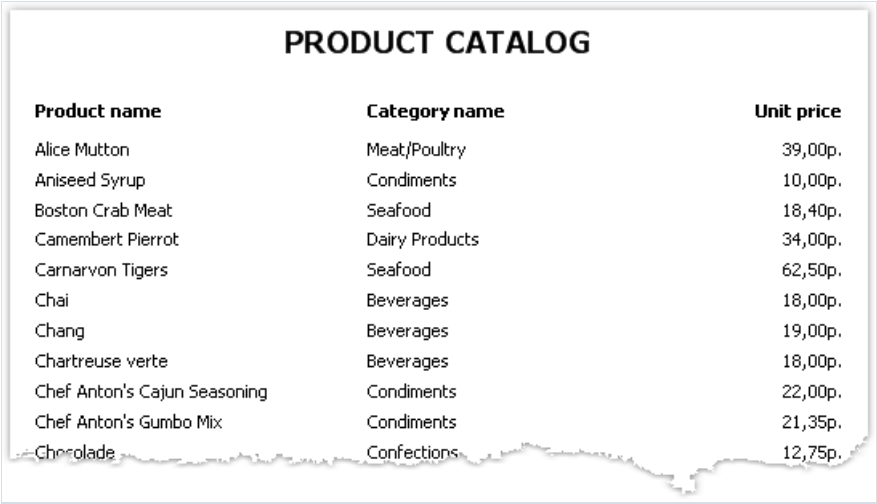
Рассмотрим отчет типа "Простой список", который печатает список продуктов. Около каждого продукта печатается название категории, в которую он входит. Это делается с помощью связи:

```
[Products.Categories.CategoryName]
```

Отчет выглядит следующим образом:



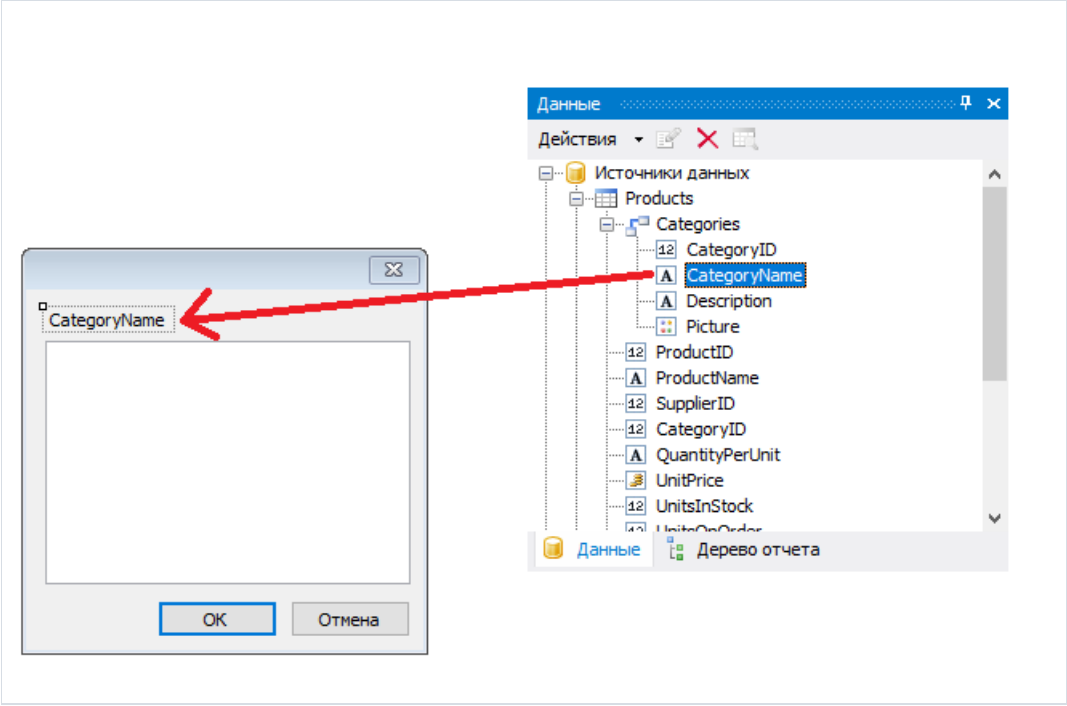
При запуске отчета мы увидим следующее:



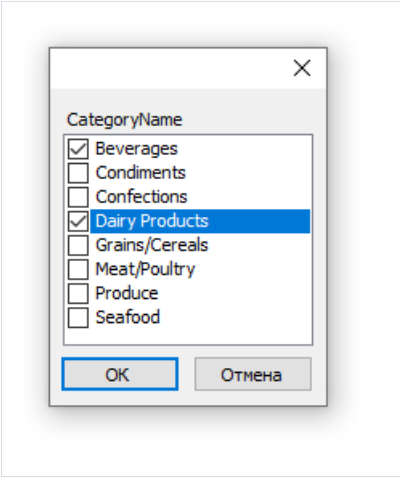
PRODUCT CATALOG

Product name	Category name	Unit price
Alice Mutton	Meat/Poultry	39,00р.
Aniseed Syrup	Condiments	10,00р.
Boston Crab Meat	Seafood	18,40р.
Camembert Pierrot	Dairy Products	34,00р.
Carnarvon Tigers	Seafood	62,50р.
Chai	Beverages	18,00р.
Chang	Beverages	19,00р.
Chartreuse verte	Beverages	18,00р.
Chef Anton's Cajun Seasoning	Condiments	22,00р.
Chef Anton's Gumbo Mix	Condiments	21,35р.
Chocolade	Confections	12,75р.

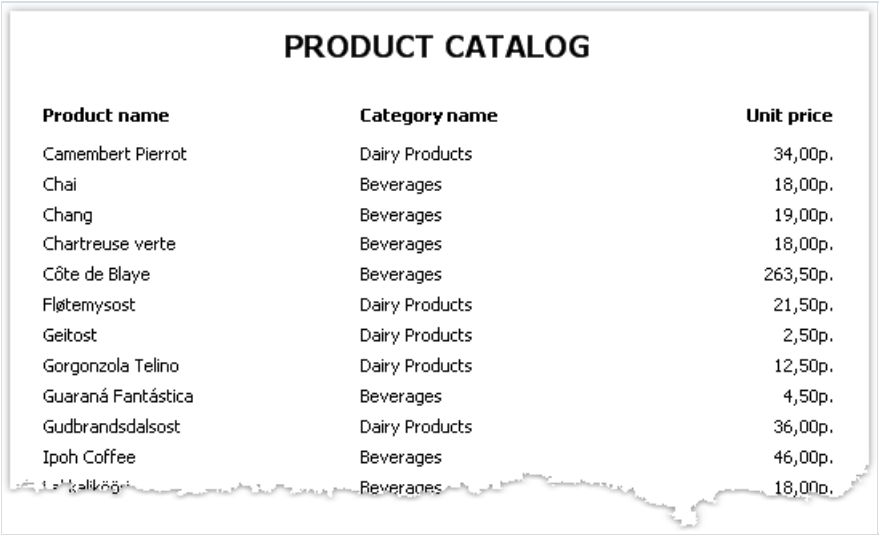
Давайте добавим фильтрацию по названию категории. Для этого добавьте новый диалог и перетащите на него поле `Products.Categories.CategoryName`:



При создании элемента управления будет предложено выбрать его тип. Выберите CheckedListBoxControl. Если запустить отчет, мы увидим следующий диалог:



Выберите несколько элементов и нажмите кнопку "OK". После этого данные будут отфильтрованы и вы увидите следующий отчет:



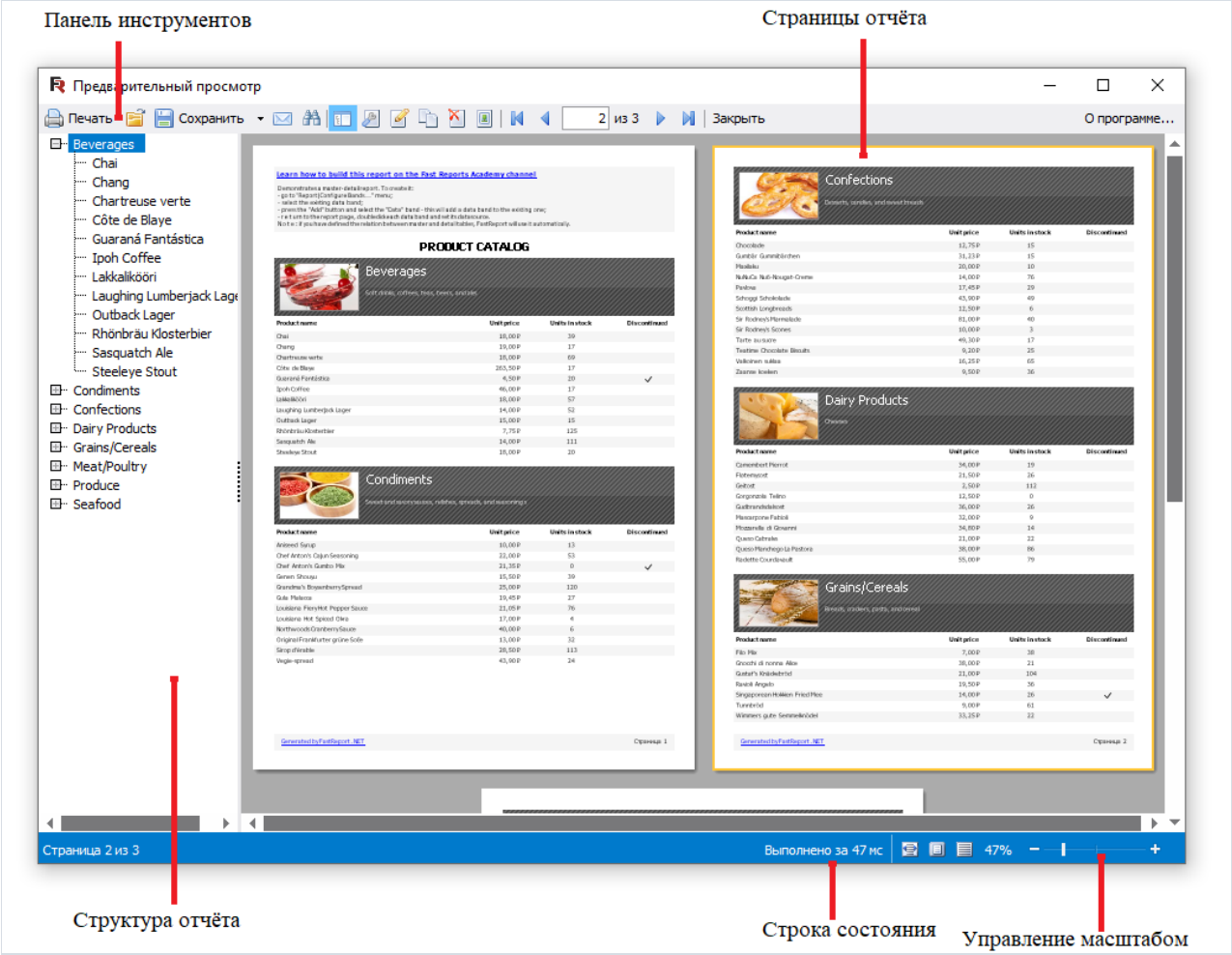
PRODUCT CATALOG

Product name	Category name	Unit price
Camembert Pierrot	Dairy Products	34,00р.
Chai	Beverages	18,00р.
Chang	Beverages	19,00р.
Chartreuse verte	Beverages	18,00р.
Côte de Blaye	Beverages	263,50р.
Fløtemysost	Dairy Products	21,50р.
Geitost	Dairy Products	2,50р.
Gorgonzola Telino	Dairy Products	12,50р.
Guaraná Fantástica	Beverages	4,50р.
Gudbrandsdalsost	Dairy Products	36,00р.
Ipoh Coffee	Beverages	46,00р.
Lakkalikööri	Beverages	18,00р.

Как видно, остались только продукты, которые входят в выбранные категории.

Просмотр, печать, экспорт

Построенный отчет можно показать на экране, распечатать на принтере или экспортировать в один из поддерживаемых форматов. Все это можно сделать в окне предварительного просмотра:



На панели инструментов имеются следующие кнопки:

Кнопка	Описание
	Печать отчета.
	Открыть файл готового отчета в формате FPX.
	Сохранить отчет в одном из поддерживаемых форматов.
	Отправить отчет по email.
	Поиск текста в отчете.

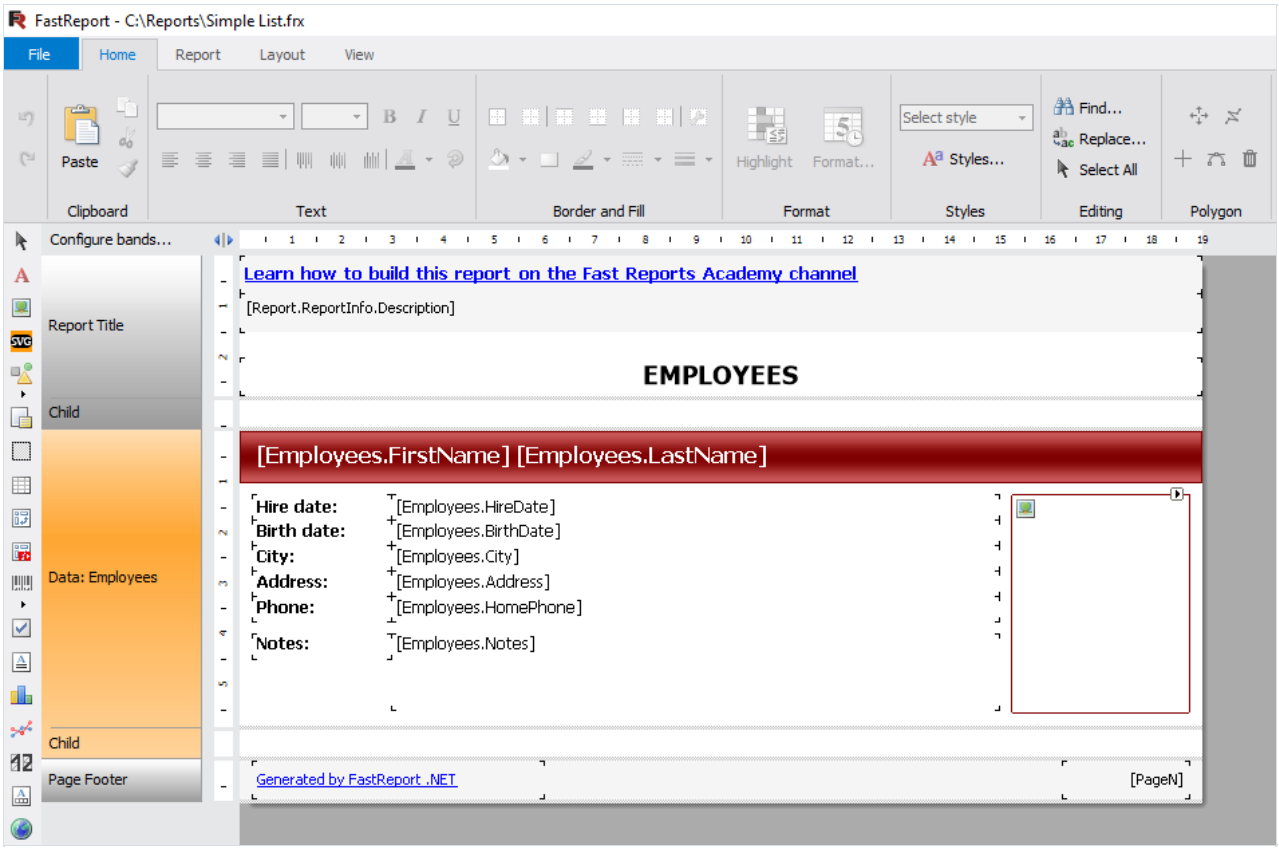
Кнопка	Описание
	Показывает или скрывает структуру отчета.
	Настройка параметров страницы.
	Редактирование текущей страницы отчета.
	Настройка параметров водяного знака.
	Переход на первую страницу отчета.
	Переход на предыдущую страницу отчета.
	Переход на указанную страницу отчета. Введите номер и нажмите `Enter`.
	Переход на следующую страницу отчета.
	Переход на последнюю страницу отчета.

Вы можете использовать следующие клавиши управления:

Клавиша	Описание
`Ctrl+P`	Печатать отчет.
`Ctrl+F`	Поиск текста.
Стрелки	Плавная прокрутка.
`PageUp, PageDown`	Листание вверх/вниз.
`Home`	Переход на первую страницу.
`End`	Переход на последнюю страницу.
`Esc`	Закрыть окно просмотра.

Редактирование отчета

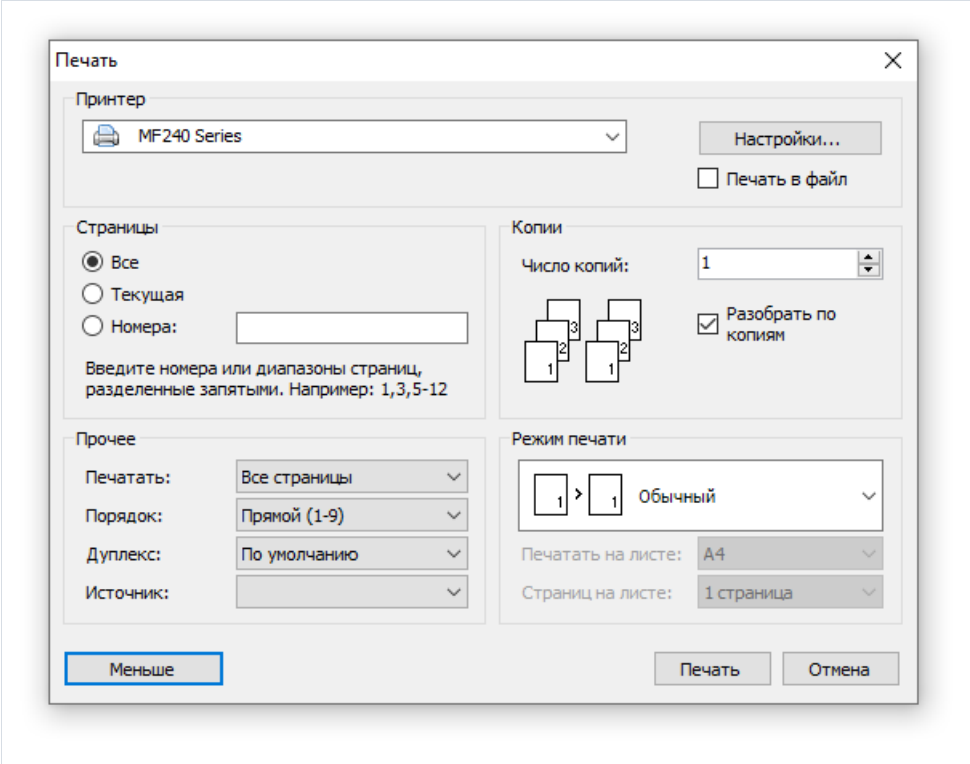
Для редактирования страницы готового отчета нажмите кнопку  в окне предварительного просмотра. При этом текущая страница будет загружена в дизайнер отчета, где вы можете сделать с ней все, что захотите:



После редактирования закройте дизайнер, при этом будет предложено сохранить изменения в странице отчета.

Печать отчета

Для того, чтобы распечатать отчет на принтере, нажмите кнопку (или комбинацию клавиш `Ctrl+P`). Появится окно – диалог печати:



Рассмотрим настройки, доступные в этом диалоговом окне:

- кнопка "Больше/Меньше": позволяет показывать весь диалог или только базовые настройки. По умолчанию диалог показывается в упрощенном виде;
- группа "Принтер": здесь можно выбрать принтер, задать его настройки (кнопка "Настройки") и выбрать печать в файл;
- группа "Страницы": здесь можно выбрать, какие страницы печатать (все, текущую или заданные номера страниц);
- группа "Копии": здесь можно задать количество копий и выбрать порядок страниц в копиях ("Разобрать по копиям");
- группа "Прочее": здесь можно выбрать, какие страницы из выбранного диапазона печатать (все, четные, нечетные), выбрать порядок печати (прямой, обратный), задать настройки для двусторонней печати ("Дуплекс" - если ваш принтер его поддерживает) и выбрать источник бумаги (лоток принтера).
- группа "Режим печати" позволяет выбрать один из режимов печати:

Режим	Описание
Обычный	Принтер печатает на формате бумаги, указанной в отчете. Одной странице отчета соответствует один лист распечатки.
Разрезать большие страницы	Этот режим используется, если необходимо распечатать отчет формата А3 на бумаге А4. При этом из одной страницы отчета получается два печатных листа. При использовании этого режима надо выбрать формат бумаги, на котором вы хотите печатать, из списка "Печатать на листе".

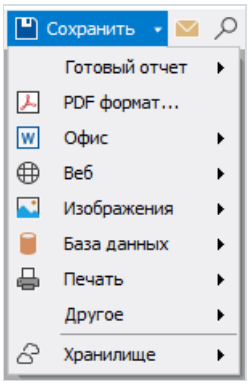
Режим	Описание
Масштабирование	Этот режим используется, если необходимо напечатать отчет А4 на формате А3. На одном печатном листе можно напечатать 1, 2, 4 или 8 страниц отчета. При использовании этого режима надо выбрать формат бумаги, на котором вы хотите печатать, из списка "Печатать на листе", а также указать количество страниц в списке "Страниц на листе".

После нажатия на кнопку "Печать" начинается печать отчета. Если выбран флажок "Печать в файл", то будет запрошено имя файла и отчет будет сохранен в этот файл (файл с расширением PRN, содержит копию информации, отправляемой на принтер).

Экспорт отчета

FastReport позволяет осуществлять экспорт построенного отчета в различные форматы для последующего редактирования, архивирования, пересылки по электронной почте и других целей.

Для выбора экспорта нажмите кнопку "Сохранить" в окне предварительного просмотра и выберите формат экспорта:



На данный момент поддерживается экспорт в следующие форматы:

- PDF;
- Офис:
  - Excel 2007;
  - Word 2007;
  - PowerPoint 2007;
  - OpenOffice Calc / OpenOffice Writer;
  - RTF;
  - Excel XML (Excel 2003+).
- Веб:
  - HTML;
  - MHT.
- Изображения:
  - Bmp, Png, Jpeg, Gif, Tiff, Metafile;
  - SVG.
- База данных:
  - CSV;
  - DBF;
  - Json.
- Печать:
  - TXT;
  - ZPL;
  - PPML;
  - PostScript;
  - XPS.
- Другое:
  - LaTeX;
  - DXF;

- XAML.

Также возможно сохранение отчета в облачных сервисах, таких как Dropbox, Box, Google Drive, OneDrive, Simple Storage Service, а также на FTP-сервере.

Сохранение в формате FPX

Формат .FPX - это "родной" формат FastReport. Преимущества этого формата заключаются в следующем:

- сохранение отчета без потери качества. Открыв сохраненный ранее файл, вы можете выполнять над ним все операции, такие как печать, экспорт, редактирование;
- компактный формат хранения - XML, сжатый с помощью ZIP;
- при необходимости файл отчета можно распаковать любым архиватором, поддерживающим ZIP-формат и поправить вручную в любом текстовом редакторе.

Недостаток формата заключается только в том, что для его просмотра необходимо наличие FastReport .NET.

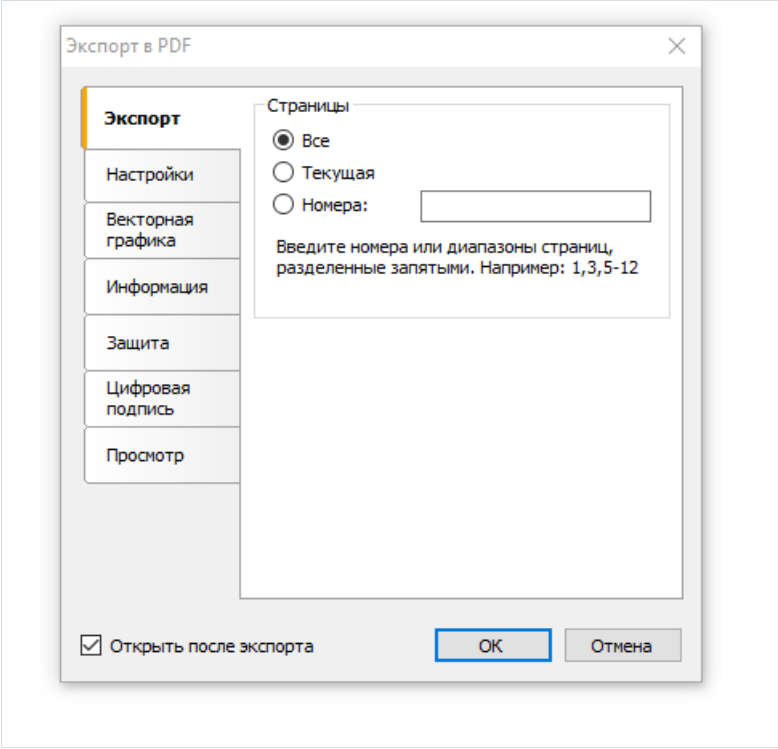
Для сохранения в формате FPX нажмите кнопку "Сохранить" в окне предварительного просмотра и выберите тип файла "Готовый отчет". Для открытия ранее сохраненного файла нажмите кнопку "Открыть".

Экспорт в Adobe Acrobat (PDF)

PDF (Portable Document Format) – платформо-независимый формат электронных документов, созданный фирмой Adobe Systems. Для просмотра используется бесплатный пакет Acrobat Reader. Данный формат достаточно гибкий - позволяет внедрять необходимые шрифты, векторные и растровые изображения, очень хорошо подходит для передачи и хранения документов, предназначенных для просмотра и последующей печати.

Метод экспорта – послойный.

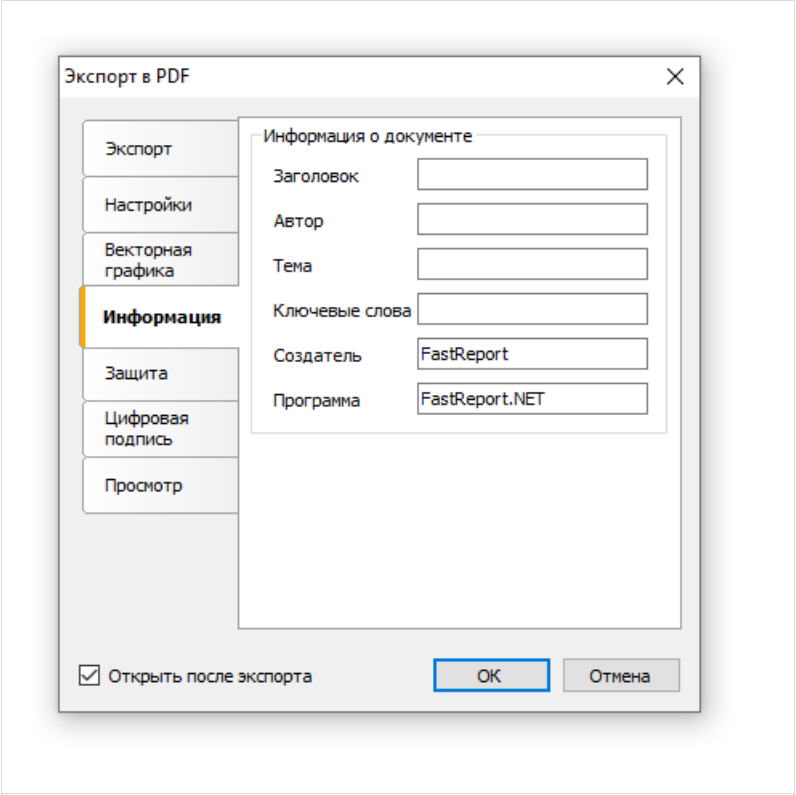
При экспорте в формат PDF будет предложено диалоговое окно для настройки параметров выходного файла:



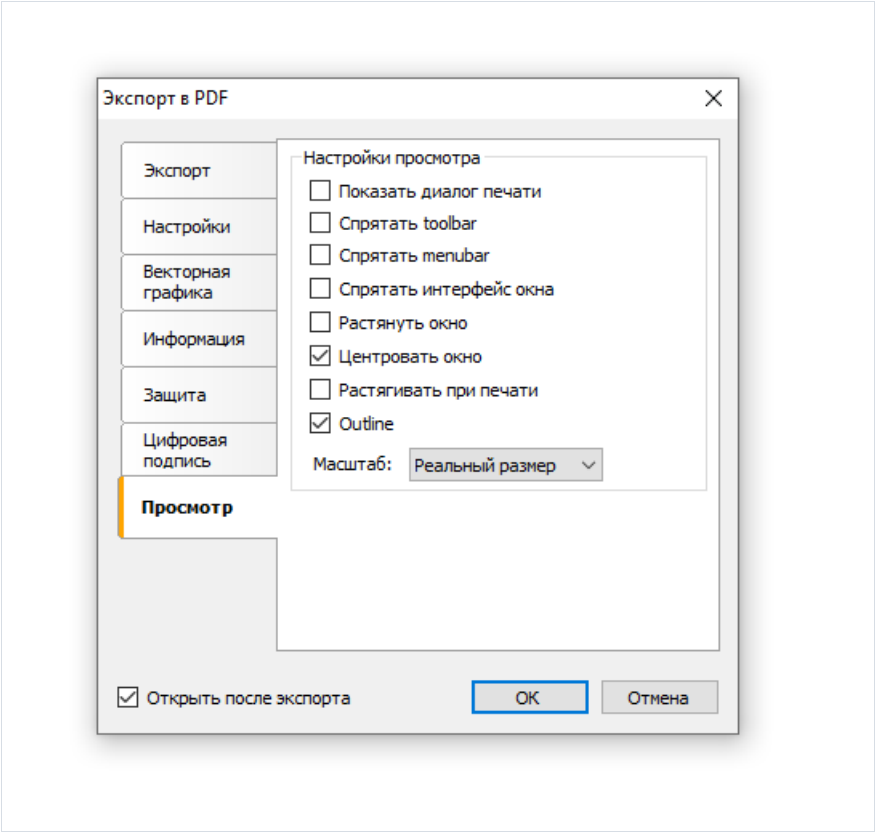
Параметры экспорта:

- Сжатый – сжатие выходного файла, уменьшает размер файла, но увеличивает время экспорта;
- Со шрифтами – все шрифты, использованные в отчете, будут также помещены в выходной файл PDF для корректного отображения файла на компьютере, где этих шрифтов может не быть. Размер выходного файла значительно увеличивается;
- Фон – экспорт водяного знака страницы в файл PDF, значительно увеличивает размер выходного файла;
- Для печати – вывод графических изображений в высоком разрешении для последующей печати на принтере. Включение этой опции нужно только в том случае, если документ содержит графику и будет необходима его печать, значительно увеличивает размер выходного файла;
- Открыть после экспорта – результирующий файл будет открыт сразу же после экспорта в Adobe Acrobat Reader.

На закладке "Информация" можно заполнить поля - информацию о документе:



На закладке "Просмотр" можно задать некоторые опции, влияющие на просмотр документа в Adobe Acrobat Reader:

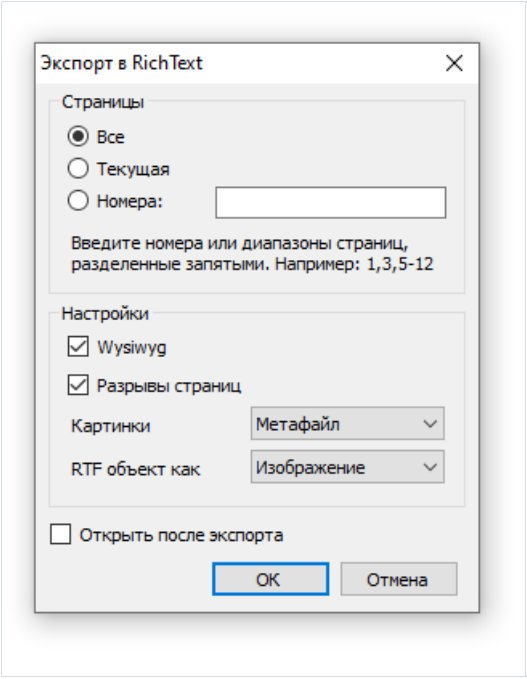


Экспорт в RTF

RTF (Rich Text Format) был разработан фирмой Microsoft как стандартный формат для обмена текстовыми документами. На данный момент RTF-документы совместимы с большинством современных текстовых редакторов и операционных систем.

Метод экспорта – табличный.

При экспорте в формат RTF будет предложено диалоговое окно для настройки параметров выходного файла:



Параметры экспорта:

- Wysiwyg – максимальное соответствие внешнему виду отчета, при отключении этой опции будет производиться оптимизация по уменьшению количества строк и столбцов в результирующей таблице;
- Разрывы страниц – включает разрыв страниц в RTF файле;
- Картинки – выбор формата графических изображений в результирующем файле. По умолчанию предлагается формат метафайла (.EMF), что позволяет экспортировать такие объекты, как "Диаграмма" и "Фигура" с максимальным качеством;
- Открыть после экспорта – результирующий файл будет открыт сразу же после экспорта программой просмотра RTF файлов, назначенной в операционной системе по умолчанию (к примеру, MS Word).

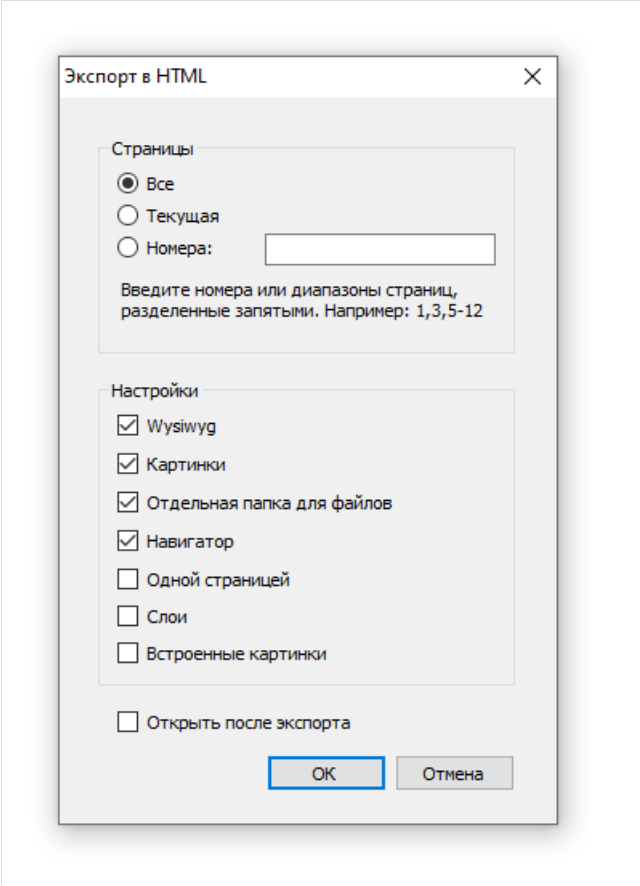
> Внешний вид и объем файла сильно зависят от шаблона отчета (см. раздел "Рекомендации по разработке отчетов").

Экспорт в HTML

HTML (Hypertext Markup Language) – считается стандартным языком для разметки документов в Internet. HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области верстки. Служит для создания относительно простых, но красиво оформленных документов. Помимо упрощения структуры документа, в HTML внесена поддержка гипертекста.

Метод экспорта – табличный.

При экспорте в HTML будет предложено диалоговое окно для настройки параметров выходного документа:



Параметры экспорта:

- Wysiwyg - результат экспорта будет выглядеть максимально приближенно к отчету;
- Картинки – включает возможность экспорта графических изображений;
- Отдельная папка для файлов – все дополнительные файлы сохраняются в отдельной папке с названием .files;
- Навигатор – создается специальный навигатор для быстрого перемещения по страницам;
- Одной страницей – все страницы будут сохранены в один файл;
- Открыть после экспорта – результирующий файл будет открыт сразу же после экспорта программой просмотра HTML файлов, назначенной в операционной системе по умолчанию.

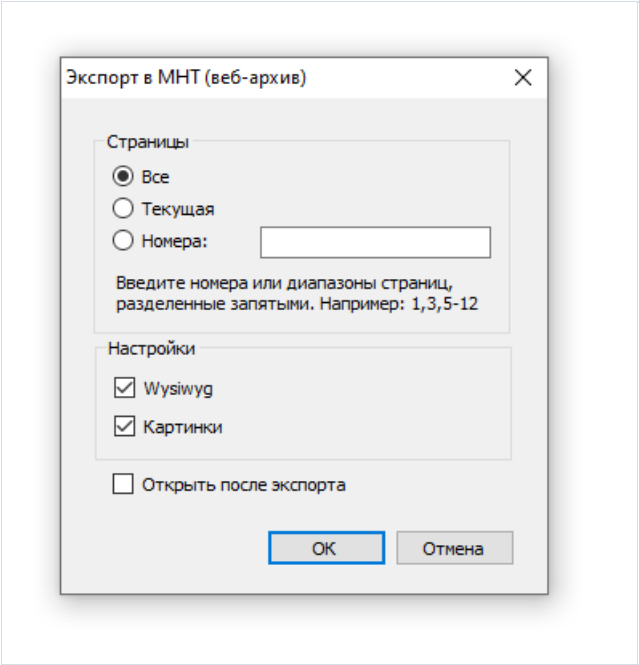
> Внешний вид и объем файла сильно зависят от шаблона отчета (см. раздел "Рекомендации по разработке отчетов").

Экспорт в MHT (веб-архив)

Формат MHT (веб-архив) основан на MIME кодировании и позволяет хранить в одном файле текстовый и графический контент страницы. Сохранение страниц в MHT поддерживается Internet Explorer.

Метод экспорта – табличный.

При экспорте в MHT будет предложено диалоговое окно для настройки параметров выходного документа:



Параметры экспорта:

- Wysiwyg - результат экспорта будет выглядеть максимально приближенно к отчету;
- Картинки – включает возможность экспорта графических изображений;
- Открыть после экспорта – результирующий файл будет открыт сразу же после экспорта в Internet Explorer.

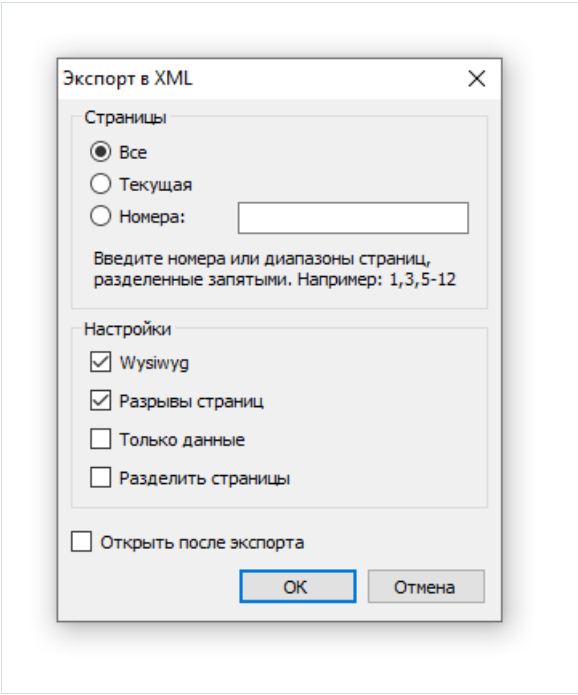
> Внешний вид и объем файла сильно зависят от шаблона отчета (см. раздел "Рекомендации по разработке отчетов").

Экспорт в Excel (XML)

Excel – приложение для работы с электронными таблицами, включенное в систему Microsoft Office.

Метод экспорта – табличный.

При экспорте в Excel будет предложено диалоговое окно для настройки параметров выходного документа:



Параметры экспорта:

- Wysiwyg – максимальное соответствие внешнему виду отчета, при отключении этой опции будет производиться оптимизация по уменьшению количества строк и столбцов в результирующей таблице;
- Разрывы страниц – включает разрыв страниц;
- Открыть после экспорта – результирующий файл будет открыт в Excel сразу же после экспорта.

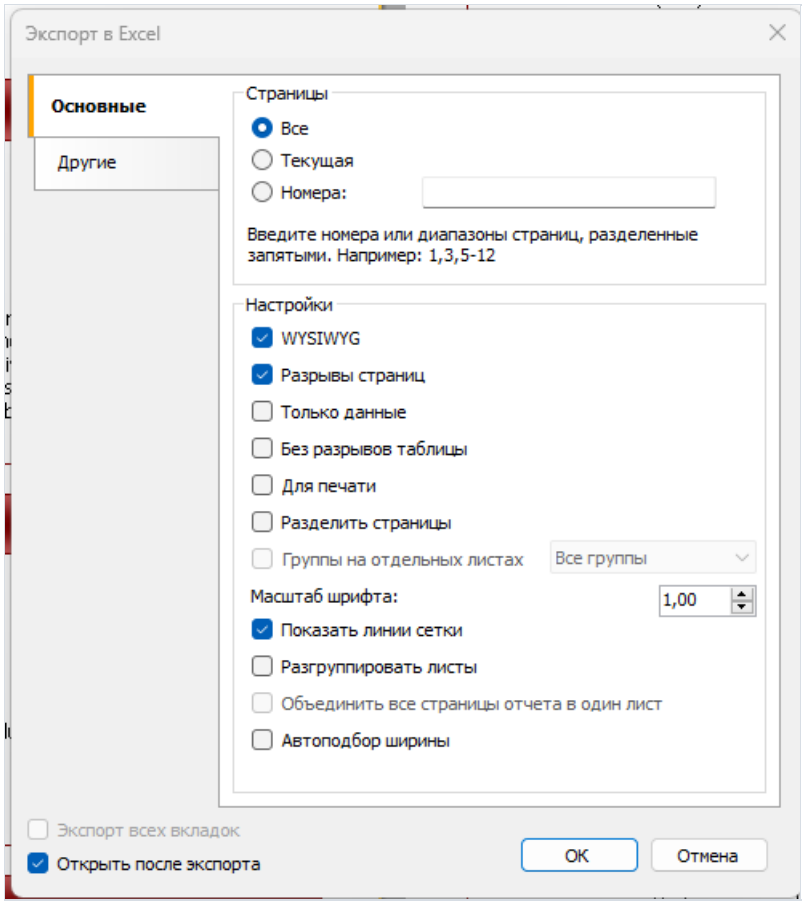
> Внешний вид и объем файла сильно зависят от шаблона отчета (см. раздел "Рекомендации по разработке отчетов").

Экспорт в Excel 2007

Excel 2007 – приложение для работы с электронными таблицами, включенное в систему Microsoft Office 2007.

Метод экспорта – табличный.

При экспорте в Excel будет предложено диалоговое окно для настройки параметров выходного документа:

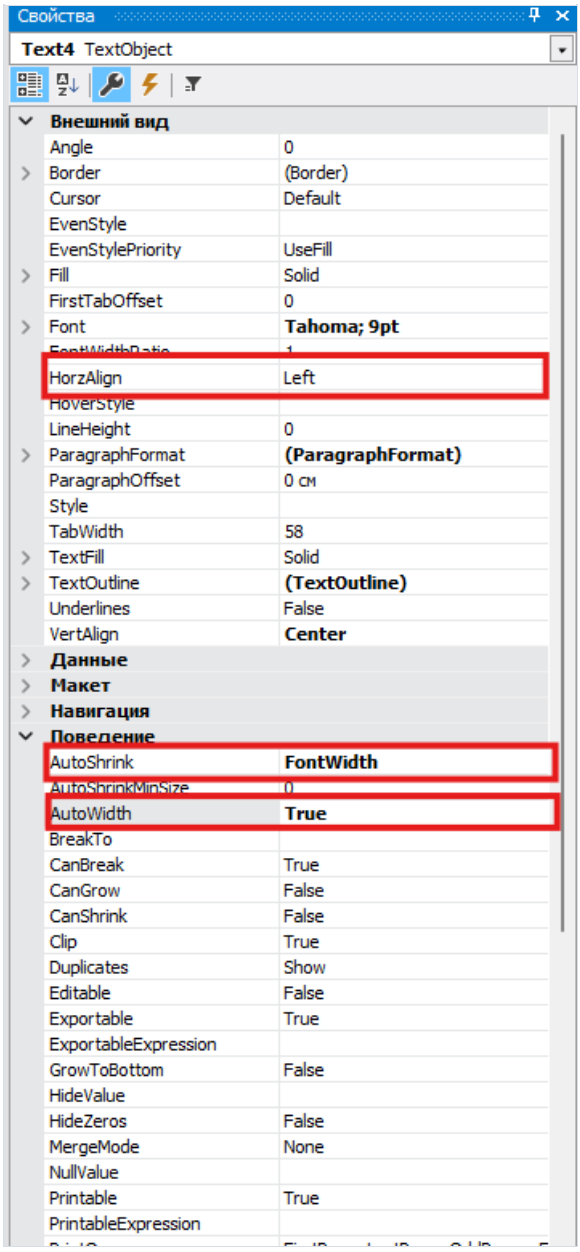


Параметры экспорта:

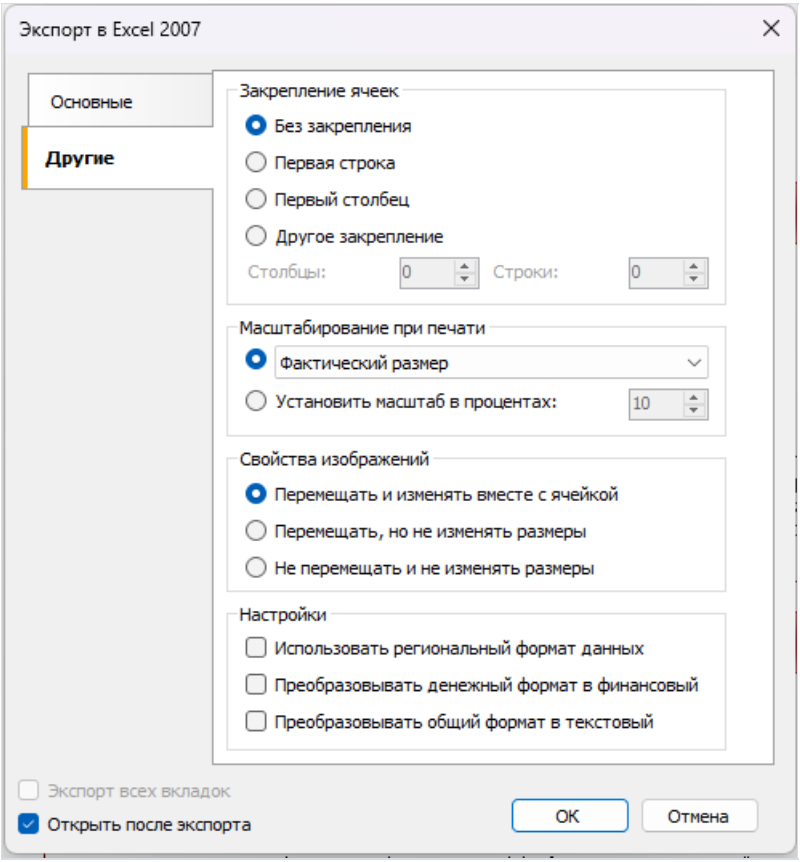
- Wysiwyg – максимальное соответствие внешнему виду отчета, при отключении этой опции будет производиться оптимизация по уменьшению количества строк и столбцов в результирующей таблице;
- Разрывы страниц – включает разрыв страниц;
- Разделить страницы – каждая страница подготовленного отчета экспортируется в отдельный лист;
- Группы на отдельных листах – позволяет разместить данные из каждой группы в отдельном листе. Если в документе присутствует только одна группа или же группы отсутствуют, в выпадающем списке будет доступен только один вариант – "Все группы";
- Разгруппировать листы – листы, объединенные в группу, автоматически синхронизируются: любые изменения, внесенные на одном из них, применяются ко всем выделенным листам. По умолчанию все листы сгруппированы;
- Автоподбор ширины – включает в Excel одно свойство из категории "Отображение" – автоподбор ширины для ячейки;
- Экспорт всех вкладок – экспорт всех вкладок для интерактивных отчетов, в которых можно открывать детальные отчеты в новых вкладках;
- Открыть после экспорта – результирующий файл будет открыт в Excel сразу же после экспорта.

Автоподбор ширины работает только при определенных значениях свойств текстового объекта:

- Автоматическая ширина = true;
- AutoShrink = режим автоматической обрезки - ширина шрифта;
- HorzAlign имеет любое значение, кроме Justify.



Разделы на вкладке "Другие" предоставляют инструменты для настройки внешнего вида и поведения данных при экспорте в Excel. В каждом разделе предоставлены опции, позволяющие точно настроить формат и отображение данных в Excel:



Раздел "Закрепление ячеек" позволяет выбрать способ закрепления определенных областей листа Excel при экспорте. Это может быть полезно для сохранения видимости определенных строк или столбцов при прокрутке или масштабировании таблицы в Excel.

Раздел "Масштабирование при печати" позволяет настроить способ печати данных, определяя, каким образом они будут распределены на страницах при выводе на печать. Это включает в себя выбор масштаба и размещение данных на странице для оптимального отображения.

Раздел "Свойства изображений" предоставляет возможность настройки поведения изображений при их размещении в ячейках Excel. Выбор доступных опций определяет, будут ли изображения перемещаться и изменять размеры вместе с ячейками при изменении размеров ячеек или перемещении данных в таблице.

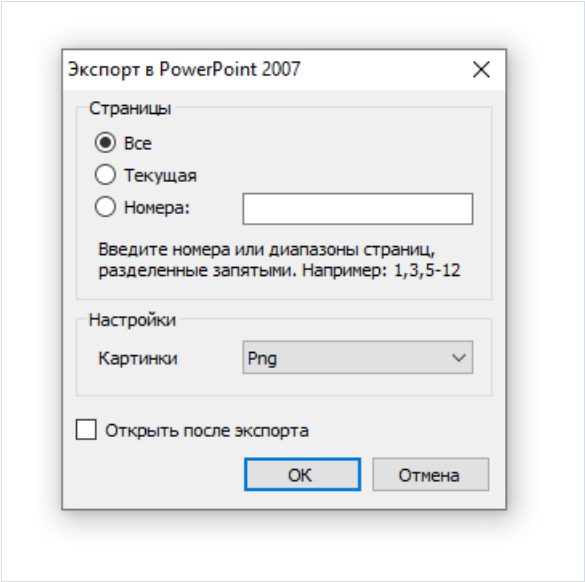
> Внешний вид и объем файла сильно зависят от шаблона отчета (см. раздел "Рекомендации по разработке отчетов").

Экспорт в PowerPoint 2007

PowerPoint 2007 – приложение для работы с презентациями, включенное в систему Microsoft Office 2007.

Метод экспорта – послойный.

При экспорте в PowerPoint 2007 будет предложено диалоговое окно для настройки параметров выходного документа:



Параметры экспорта:

- Картинки - выбор формата, в котором сохранять графические изображения;
- Открыть после экспорта – результирующий файл будет открыт в PowerPoint сразу же после экспорта.

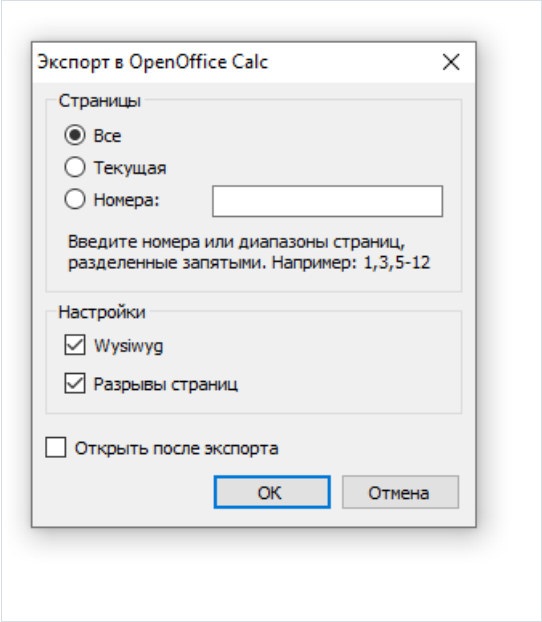
Экспорт в OpenOffice Calc

OpenDocument Format - открытый формат файлов документов для хранения и обмена редактируемыми офисными документами, в том числе текстовыми документами (такими как заметки, отчёты и книги), электронными таблицами, рисунками, базами данных, презентациями. Этот стандарт был разработан индустриальным сообществом OASIS и основан на XML-формате, изначально созданном OpenOffice.org. 1 мая 2006 года принят как международный стандарт ISO/IEC 26300.

FastReport поддерживает экспорт в таблицу (расширение .ods) OpenDocumet. Эти файлы могут быть открыты с помощью бесплатного офисного пакета OpenOffice.

Метод экспорта – табличный.

При экспорте будет предложено диалоговое окно для настройки параметров выходного документа:



Параметры экспорта:

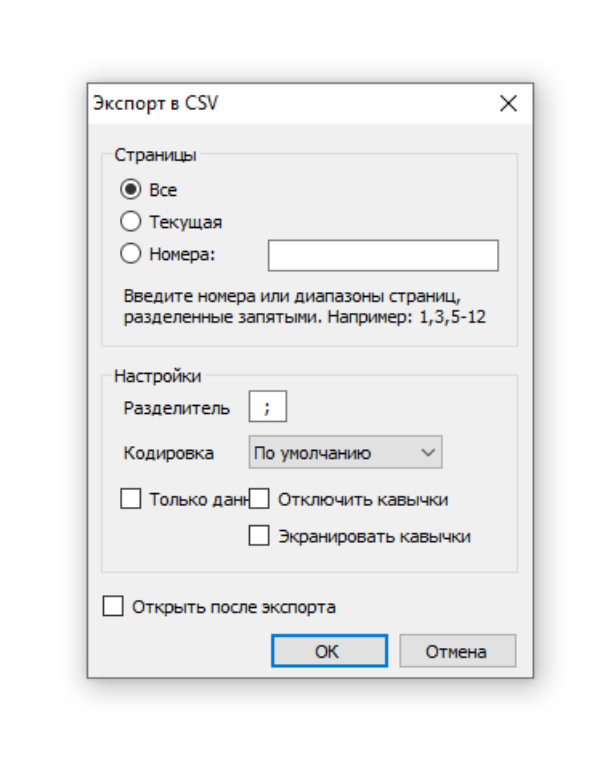
- Wysiwyg – максимальное соответствие внешнему виду отчета, при отключении этой опции будет производиться оптимизация по уменьшению количества строк и столбцов в результирующей таблице;
- Разрывы страниц – включает разрыв страниц;
- Открыть после экспорта – результирующий файл будет открыт сразу же после экспорта.

Экспорт в CSV

CSV-файл содержит значения, отформатированные в виде таблицы и упорядоченные таким образом, что каждое значение в столбце отделено от значения в следующем столбце разделителем, а каждый новый ряд начинается с новой строки. Данный формат может быть импортирован в различные табличные редакторы.

Метод экспорта – табличный.

При экспорте в CSV будет предложено диалоговое окно для настройки параметров выходного документа:



Параметры экспорта:

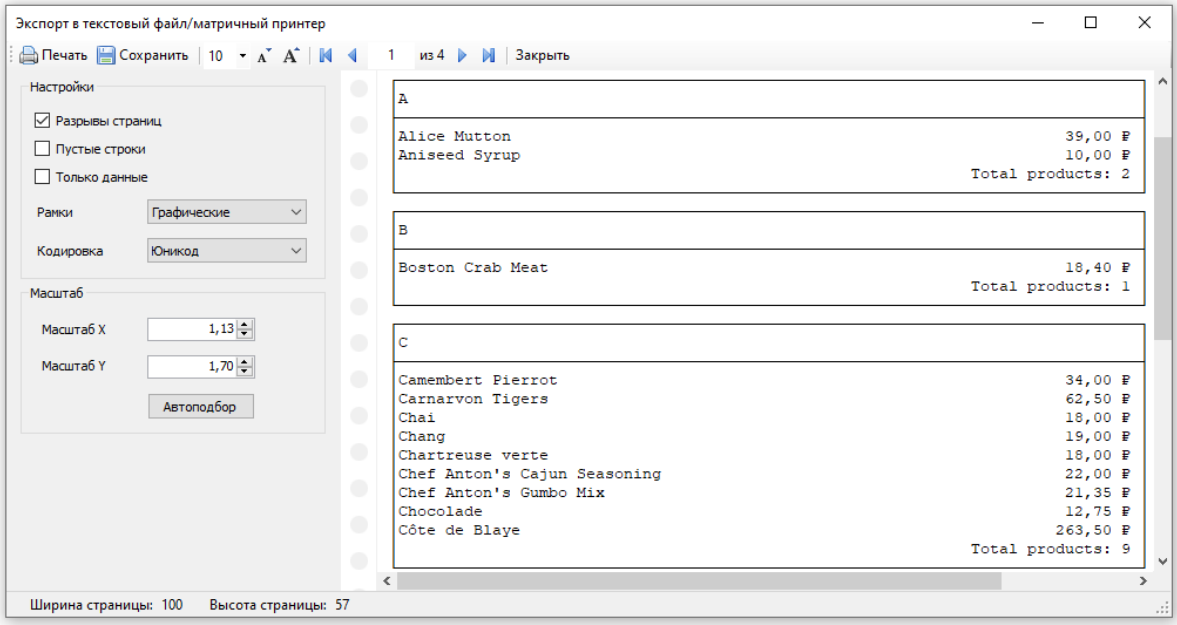
- Разделитель - выбор символа-разделителя;
- Кодировка - выбор кодировки документа. Кодировка "По умолчанию" соответствует текущей кодовой странице Windows;
- Только данные - экспортированы будут только объекты, лежащие на бэнде "Данные";
- Открыть после экспорта – результирующий файл будет открыт в Excel сразу же после экспорта.

Экспорт в TXT

Обычный текстовый файл – содержит информацию из отчета, максимально оптимизированную и преобразованную в связи со спецификой данного формата. Полученный документ может быть сохранен в файл или распечатан на матричном принтере.

Метод экспорта – табличный.

При экспорте в текст будет предложено диалоговое окно для настройки параметров выходного документа:



Параметры экспорта:

- Разрывы страниц - включает разрыв страниц;
- Пустые строки - включает пустые строки в результирующий файл;
- Только данные - экспортированы будут только объекты, лежащие на бэнде "Данные";
- Рамки - выбор типа рамок (графические/текстовые/без рамок);
- Кодировка - выбор кодировки текста;
- Масштаб X - масштабирование по горизонтали;
- Масштаб Y - масштабирование по вертикали;
- Автоподбор - автоматический подбор таких коэффициентов X и Y, чтобы не было потери информации.

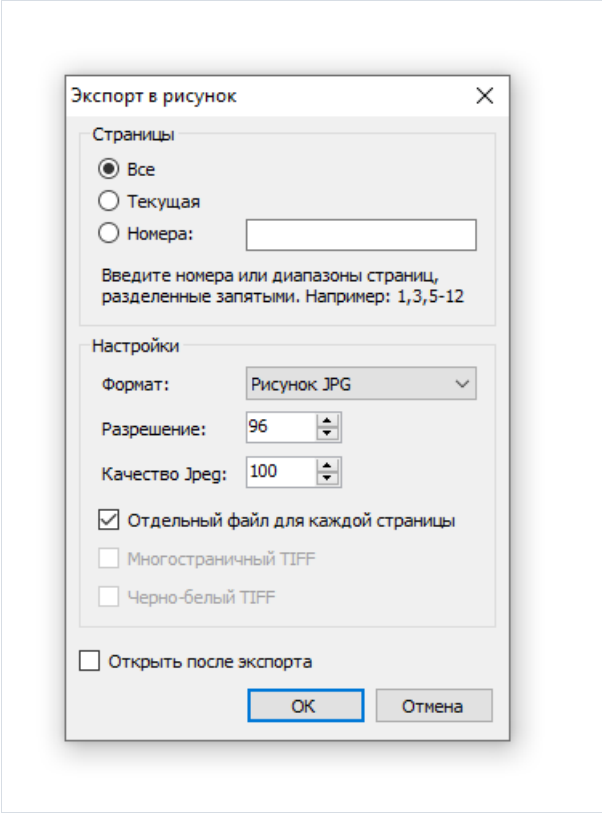
Экспорт в файлы рисунков

FastReport позволяет экспортировать информацию в следующие графические форматы:

- BMP;
- PNG;
- JPG;
- GIF;
- TIFF;
- Метафайл (EMF,WMF).

Принцип экспорта – отрисовка.

При экспорте в файл рисунка будет предложено диалоговое окно для настройки параметров изображения:



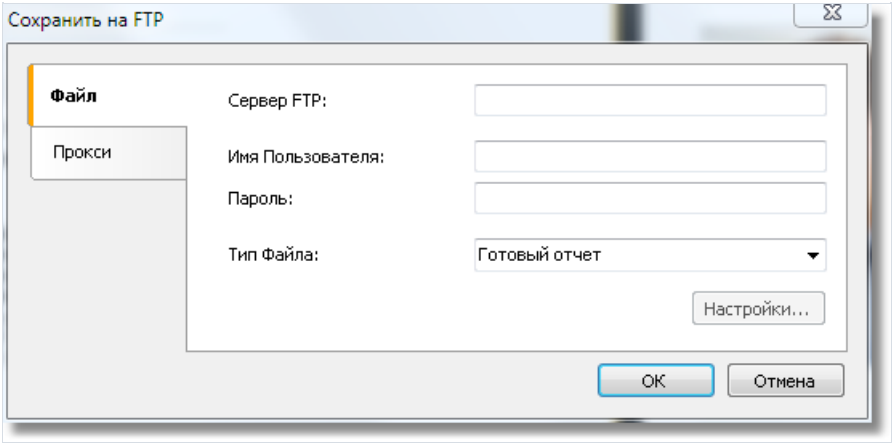
Параметры экспорта:

- Формат - формат изображения;
- Разрешение (dpi) – разрешение изображения. При экспорте в TIFF формат можно задать разные значения для разрешения по вертикали и горизонтали;
- Качество Jpeg – степень сжатия JPG файла. Опция используется только при экспорте в Jpeg формат;
- Отдельный файл для каждой страницы – если опция включена, то каждая страница отчета будет экспортирована в отдельный файл, имя файла будет сформировано на основе выбранного с добавлением номера страницы;
- Многостраничный TIFF - при включении опции будет сформирован один TIFF файл, содержащий несколько страниц. Опция используется только при экспорте в TIFF формат. При этом опция "Отдельный файл для каждой страницы" игнорируется;
- Черно-белый TIFF - при включении опции будет сформирован черно-белый TIFF файл. Опция используется только при экспорте в TIFF формат.

При экспорте нескольких страниц в один файл (при отключенной опции "Отдельный файл для каждой страницы") нужно помнить о большой ресурсоемкости экспорта.

Экспорт на FTP

В превью FastReport .NET есть возможность сохранить готовый отчет на FTP сервер. Отчет также можно экспортировать в любой из поддерживаемых форматов и после этого сохранить на FTP. Для этого нужно нажать кнопку "Сохранить" и выбрать пункт "FTP...". Появится окно сохранения на FTP сервер:

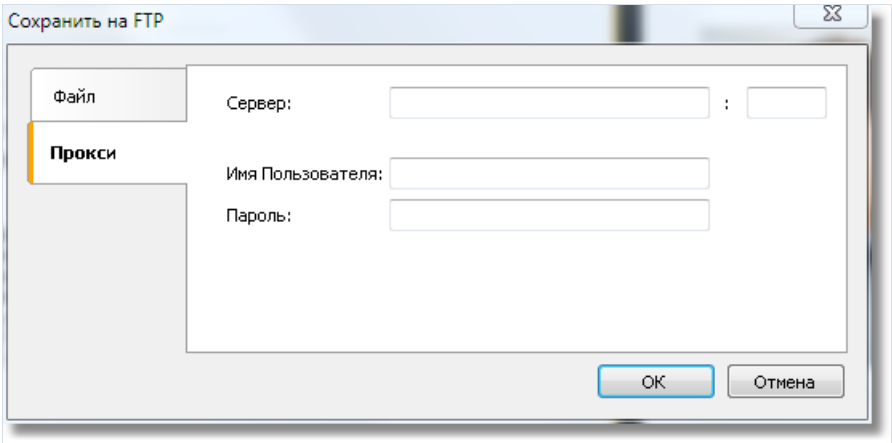


На вкладке "Файл" есть следующие поля:

- Сервер FTP. В это поле нужно ввести URL-адрес FTP сервера, на который нужно сохранить файл;
- Имя Пользователя и Пароль. Нужно ввести ваши логин и пароль на данном FTP-сервере;
- Тип файла. Выпадающий список, в котором можно выбрать формат сохранения файла (Готовый отчет или один из экспортов);

При выборе одного из экспортов становится доступна кнопка "Настройки...". Нажав на эту кнопку, можно перейти к окну настроек выбранного экспорта.

Если вы используете прокси-сервер, то на вкладке прокси вы можете ввести URL-адрес прокси сервера, порт, логин и пароль:



Введя настройки и выбрав формат остается только нажать кнопку OK и файл будет сохранен на сервере FTP.

Экспорт в Dropbox

В превью FastReport .NET есть возможность сохранить готовый отчет в Dropbox. Отчет также можно предварительно экспортировать в любой из поддерживаемых форматов.

Перед тем как воспользоваться этой возможностью, необходимо создать приложение в Dropbox аккаунте. Для этого нужно войти в свой аккаунт на Dropbox и выполнить следующие шаги:

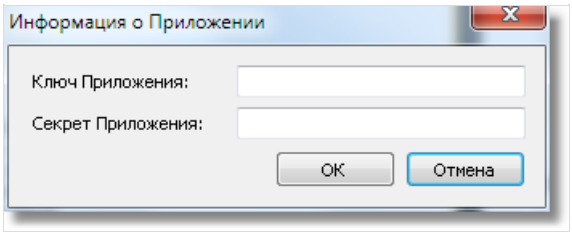
- Щелкнуть кнопку "Еще (More)". Она находится внизу страницы Dropbox.
- В выпадающем списке выбрать "Разработчикам (Developers)". Вы попадете на страницу для разработчиков.
- Перейти по ссылке "App Console". В результате вы попадете к списку приложений.
- Щелкнуть кнопку "Create App". Dropbox захочет проверить ваш E-mail. Нажмите кнопку "Send Email".

На вашу почту будет отправлено письмо, в котором нужно нажать "Подтвердить адрес электронной почты".

В итоге вы попадете на страницу "Create a new Dropbox Platform app". Здесь нужно выбрать "Dropbox API app" и на вопрос "What type of data does your app need to store on Dropbox?" выбрать ответ "Files and datastores". А на вопрос "Can your app be limited to its own, private folder?" можно выбрать любой из двух предложенных ответов. Последним на этой странице нужно ввести имя приложения (оно может быть любым). После нажатия на кнопку "Create app" система проверит, не занято ли уже введенное вами имя приложения, и создаст приложение.

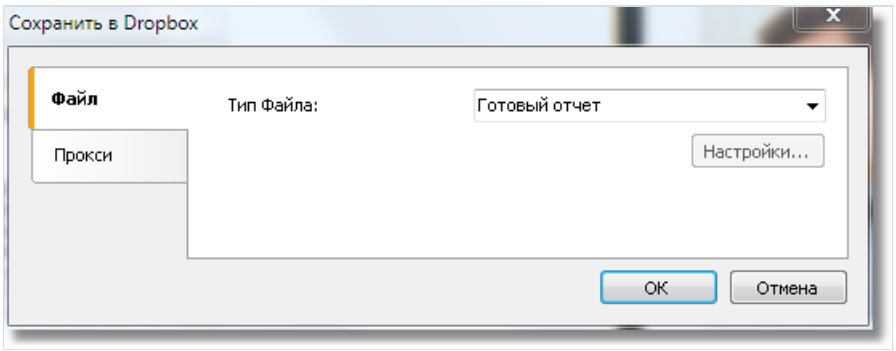
В итоге мы попадем на страницу настройки приложения. Здесь нам интересны "App key" и "App secret", они понадобятся при экспорте в Dropbox.

Теперь можно перейти в превью FastReport .NET и экспортировать файл в Dropbox. Для этого нужно нажать на кнопку "Сохранить" и выбрать пункт "Dropbox...". При первой попытке экспортировать в Dropbox появится окно "Информация о Приложении":



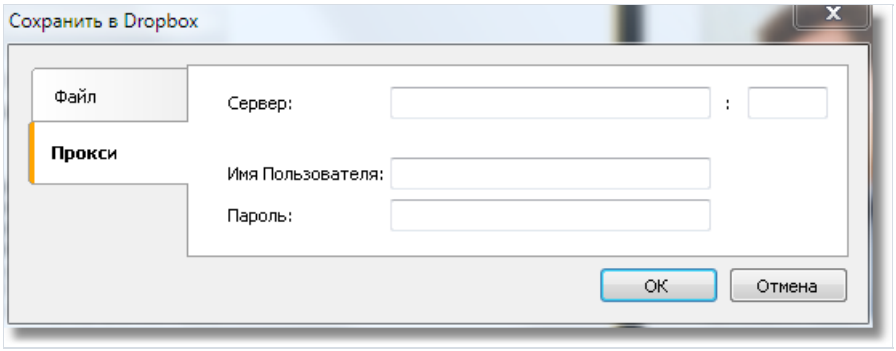
Здесь нужно ввести полученные ранее "Ключ Приложения (App key)" и "Секрет Приложения (App secret)". После нажатия на кнопку "OK" FastReport сохранит эти значения и больше их вводить не потребуется.

Появится окно сохранения в Dropbox:



На вкладке "Файл" можно выбрать тип сохраняемого файла (Готовый отчет или один из экспортов). При выборе одного из экспортов становится доступна кнопка "Настройки...". Нажав на эту кнопку, можно перейти к окну настроек выбранного экспорта.

Если вы используете прокси-сервер, то на вкладке "Прокси" вы можете ввести URL-адрес прокси сервера, порт, логин и пароль:



Введя настройки и выбрав формат, остается только нажать кнопку "OK" и файл будет сохранен в Dropbox.

Экспорт в Google Drive

В превью FastReport .NET есть возможность сохранить готовый отчет в GoogleDrive. Отчет также можно предварительно экспортировать в любой из поддерживаемых форматов.

Для начала нужно создать приложение в GoogleDrive. Для этого нужно зайти по адресу https://code.google.com/apis/console

На этой странице принимаем условия лицензионного соглашения.

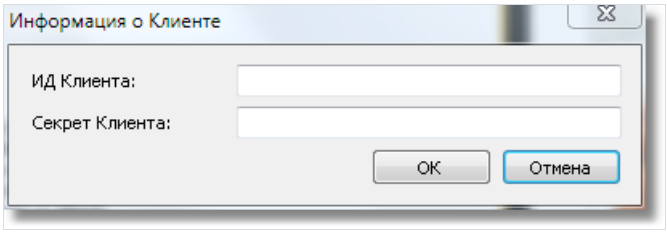
После этого мы попадаем на страницу управления проектами. Здесь переходим на вкладку "Services" и активируем Drive API. Переходим на вкладку "API Access" и жмем "Create an OAuth 2.0 client ID". В секции "Branding Information" вводим имя приложения и жмем Next. В секции "Client ID Settings" выбираем следующее:

- "Installed application" для Application type.
- "Other" для Installed application type.

Жмем "Create Client ID".

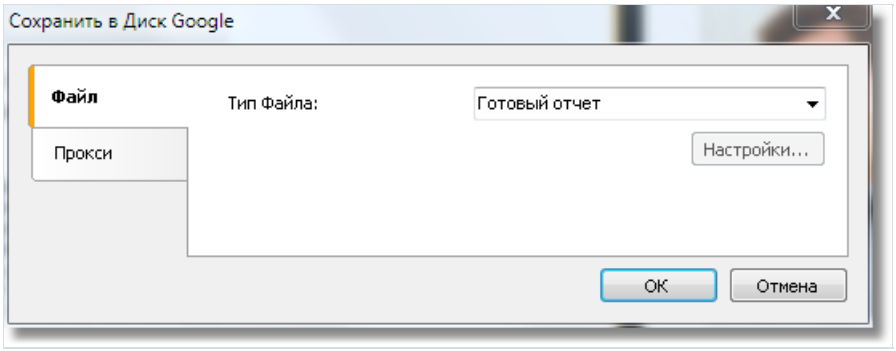
В результате на странице "API Access page" появится секция "Client ID for installed applications", в которой мы увидим Client ID и Client secret. Оба этих значения нам понадобятся в дальнейшем.

Теперь можно перейти в превью FastReport .NET и экспортировать файл в GoogleDrive. Для этого нужно нажать кнопку "Сохранить" и выбрать пункт "GoogleDrive...". При первой попытке экспортировать в GoogleDrive появится окно "Информация о Клиенте":



Здесь нужно ввести полученные ранее ИД Клиента (Client ID) и Секрет Клиента (Client secret). После нажатия на кнопку "OK" FastReport сохранит эти значения, и больше их вводить не потребуется.

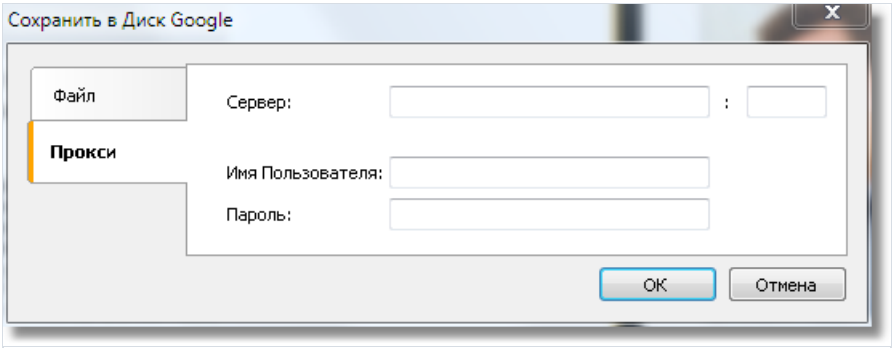
Появится окно сохранения в GoogleDrive:



На вкладке "Файл" можно выбрать тип сохраняемого файла (Готовый отчет или один из экспортов). При выборе одного из экспортов становится доступна кнопка "Настройки...". Нажав на эту кнопку, можно перейти к окну настроек выбранного экспорта.

Если вы используете прокси-сервер, то на вкладке "Прокси" вы можете ввести URL-адрес прокси сервера,

порт, логин и пароль:



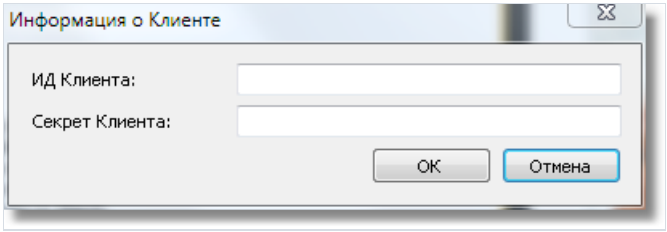
Введя настройки и выбрав формат, остается только нажать кнопку "OK" и файл будет сохранен в GoogleDrive.

Экспорт в OneDrive

В превью FastReport .NET есть возможность сохранить готовый отчет в OneDrive. Отчет также можно предварительно экспортировать в любой из поддерживаемых форматов.

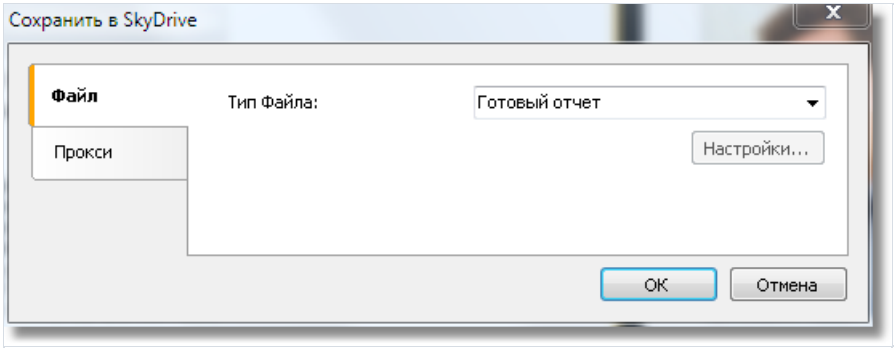
Для начала нужно создать приложение в OneDrive. Для этого на странице OneDrive переходим по ссылке "Разработчикам". На странице центра разработки жмем "Мои приложения". Далее жмем "Создать приложение". Вводим имя приложения и выбираем язык. Читаем "условия использования" и "заявление о конфиденциальности" и жмем кнопку "Я принимаю". В итоге попадаем на страницу настроек приложения. Здесь мы видим "ИД клиента" и "Секрет клиента", которые нам и нужны. Остается только указать домен перенаправления и нажать кнопку "Сохранить".

Теперь можно перейти в превью FastReport .NET и экспортировать файл в OneDrive. Для этого нужно нажать кнопку "Сохранить" и выбрать пункт "OneDrive...". При первой попытке экспортировать в OneDrive появится окно Информация о Клиенте:



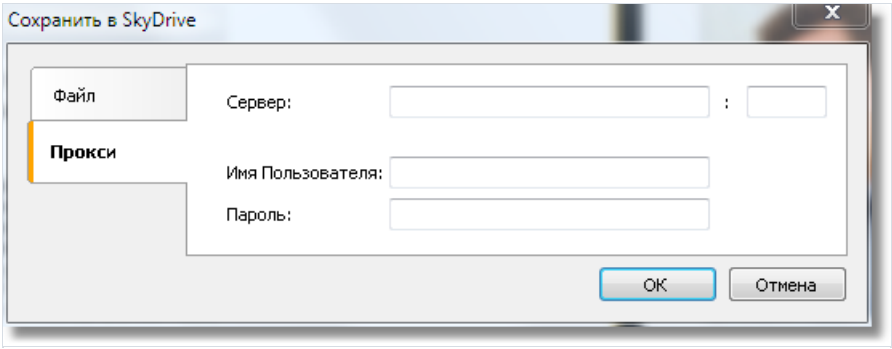
Здесь нужно ввести полученные ранее ИД Клиента (Client ID) и Секрет Клиента (Client secret). После нажатия на кнопку "OK" FastReport сохранит эти значения, и больше их вводить не потребуется.

Появится окно сохранения в OneDrive:



На вкладке "Файл" можно выбрать тип сохраняемого файла (Готовый отчет или один из экспортов). При выборе одного из экспортов становится доступна кнопка "Настройки...". Нажав на эту кнопку, можно перейти к окну настроек выбранного экспорта.

Если вы используете прокси-сервер, то на вкладке "Прокси" вы можете ввести URL-адрес прокси сервера, порт, логин и пароль:



Введя настройки и выбрав формат, остается только нажать кнопку "OK" и файл будет сохранен в OneDrive.

Рекомендации по разработке отчетов

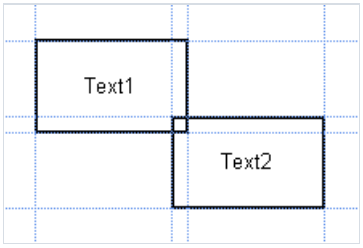
В этом разделе будут даны рекомендации по разработке отчетов, предназначенных для экспорта в другие форматы данных.

FastReport предлагает широкие возможности для манипуляции объектами при создании отчета, что дает заметное преимущество при быстрой разработке любых отчетов и последующей их печати на принтере. Отпечатанный документ будет выглядеть точно так же, как и на экране, что и является основной целью применения генератора отчетов FastReport.

Обратная сторона такой свободы разработки – сложность экспорта полученного документа в различные форматы данных, имеющие свои, иногда довольно существенные, ограничения в представлении информации. Очень многие форматы используют табличное представление данных. Речь идет о таких форматах, как HTML, XLS, RTF, CSV. Никакие пересечения или наслоения ячеек в подобных форматах недопустимы (если брать в рассмотрение именно табличную разметку, это касается HTML и RTF).

Фильтры экспорта, как правило, максимально учитывают эти требования при экспорте объектов. Это реализуется при помощи специальных алгоритмов учета пересечений объектов и оптимального их расположения. В местах пересечений объектов возникают новые столбцы и строки в результирующей таблице. Это необходимо для получения максимального сходства результата экспорта и оригинального отчета. Большое количество пересекающихся объектов в отчете приводит к росту числа столбцов и строк в таблице, что замедляет процесс экспорта и усложняет использование результирующего файла.

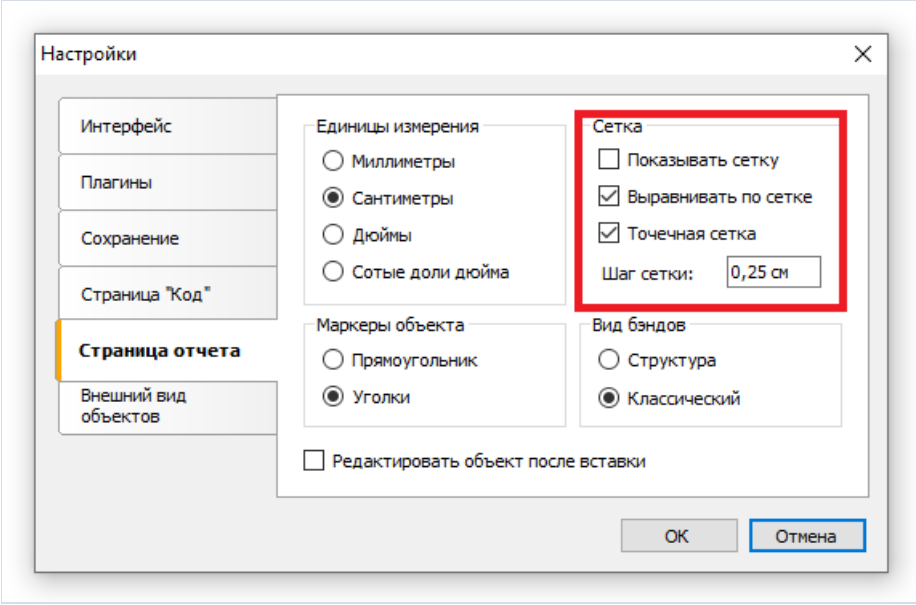
Качество экспорта в тот или иной формат сильно зависит от грамотной разработки шаблона отчета. Рассмотрим следующий отчет:



При разработке этого отчета было допущено пересечение двух объектов, находящихся на одном бэнде. Предположим, что число записей при формировании отчета составило 150. Тогда при экспорте в формат RTF будет создано 450 строк в результирующей таблице (150 строк на каждый объект и 150 строк на пересечение). Если пересечение устранить, количество строк будет уже 300. На больших отчетах и при большем количестве объектов разница будет просто огромной, что, соответственно, скажется и на размере выходного файла.

При создании таблиц в отчетах проследите, чтобы границы соседних ячеек соприкасались друг с другом, но не пересекались или наслаивались. Алгоритм фильтра экспорта сделает отсечение ячеек, но результат экспорта может быть далек от желаемого (вы увидите не совсем то, что хотели). Располагайте объекты так, чтобы они находились на одной линии, как по вертикали, так и по горизонтали.

Избежать перекрытия ячеек может помочь выравнивание объектов по сетке. Проследите за тем, чтобы было включено выравнивание по сетке в опциях дизайнера. Для упрощения выравнивания можно увеличить шаг сетки. Настройки шага сетки и выравнивания можно найти в меню дизайнера «Вид|Настройки...»:



Для обрамления текста рамкой лучше использовать рамку объекта "Текст", а не отдельные графические объекты – линии, прямоугольники и пр. Старайтесь не использовать фоновых объектов под прозрачными текстовыми объектами.

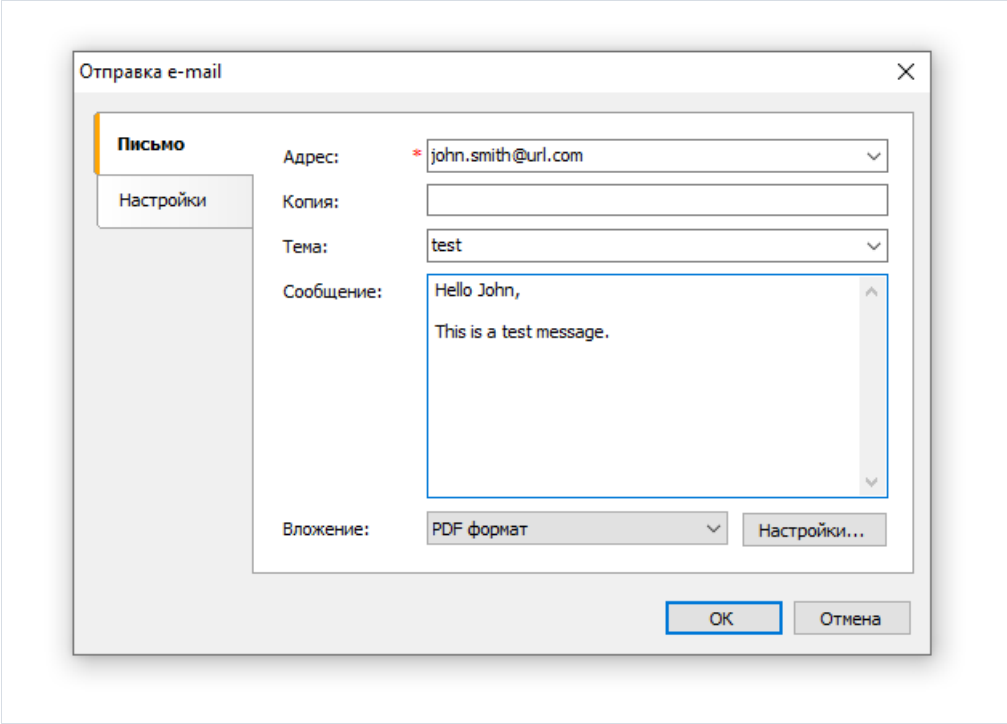
Применение этих простых правил на практике поможет вам создать отчет, который будет прекрасно выглядеть после экспорта в любой из форматов, которые используют табличную (или основанную на табличной) разметку для представления данных.

Отправка отчета по email

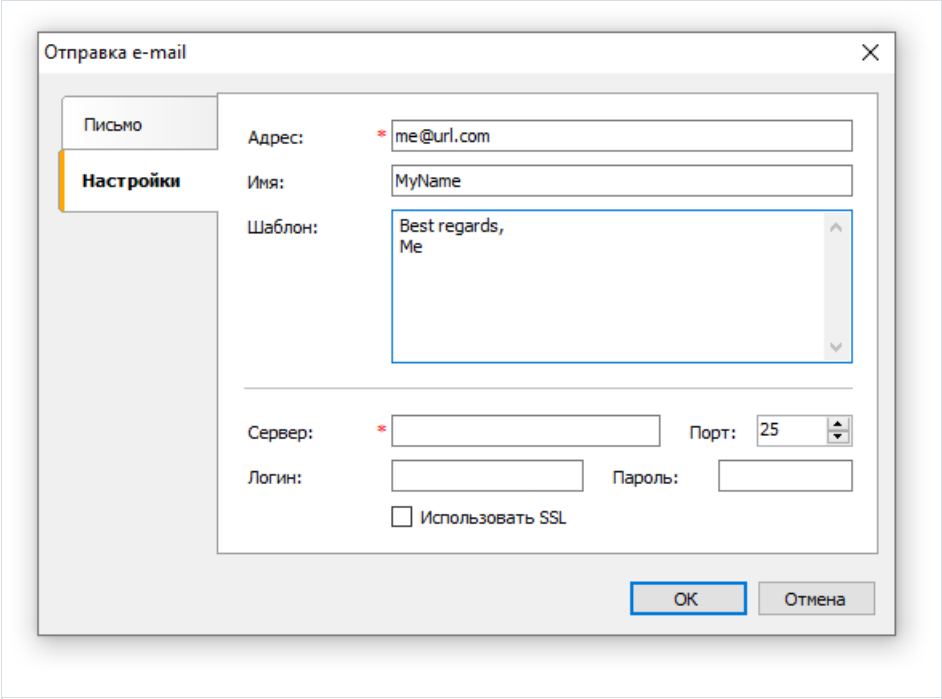
FastReport позволяет отправить готовый отчет по электронной почте в нужном вам формате. Отправка работает в двух режимах:

Режим	Описание
SMTP	Режим работы по умолчанию. Для отправки письма не нужны никакие вспомогательные программы.
MAPI	Включить этот режим работы можно программно, указав свойство `Config.EmailSettings.UseMAPI = true`, или, используя компонент EnvironmentSettings, указав свойство `EnvironmentSettings.EmailSettings.UseMAPI = true`. Для отправки письма используется почтовая программа, установленная в системе. Программа должна быть совместима с MAPI.

Для отправки письма необходимо указать адрес получателя. Также желательно, но не обязательно, заполнить поля "Тема" и "Сообщение". В нижней части окна выберите формат файла отчета, который будет прикреплен к письму:



Если вы используете режим SMTP, перед отправкой письма необходимо настроить параметры почтового ящика. Сделать это необходимо только один раз. Все параметры сохраняются в файле конфигурации FastReport и будут использованы в дальнейшем. Параметры находятся на закладке "Настройки". Поля, обязательные к заполнению, помечены красными звездочками:



Если ваш сервер требует аутентификацию, необходимо также заполнить поля "Логин" и "Пароль".

Минимальные системные требования

Минимальные системные требования для установки и использования FastReport .NET:

- Операционная система MS Windows 7-11, Windows Server 2012-2019;
- Процессор: 1 ГГц;
- ОЗУ: 512 Мб;
- Также для работы приложения вам необходим установленный .NET Framework не ниже версии 4.6.2.

Контакты и техподдержка

Вы всегда можете задать вопросы по использованию продукта с помощью email, либо с помощью формы на сайте.

Также мы будем рады вашим предложениям по улучшению нашего продукта.