**Лекция 8\_****Создание отчетов(Использование контролаReportViewer)**

1. УстановкаПО
2. Создание отчета с помощью мастера отчетов VisualStudio
3. Создание текстового документа на основе базы данных
4. Добавление диаграммы в отчет
5. [Справочный материал (Табликс)](http://wladm.narod.ru/C_Sharp/ReportViewer.html#prs)

Отчетом называется объект, представляющий собой приготовленный для печати документ, содержащий данные из приложения (из базы данных).

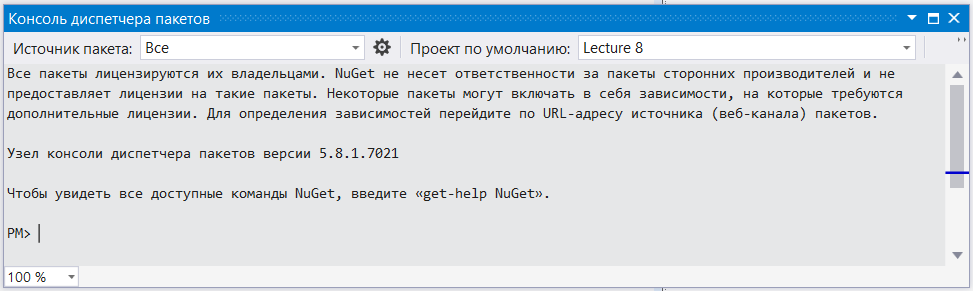
Отчеты позволяют пользователю быстро формировать и распечатывать документы, содержащие актуальные данные.

Элемент управления ReportViewer- это элемент управления среды VisualStudio, который можно добавить на веб-страницу в проекте решения ASP.NET или в форму приложения Windows. Этот элемент управления включает возможности обработки и просмотра отчетов непосредственно в приложении.

1. **Установка ПО**

**1.1 Установка пакета ReportViewer**

1. Открыть проект.
2. Открыть консоль диспетчера пакетов NuGet (Tools/NuGetPackageManager/PackageManagerConsole Средства-Диспетчер пакетов NuGet-Консоль диспетчера пакетов). В нижней части проекта появится консоль диспетчера пакетов



1. В командной строке PM > ввести следующую команду

***Install-Package Microsoft.ReportingServices.ReportViewerControl.WinForms***

При этом будет отображаться текст, описывающий установку пакета.

Затемпоявитсяследующеесообщение

# Action required to load native assemblies

To deploy an application that uses spatial data types to a machine that does not have 'System CLR Types for SQL Server' installed you also need to deploy the native assembly SqlServerSpatial140.dll. Both x86 (32 bit) and x64 (64 bit) versions of this assembly have been added to your project under the SqlServerTypes\x86 and SqlServerTypes\x64 subdirectories. The native assembly msvcr120.dll is also included in case the C++ runtime is not installed.

You need to add code to load the correct one of these assemblies at runtime (depending on the current architecture).

## ASP.NET Web Sites

For ASP.NET Web Sites, add the following block of code to the code behind file of the Web Form where you have added Report Viewer Control:

Default.aspx.cs:

public partial class \_Default :System.Web.UI.Page

{

static bool \_isSqlTypesLoaded = false;

public \_Default()

{

if (!\_isSqlTypesLoaded)

{

SqlServerTypes.Utilities.LoadNativeAssemblies(Server.MapPath("~"));

\_isSqlTypesLoaded = true;

}

}

}

## ASP.NET Web Applications

For ASP.NET Web Applications, add the following line of code to the Application\_Start method in Global.asax.cs:

SqlServerTypes.Utilities.LoadNativeAssemblies(Server.MapPath("~/bin"));

## Desktop Applications

For desktop applications, add the following line of code to run before any spatial operations are performed:

SqlServerTypes.Utilities.LoadNativeAssemblies(AppDomain.CurrentDomain.BaseDirectory);

1. Добавить элемент ReportViewer в проект c помощью кода или в панель элементов

Код

***private Microsoft.Reporting.WinForms.ReportViewer reportViewer1;***

***private void InitializeComponent()***

***{***

***this.reportViewer1 = new Microsoft.Reporting.WinForms.ReportViewer();***

***this.SuspendLayout();***

***//***

***// reportViewer1***

***//***

***this.reportViewer1.Location = new System.Drawing.Point(168, 132);***

***this.reportViewer1.Name = "reportViewer1";***

***this.reportViewer1.ServerReport.BearerToken = null;***

***this.reportViewer1.Size = new System.Drawing.Size(396, 246);***

***this.reportViewer1.TabIndex = 0;***

***//***

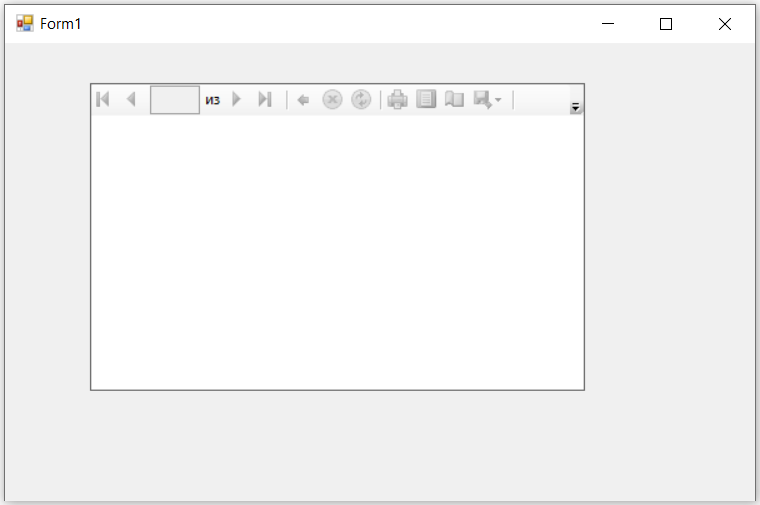
***// Form1***

***//***

***this.Controls.Add(this.reportViewer1);***

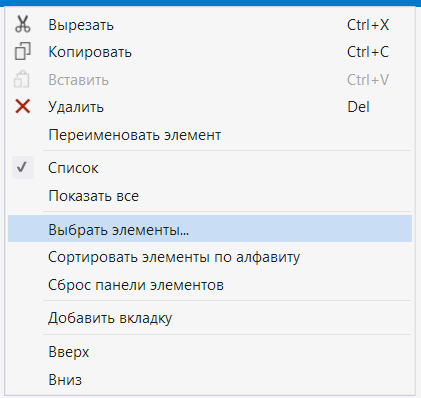
***}***

Форма с установленным объектом ReportViewerимеет вид:

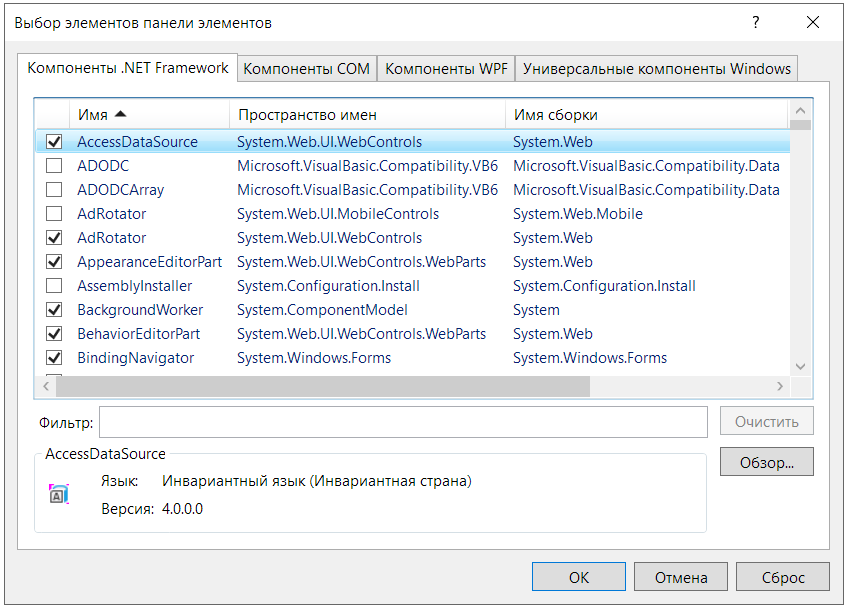


## ДобавлениеэлементаReportViewerto Visual Studio toolbar

1. Щелкнуть правой кнопкой мыши на панели элементов и выбрать в открывшемся меню пункт меню  «Выбратьэлементы» (ChooseItems).



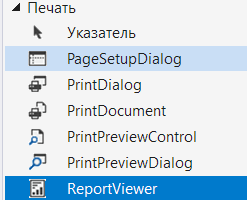
При этом откроется окно «Выбор элементов панели элементов»



1. В открывшемся окне используя кнопку «Обзор»перейти к соответствующей библиотеке DLL, расположенной в папке решений Packagesтекущего проекта. Для этого открыть папку с именемMicrosoft.ReportingServices.ReportViewerControl.Winforms.150.1404.0
2. Затем открыть папку lib и в ней открыть папку net40.
3. Наконец, выбрать файл Microsoft.ReportViewer.WinForms.dll и нажать кнопку «Открыть»
4. В окне «Выбор элементов панели элементов» появится строка «ReportViewer»

Нажать кнопку “OK” в этом окне.

В панели элементов появится элемент ReportViewer.



Для использования инструмента ReportViewer его необходимо перетащить в нужную форму(ы).

**1.2. Установкапакета Microsoft RDLC Report Designer**

* 1. Скачать с сайта <https://marketplace.visualstudio.com/items?itemName=ProBITools.MicrosoftRdlcReportDesignerforVisualStudio-18001>

Установочный пакет

Microsoft.RdlcDesigner.vsix

* 1. Запустить установочный пакет дважды щелкнув по нему мышкой.

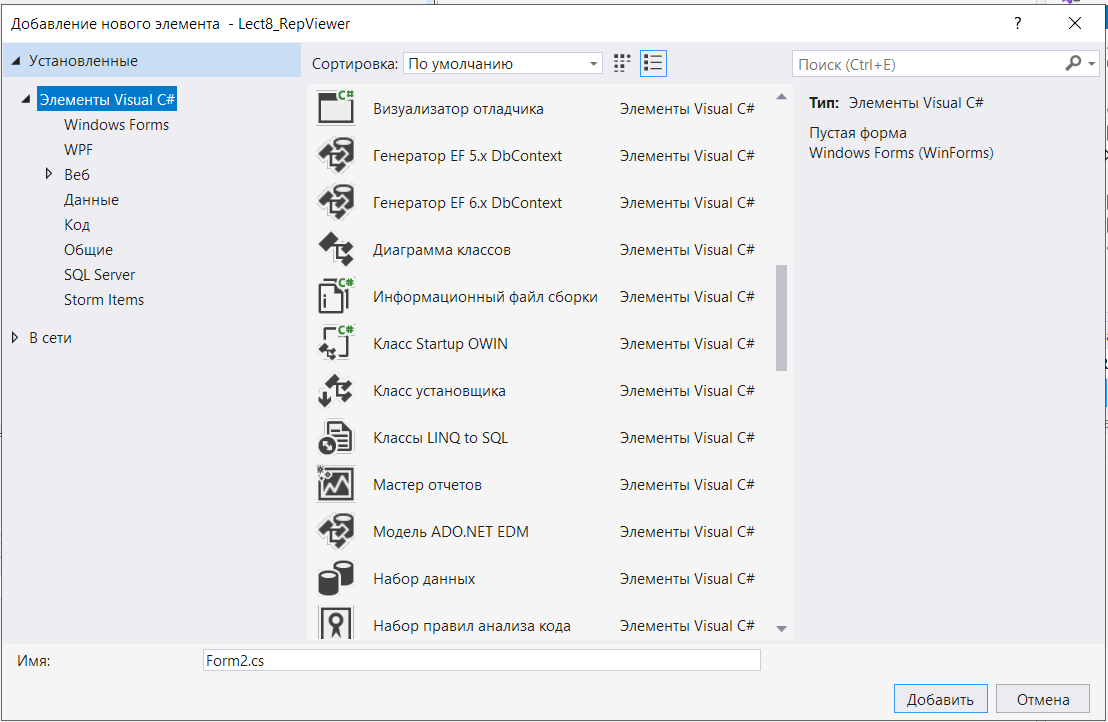
**2. Создание отчета с помощью мастера отчетов VisualStudio**

Мастер отчетов среды VisualStudio представляет собой средство в составе VisualStudio для создания RDLC-файлов, связанных с элементами управления ReportViewer. Мастер руководит процессом создания определения отчета. В мастере можно указать данные отчета, упорядочить данные по группам строк и столбцов в области данных табликса, выбрать формат макета и выбрать стиль.

Чтобы создать RDLC-файл, мастер отчетов можно открыть несколькими способами. Пример.

* Для существующего проекта или веб-узла щелкните его правой кнопкой мыши в окне **Обозреватель решений**, выберите команду **Добавить** и пункт **Создать элемент**. В

открывшемся окне выбрать **«Мастер отчетов»** и нажать кнопку **«Добавить»**



Откроется мастер отчетов.

После завершения работы мастера можно изменять и дорабатывать отчет в конструкторе отчетов среды VisualStudio.

## Шаги мастера

Сначала необходимо определить используемый набор данных на основе источника данных. Помимо возможности создания нового источника данных, мастер отчетов предоставляет список всех источников данных и связанных с ними наборов данных в проекте VisualStudio. Если нажать кнопку **Создать**, мастер настройки источника данных VisualStudio поможет выполнить действия по созданию источника данных.

### Размещение полей

Затем нужно распределить поля по группам строк, группам столбцов и строкам детализации в области данных табликса. Данные в области данных табликса выводятся в виде настраиваемой сетки со статическими и динамическими строками на основе указанных групп строк и столбцов.

### Выбор макета

Затем создайте макет отчета. Здесь можно добавить общие итоги и подытоги к сгруппированным данным. Подытоги могут располагаться над группой или под группой, а сами группы могут выводиться развернутыми или свернутыми.

### Выбор стиля

Наконец, примените стиль к отчету, используя шаблон стиля.

После выполнения всех шагов конструктор отчетов создает отчет и добавляет его в проект VisualStudio.

## Изменение определения клиентского отчета

После завершения работы мастера можно открыть отчет в конструкторе отчетов и продолжить его разработку. Например, можно добавить графические элементы, такие как линии и изображения, изменить выражения для группирования, сортировки и условной видимости, а также включить пользовательский код.

Пример построения списка сотрудников с указанием их дня рождения. В качестве источника данных будем использовать таблицуemployees из базы данных kafedraDB

1. Создадим в проекте новую форму (Form 3) и поместим на нее объектReportViewer как показано на рисунке 3.1.
2. Нажимаем в правом верхнем углу ReportViewer треугольничек и выбираем пункт Создать новый отчет. В открывшемся окне «Свойства набора данных» выбираем источник данных (DataSet1) и набор данных (positions)(Рис.3.2.) и нажимаем кнопку «Далее»

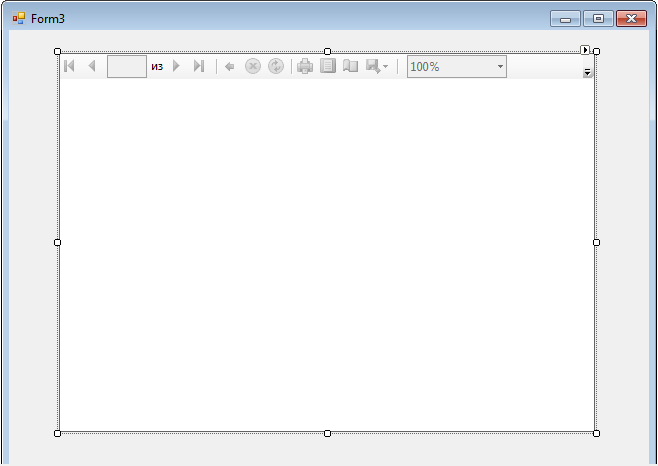


Рисунок 3.1. Форма с элементом управления ReportViewer

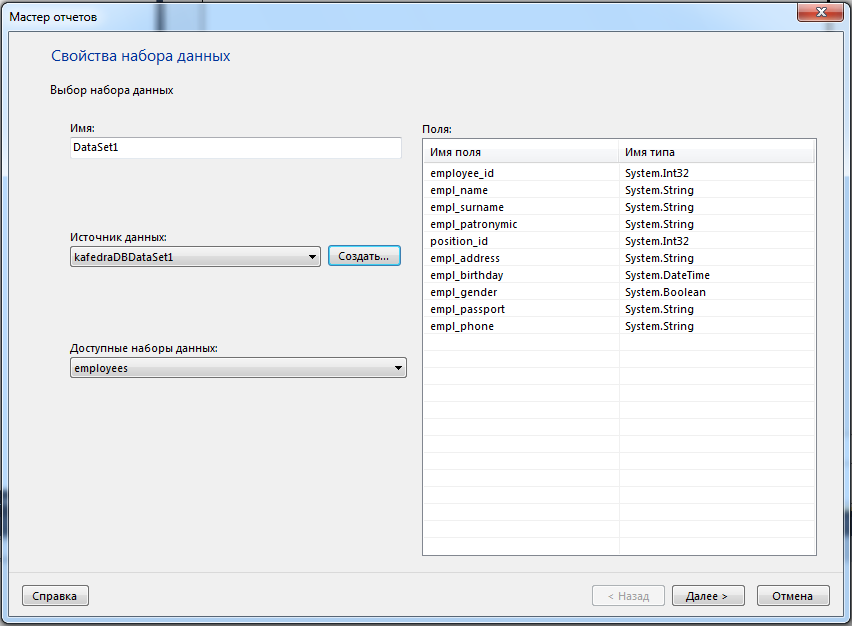


Рисунок 3.2. Окно «Свойства набора данных»

1. В открывшемся окне «Размещение полей» (рисунок 3.3) требуется выбрать строки, столбцы и агрегатные функции.

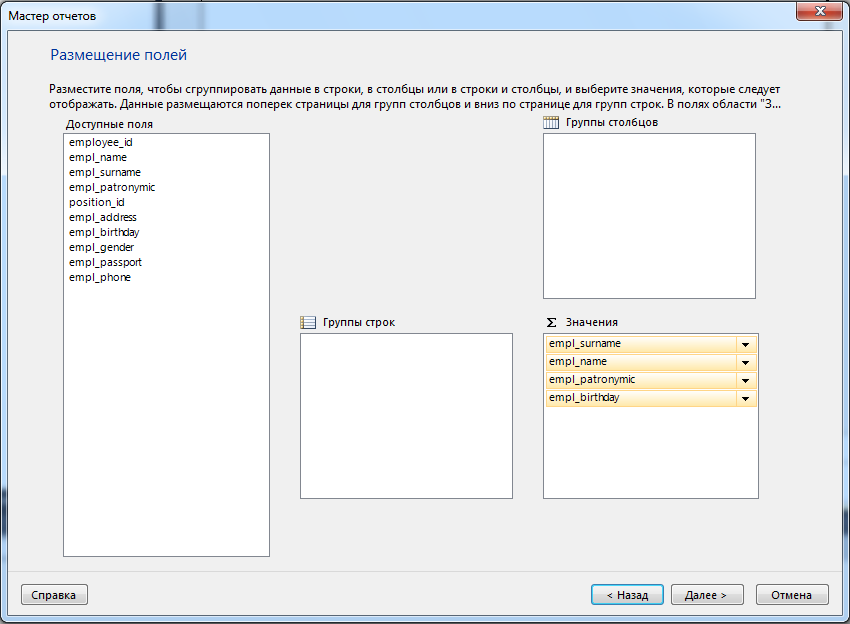


Рисунок 3.3. Окно «Размещение полей»

В поле «Значения» перетаскиваем поляempl\_surname, empl\_name, empl\_patronymic, empl\_birthday. Группы строк и столбцов не создаем, оставляя соответствующие поля пустыми. Нажимаем кнопку «Далее».

1. В открывшемся окне«Выбор макета» (Рисунок 3.4) убираем отметку в поле «Показывать общие итоги и подытоги» и нажимаем кнопку «Далее»

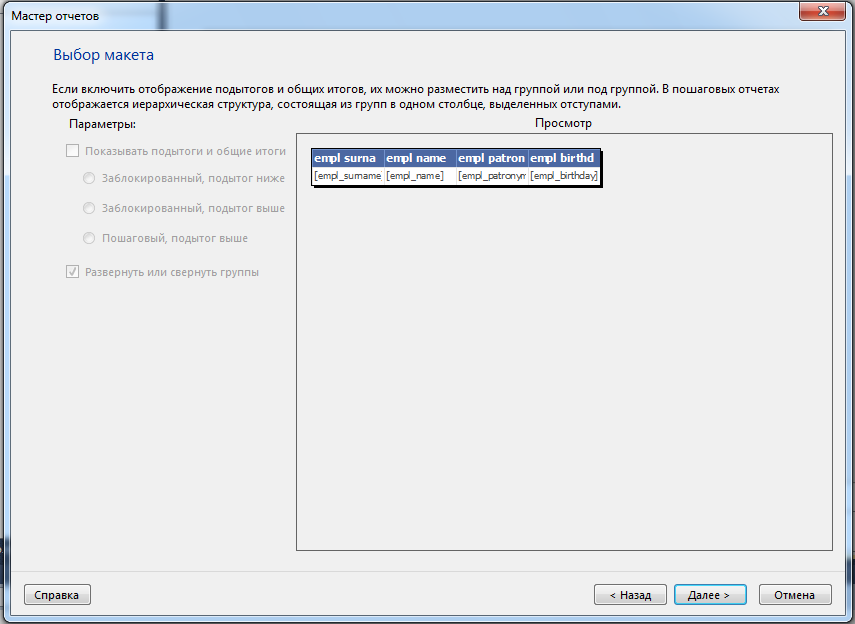


Рисунок 3.4. Окно «Выбор макета»

1. В окне«Выбор стиля» отчетов выбираем стиль и нажимаем кнопку «Готово»
2. В открывшемся окне конструктора отчета (Рисунок 3.5.) отображается структура отчета.

Отчет содержит четыре столбца – empl\_surname, empl\_name, empl\_patrinymic, empl\_birthday.

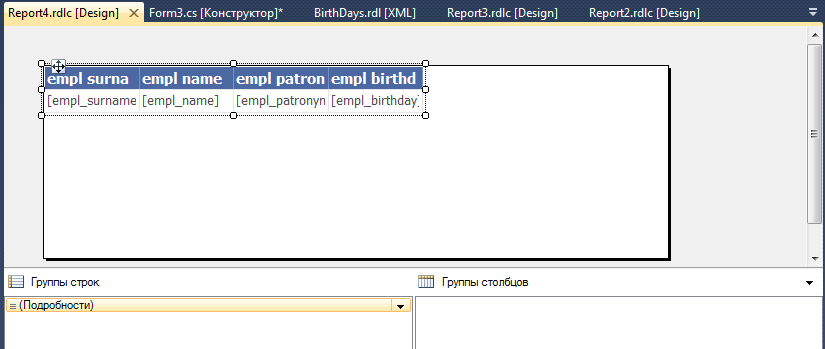


Рисунок 3.5. Конструктор отчета

В конструкторе можно внести изменения в отчет – добавить название отчета в виде текстового поля, изменить названия столбцов, выбрать шрифт и форматы представления данных.

Добавим в табликс первый столбец с функцией =RowNumber(Nothing), в котором будет отображаться номер строки по порядку. Для отображения даты рождения в нужном формате в поле empl\_birthdate вставим выражение =FORMAT(Fields!empl\_birthday.Value, "dd-MM-yyyy")

После внесения изменения отчет в режиме конструктора будет иметь вид, показанный на рисунке 3.6.

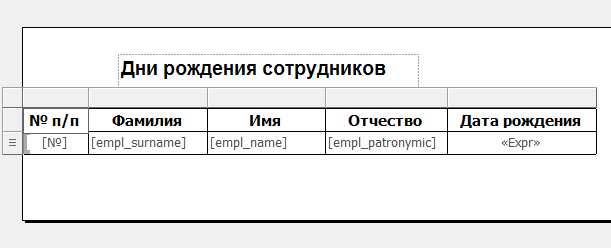


Рисунок 3.6. Отчет в режиме «Конструктор» после редактирования

1. После создания отчета его необходимо связать с элементомReportViewer. Для этого нужно щелкнуть мышкой по треугольнику в верхней правой части ReportViewer и в поле «Выбрать отчет» выбрать имя созданного отчета.

При запуске приложения и открытии формы Form3 отчет будет иметь вид, показаный на рисунке 3.7.

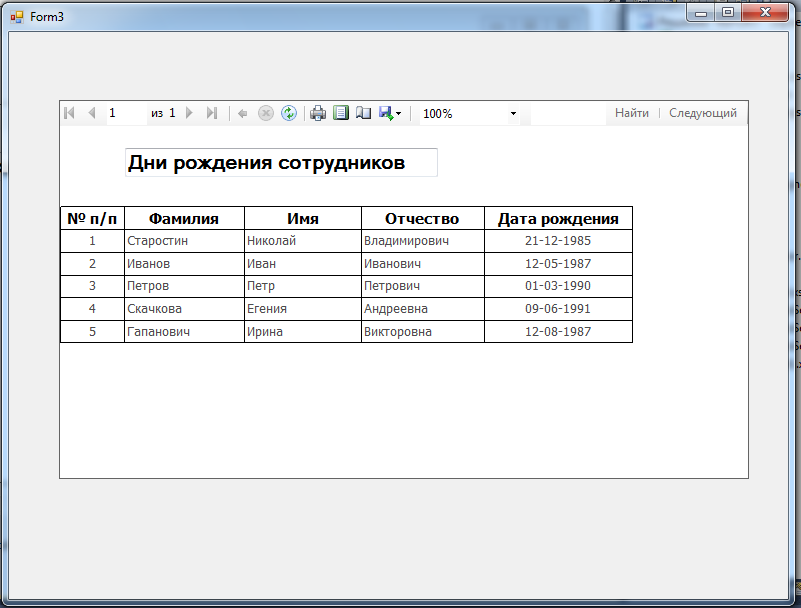


Рисунок 3.7 Отображение данных в элементе ReportViewer

С помощью меню элемента ReportViewer можно отправить данные на принтер или сохранить их в виде файла формата Word или Excel или pdf.

# 3. Создание текстового документа на основе базы данных

Часто требуется создавать обычные текстовые документы (письма, справки и т.п.), в которые требуется вставлять данные из базы данных. Рассмотрим пример создания справки с места работы для выбранного работника. Воспользуемся ранее созданным приложением, в котором выводятся в форму данные о каждом сотруднике (Рисунок 4.1).

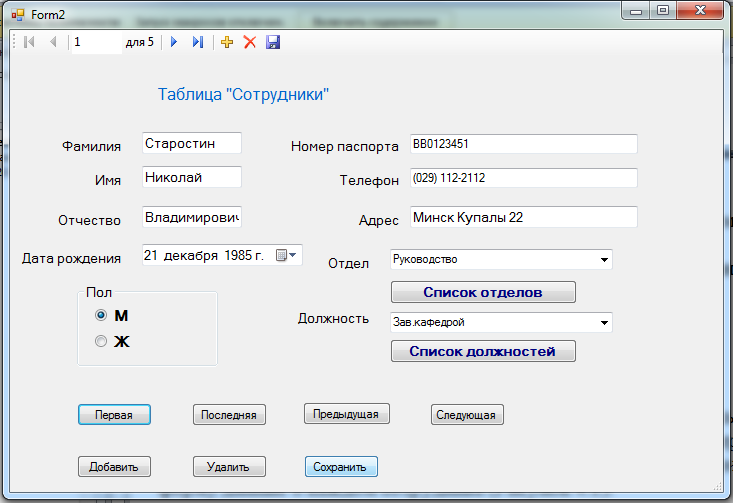


Рисунок 4.1. Форма с данными о сотруднике

Для создания справки с места работы сотрудника нужно выполнить следующие действия.

1. Создать форму и поместить в нее элемент ReportViewer (Form5).
2. В ReportViewer создать новый отчет и поместить в него текстовое поле с названием документа (Справка) и связать созданный отчет с элементом ReportViewer.
3. В форму Form2 поместить кнопку с надписью «Справка с места работы», при нажатии которой будет открываться формаForm5, в которой будет расположен отчет с текстом справки.

Обработчик события «нажатие кнопки» показан ниже.

***private void btCertificate\_Click(object sender, EventArgs e)***

***{***

***Form5 f5 = new Form5(); //то создать форму***

***f5.ShowDialog(this);***

***}***

Свойство Modifiers для полей фамилии, имени и отчества сотрудника установить Public.

1. Создать для формы Form5 обработчик события «Загрузка формы» следующего вида

***private void Form5\_Load(object sender, EventArgs e)***

***{***

***string strFIO;***

***Form2 main = this.Owner as Form2;***

***strFIO = main.empl\_surnameTextBox.Text + " " +***

***main.empl\_nameTextBox.Text + " " +main.empl\_patronymicTextBox.Text;***

***ReportParameterpFIO = new ReportParameter("pFIO", strFIO);***

***reportViewer1.LocalReport.SetParameters(new ReportParameter[] { pFIO });***

***this.reportViewer1.RefreshReport();***

***}***

В коде определен параметр отчета pFIO, содержащий фамилию, имя и отчество сотрудника.

1. В форму Form5 поместить элемент управления ReportViewer и вызвать мастера построения отчетов, в котором выберем источник данных и набор данных employees. В окне «Размещение полей» выберем любое поле. Сформированный таким образом отчет будет иметь вид, показанный на рисунке 4.2.

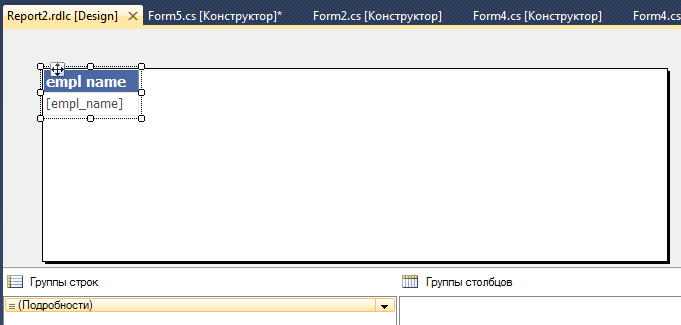


Рисунок 4.2. Отчет в режиме «Конструктор»

1. В отчет нужно добавить параметр. Для этого открыть отчет в режиме «Конструктор», ЩЕЛКНУТЬ ПО НЕМУ МЫШКОЙ . В этом случае в меню Вид (View) появится пункт “Данныe отчета” (ReportData). Выбрать этот пункт. На экране появится панель данных отчета, показанная на рисунке 4.3.

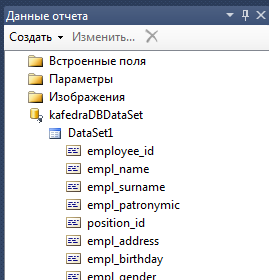


Рисунок 4.3. Панель данных отчета

1. В панели данных отчета щелкнуть правой кнопкой мышки по папке «Параметры» и выбрать «Добавить параметр…». В открывшемся окне свойства параметра отчета выбрать имя параметра (pFIO) и тип данных параметра (Text)
2. Поместить в отчет текстовое поле и набрать в нем текст «СПРАВКА» и удалить табликс
3. Вставить в отчет второе текстовое поле, щелкнуть по нему правой кнопкой мышки и в открывшемся меню выберем «Выражение…». При этом откроется окно построителя выражений, показанное на рисунке 4.3.

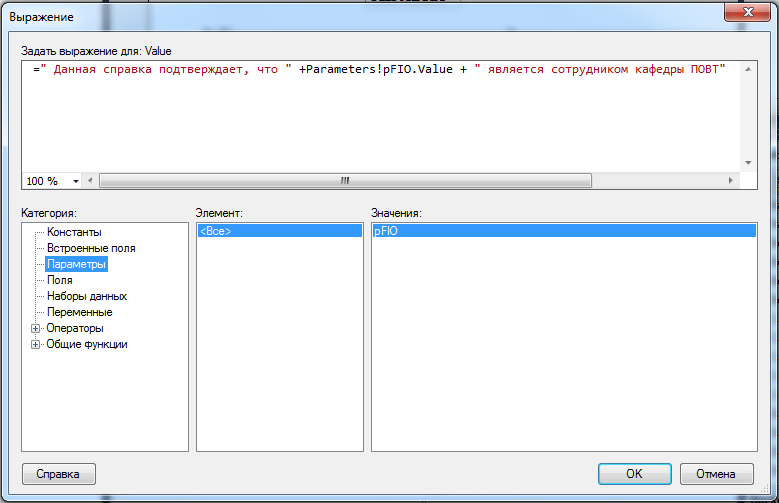


Рисунок 4.3. Окно построителя выражений

1. В окне Построителя выражений сформировать следующее выражение

=” Данная справка подтверждает, что “ + Parameters!pFIO.Value + “ кафедры ПОВТ”

При запуске приложения и выборе сотрудника в форме Form2 и нажатии кнопки “Справка с места работы” будет открыт отчет, показанный на рисунке 4.5

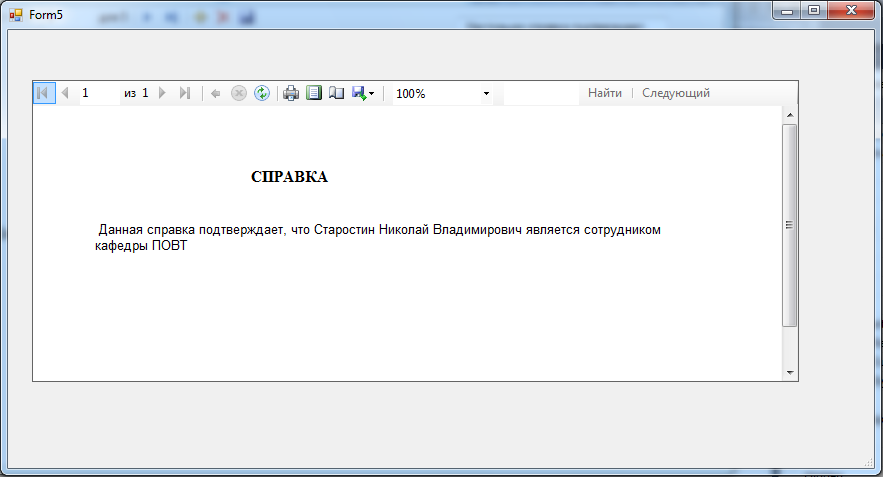


Рисунок 4.5. Сформированный текстовый отчет.

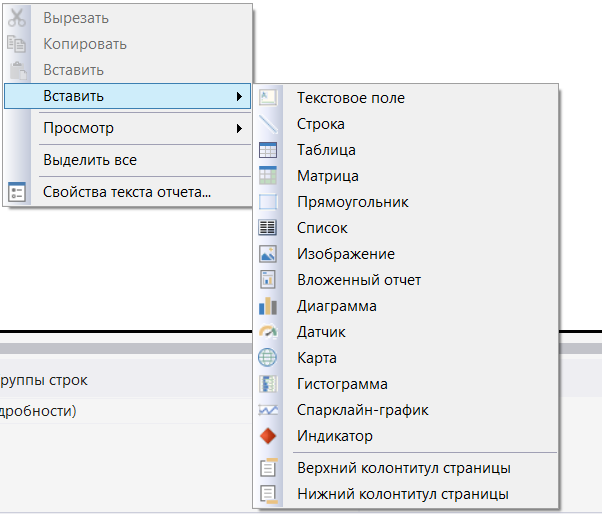
Созданный отчет может быть распечатан или передан в Word.

# 4. Редактирование отчета

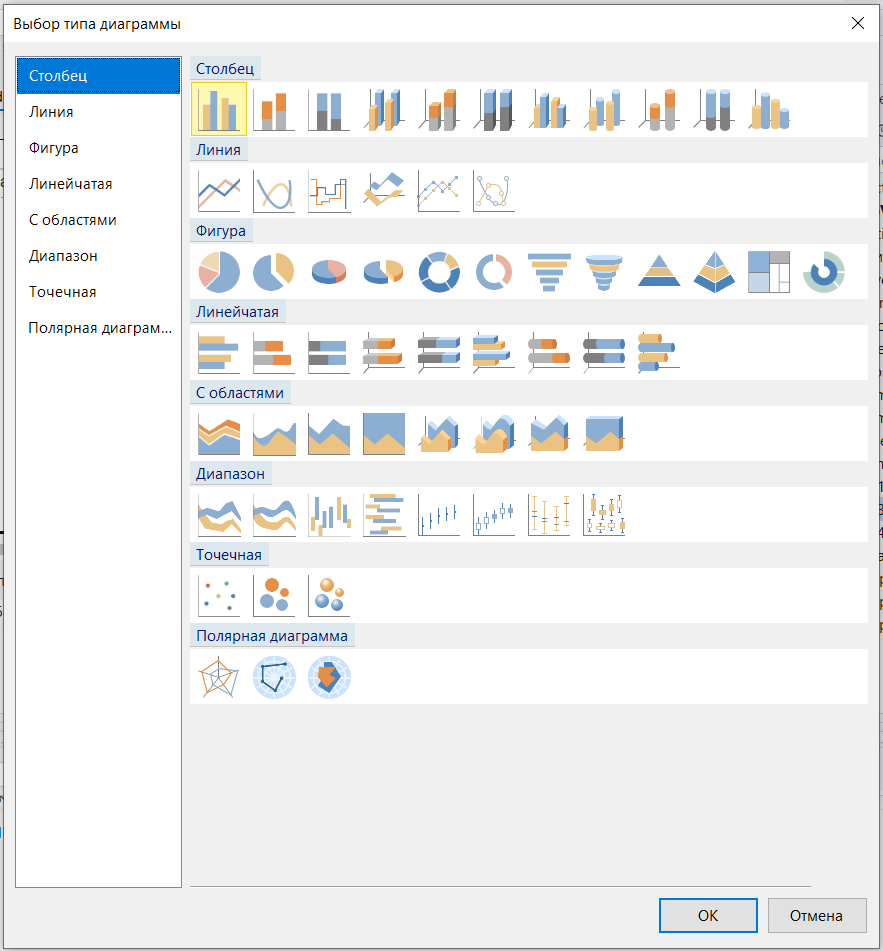
Для редактирования отчета в VisualStudio используется Конструктор отчетов.

Чтобы открыть отчет с помощью Конструктора отчетов нужно дважды щелкнуть ЛКМ по файлу .rdlc и по умолчанию он откроется в Конструкторе отчетов.

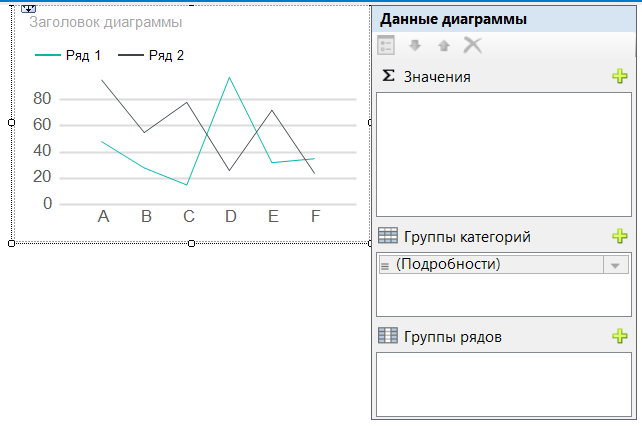
Для добавленияразличных элементов в отчет необходимо щелкнуть ПКМ в поле ReportViewer и в открывшемся меню выбрать пункт «Вставить». При этом откроется список вставляемых объектов, показанный ниже.



На рисунке ниже показан перечень вставляемых диаграмм



При вставке диаграммы можно определить отображаемые данные с помощью встроенных средств как показано ниже.



**Справочный материал**

**1. Структура отчета**

Отчет формируется на основе специальной таблицы - табликса.

Область данных табликса имеет четыре подобласти, которые содержат ячейки табликса:

* угол,
* область группы строк,
* область группы столбцов,
* область данных (текст).

Ячейки в каждой области имеют различное назначение. Добавление ячеек в область данных табликса осуществляется для отображения подробных данных и сгруппированных данных. Конструктор отчетов добавляет ячейки к области группы строк или группы столбцов при создании группы для отображения значений экземпляра группы. В конструкторе отчетов создаются угловые ячейки табликса, если они существуют, группы строк и группы столбцов.

На рисунке 1 показаны рассматриваемые области, относящиеся к области табликса, с вложенными группами строк на основе категории и подкатегории, вложенными группами столбцов на основе территориального расположения страны или региона и смежной группой столбцов на основе года.

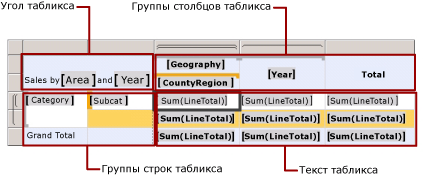


Рисунок 1. Структура табликса

* **Область угла табликса**. (Необязательно) Угол табликса находится в левом верхнем углу. Эта область создается автоматически при добавлении группы строк и группы столбцов к области данных табликса. В этой области можно подвергать слиянию ячейки и добавлять метку или внедрять другой элемент отчета. На рисунке 1 в подвергнутых слиянию ячейках в углу отображается метка «Продажи по регионам и годам» (SalesbyAreaandYear).
* **Область группы столбцов табликса**. (Необязательно) Группы столбцов табликса находятся в правом верхнем углу. Эта область создается автоматически при добавлении группы столбцов. Ячейки в этой области представляют элементы иерархии групп столбцов и отображают значения экземпляра группы столбцов. На этом рисунке ячейки, в которых отображаются [Geography] и [CountryRegion], представляют собой вложенные группы столбцов, а ячейка, в которой отображается [Year], представляет собой смежную группу столбцов. В столбце [Total] отображаются статистические итоги по каждой строке.
* **Область группы строк табликса**. (Необязательно) Группы строк табликса находятся в левом нижнем углу. Эта область создается автоматически при добавлении группы строк. Ячейки в этой области представляют собой элементы иерархии групп строк и отображают значения экземпляра группы строк. На рисунке 1 ячейки, в которых отображаются [Category] и [Subcat], представляют собой вложенные группы строк. Строка «Итог» под обозначением «Подкатегория» повторяется с каждой группой категорий, отображая статистические подытоги для каждого столбца. В строке общего итога показаны итоги по всем категориям.
* **Область данных (текста) табликса**. Текст табликса находится в нижнем правом углу. В тексте табликса отображаются подробные данные и сгруппированные данные. В этом примере используются только статистические данные. В ячейках в тексте табликса отображаются подробные данные, если эти ячейки являются элементами строки с подробными сведениями, и статистические данные, если эти ячейки являются элементами строки или столбца, связанного с группой. По умолчанию вычисление значения в ячейке строки или столбца группы, содержащей простое выражение, которые не включает статистических функций, приводит к получению первого значения в группе. На этом рисунке в ячейках отображаются статистические итоги для итогов строк по всем заказам на продажу.

При выполнении отчета группы столбцов развертываются вправо на такое количество столбцов, которое соответствует количеству уникальных значений для выражения группы. Группы строк развертываются в направлении к нижней части страницы.

На рисунке 2 показан табликс в режиме просмотра.

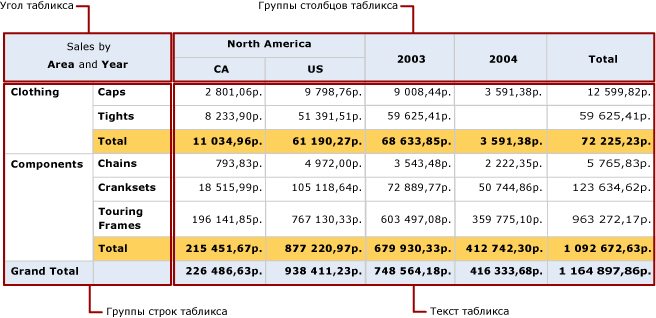


Рисунок 2. Табликс в режиме просмотра

В области группы строк отображаются два экземпляра группы категорий для Clothing и Components. Группа столбцов отображает экземпляр группы территориального расположения для Северной Америки, с двумя вложенными экземплярами группы стран или регионов для Канады (CA) и Соединенных Штатов (US). Кроме того, в смежном столбце отображаются два экземпляра группы года для 2003 года и 2004 года. В строке столбца «Итог» отображаются итоги строки; строка итогов, которая повторяется с каждой группой категорий, показывает итоги по подкатегориям, а строка общего итога показывает итоги по категориям один раз для всей области данных.

Область данных табликса отображает подробные данные в строках и столбцах сведений и сгруппированные данные в строках и столбцах групп. Когда группы строк и группы столбцов добавляются в область данных табликса, автоматически добавляются строки и столбцы, на которых будут отображаться данные. Можно вручную добавить или удалить строки и столбцы, чтобы настроить область данных табликса и определить способ отображения данных в отчете.

# 2. Основные сведения о группах

В конструкторе отчетов группа является именованным набором данных из набора данных отчета, привязанного к области данных. В основном, группа организует представление набора данных отчета. Все группы в области данных задают различные представления одного набора данных отчета.

Чтобы составить наглядное представление о группе, обратитесь к рисунку 3, который показывает область данных табликса в режиме предварительного просмотра. На этом рисунке группы строк классифицируют набор данных по типу продукта, а группы столбцов классифицируют набор данных по географическому региону и году.

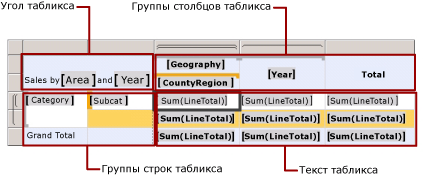
****

Рисунок 3. Область данных табликса

## 2.1. Типы групп, доступные в области данных

Область данных может содержать следующие типы групп:

* **Группа подробностей**Группа подробностей состоит из всех данных из набора данных отчета после применения конструктором отчетов фильтров набора данных и области данных. Таким образом, группа подробностей — единственная, в которой нет выражений группирования.

В сущности, группа подробностей указывает данные, которые можно видеть при выполнении запроса в конструкторе запросов. Например, имеется запрос, который получает все столбцы из таблицы заказов на покупки. Таким образом, данные в этой группе подробностей включают все значения для каждой строки для всех столбцов в таблице. Данные в этой группе подробностей также включают значения для любых вычисляемых полей набора данных, которые были созданы.

По умолчанию, при добавлении к отчету таблицы или списка конструктор отчетов автоматически создает группу подробностей и добавляет строку для отображения подробных данных. По умолчанию, при добавлении полей набора данных к ячейкам в этой строке отображаются простые выражения для полей, например, [Продажи]. При просмотре области данных строка подробностей повторяется один раз для каждого значения набора результатов.

* **Группы строк и группы столбцов** Можно организовать данные в группы по строкам и столбцам. Группы строк расширяются вертикально на странице. Группы столбцов расширяются горизонтально на странице. Группы могут быть вложенными, например, сгруппированными сначала по [Год], затем по [Квартал], затем по [Месяц]. Группы также могут быть смежными, например, сгруппированными по [Территория] и независимо по [КатегорияПродукта].

При создании группы для области данных конструктор отчетов автоматически добавляет строки и столбцы в область данных и использует эти строки или столбцы для отображения данных группы.

* **Группы рекурсивной иерархии** Группы рекурсивной иерархии организуют данные из одного набора данных отчета, который содержит несколько уровней. Например, группа рекурсивной иерархии может отобразить иерархию организации, например, [Сотрудник], который отчитывается перед [Сотрудник].

Ячейки в строке или столбце области данных табликса могут принадлежать к нескольким группам строк или столбцов. При определении выражения в текстовом поле ячейки, которая использует статистическую функцию (например, =Sum(Fields!FieldName.Value), по умолчанию область группы для ячейки — самая внутренняя дочерняя группа, к которой она принадлежит. Если ячейка принадлежит как группам строк, так и столбцов, то областью являются обе самые внутренние группы. Можно также записать выражения, которые вычисляют статистические подытоги, отнесенные к группе относительно другого набора данных. Например, можно вычислить процент группы относительно группы столбца или всех данных области данных (такой как =Sum(Fields!FieldName.Value)/Sum(Fields!FieldName.Value,"ColumnGroup")).

Группа имеет имя и набор выражений группирования, заданных пользователем. Набор выражений группирования может быть одной ссылкой на поле набора данных или комбинацией нескольких выражений. Во время выполнения конструктор отчетов объединяет и применяет выражения группирования к данным в группе. Например, имеется группа, которая использует поле дата для организации данных в области данных. Во время выполнения конструктор отчетов сначала упорядочивает данные по дате, а затем отображает и суммирует другие значения набора данных для каждой даты.В большинстве случаев конструктор отчетов автоматически создает группу при проектировании области данных. Для таблицы, матрицы или списка группы создаются при переносе полей на панель группирования. Для диаграмм группы создаются при переносе полей в зоны перетаскивания. Для датчика необходимо использовать диалоговое окно свойств датчика. Для таблицы, матрицы или списка можно создать группу вручную.

## Примеры выражений группы

В следующей таблице приведены примеры выражений группы, которые можно использовать для определения группы.

| **Description** | **Выражение** |
| --- | --- |
| Группирование по полю Region . | =Fields!Region.Value |
| Группирование по фамилии и имени. | =Fields!LastName.Value   =Fields!FirstName.Value |
| Группирование по первой букве фамилии. | =Fields!LastName.Value.Substring(0,1) |
| Группирование по параметру, основанному на выборе пользователя.  В этом примере параметр GroupBy должен быть основан на списке допустимых значений, которые предоставляет допустимый вариант выбора для группирования. | =Fields(Parameters!GroupBy.Value).Value |
| Группирование по трем отдельным возрастным диапазонам:  до 20, между 21 и 50, старше 50. | =IIF(First(Fields!Age.Value)<21,"Under 21",(IIF(First(Fields!Age.Value)>=21 AND First(Fields!Age.Value)<=50,"Between 21 and 50","Over 50"))) |
| Группирование по многим возрастным диапазонам. В этом примере показан пользовательский код, написанный на языке VisualBasic .NET, который возвращает строку, относящуюся к одному из следующих диапазонов:  от 25 и младше,  от 26 до 50;  от 51 до 75;  старше 75. | =Code.GetRangeValueByAge(Fields!Age.Value)  Пользовательскийкод (VB):   Function GetRangeValueByAge(ByVal age As Integer) As String  Select Case age  Case 0 To 25  GetRangeValueByByAge = "25 or Under"  Case 26 To 50  GetRangeValueByByAge = "26 to 50"  Case 51 to 75  GetRangeValueByByAge = "51 to 75"  Case Else  GetRangeValueByByAge = "Over 75"  End Select  Return GetRangeValueByByAge  End Function |

## Основные подсказки табликса

Визуальные подсказки в области данных табликса помогут отобразить в ней необходимые данные.

### Маркеры строк и столбцов

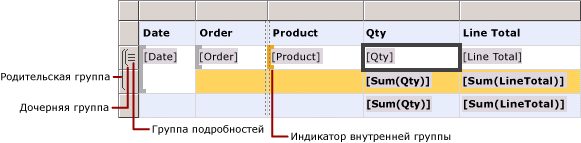
При выборе области данных табликса маркеры строк и столбцов определяют их назначение. Маркеры указывают, входит ли строка или столбец в группу. В следующей таблице показаны различные варианты отображения маркера.

|  |  |
| --- | --- |
| **Значок** | **Описание** |
| Маркер строки с 3 параллельными линиями для строки детализации | Только группа сведений в иерархии групп строк. |
| Маркер строки со строкой детализации и одной внешней группой | Одна внешняя группа и дочерняя группа сведений. |
| Две параллельных скобки, обозначающих вложенные группы | Одна внешняя группа, одна внутренняя группа, группы сведений нет. |
| 2 скобки и 3 линии друг над другом, обозначающие вложенные группы и подробные сведения | Одна внешняя группа, одна внутренняя группа и одна дочерняя группа сведений. |
| Одна внешняя группа со строкой нижнего колонтитула, одна внутренняя группа | Одна внешняя группа со строкой нижнего колонтитула для итогов и одна внутренняя группа. |
| скобка внешней группы, скобка внутренней группы, подробности | Одна внешняя группа со строкой нижнего колонтитула для итогов, одна внутренняя группа со строкой нижнего колонтитула для итогов и одна строка сведений. |
| родительские верхний и нижний колонтитулы, а также дочерняя группа | Одна внешняя группа с верхним колонтитулом для меток и нижним колонтитулом для итогов, внутренняя группа, группы сведений нет. |

### Строки группы

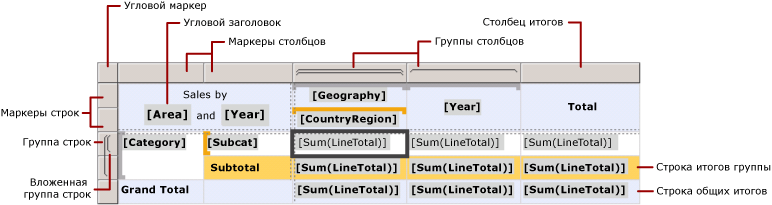
Строки внутри группы повторяются по одному разу для каждого значения в группе и обычно используются для статистических сводок. Строки за пределами группы повторяются в группе один раз и используются для меток и подитогов. При выборе ячейки маркеры и квадратные скобки столбца и строки в области данных табликса показывают группы, в которые входит эта ячейка. На рисунке показаны следующие визуальные подсказки.

1. Маркеры строки и столбца, которые показывают связи с группами.
2. Выделенные индикаторы групп, которые показывают самую внутреннюю группу, в которую входит выбранная ячейка.
3. Индикаторы групп, которые показывают все группы, в которые входит выбранная ячейка.



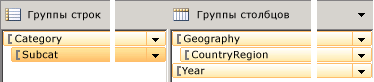
### Итоговые строки

После добавления групп строк и столбцов можно добавить строку для отображения итогов по столбцам или столбец для отображения итогов по строкам. На следующем рисунке показана матрица, содержащая группы строк и столбцов, а также итоговую строку и итоговый столбец.



### Область группирования

В области группирования отображаются группы строк и столбцов для текущей выделенной области данных табликса в области конструктора. На следующем рисунке показана область группирования для области данных табликса.



В области «Группы строк» содержится родительская группа «Category» и дочерняя группа «Subcat». Область «Группы столбцов» содержит родительскую группу «Geography», дочернюю группу «CountryRegion» и группу «Year», которая является смежной для группы «Geography». При выборе в области «Группы строк» группы «Subcat» она становится темно-оранжевой, а ячейка соответствующей группы в области конструктора выделяется.

## Отображение данных в строках и столбцах

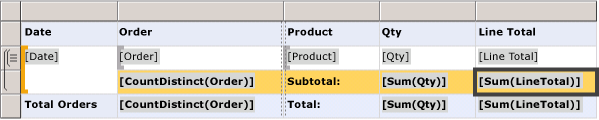
Связь групп строк со строками и связь групп столбцов со столбцами подчиняются одним и тем же правилам.

Для каждой строки в области данных табликса строка по отношению к каждой из групп строк находится либо внутри, либо снаружи. Если строка находится внутри группы строк, то она повторяется один раз для каждого уникального значения группы, которое называется *экземпляром группы*. Если строка находится за пределами группы строк, то она повторяется в этой группе всего один раз. Строки, расположенные за пределами всех групп, являются статическими и повторяются всего один раз в пределах области данных. Например, строка верхнего или нижнего колонтитула таблицы является статической. Строки, которые повторяются хотя бы в одной из групп, являются динамическими.

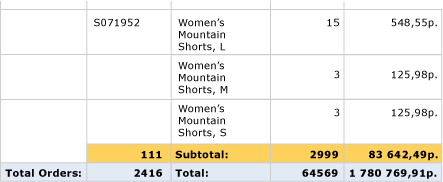
При наличии вложенных групп строка может находиться внутри родительской группы, но за пределами дочерней. Такая строка повторяется для каждого значения внешней группы, однако в дочерней группе отображается всего один раз. Чтобы отображались метки или итоги группы, добавьте строку за пределами группы. Чтобы отображались данные, которые отличаются для каждого экземпляра группы, добавьте строку внутри группы.

При наличии группы сведений все строки сведений находятся внутри этой группы. Эти строки повторяются для каждого значения результирующего набора.

На следующем рисунке показана область данных табликса с вложенными группами строк и группой сведений.



Для области данных табликса, в которой содержатся подробные данные, группа сведений является самой внутренней дочерней группой. Строки, добавленные в группу сведений, повторяются один раз на каждую строку результирующего набора для запроса к набору данных, связанного с данной областью данных табликса. На следующем рисунке показана последняя страница отчета, готового для просмотра. На нем видны последние строки сведений и строка промежуточных итогов для последнего заказа.



Все столбцы области данных табликса подчиняются одним и тем же правилам. Например, по отношению к каждой из групп столбец может находиться либо внутри, либо снаружи. Строку итогов необходимо добавлять за пределами группы.

Чтобы удалить строки и столбцы, связанные с группой, можно удалить всю группу. При удалении группы можно либо удалить только ее определение, либо удалить всю группу вместе со связанными строками и столбцами. По умолчанию удаляется только группа, а макет строк и столбцов в области данных остается без изменений. При удалении группы со всеми связанными строками и столбцами удаляются все статические (включая верхние и нижние колонтитулы) и динамические строки и столбцы (включая экземпляры групп), связанные с этой группой.

## Основные сведения о ячейках табликса

Ячейки табликса относятся к одной из четырех его областей: тексту, области группы строк, области группы столбцов или углу. Несмотря на то, что ячейка может отображать любое значение набора данных, функция по умолчанию определяется по ее местоположению. По умолчанию ячейки в областях групп строк и столбцов табликса представляют элементы группы. Эти элементы упорядочены в определении отчета в несколько древовидных структур. Иерархия группы строк расширяется по горизонтали. Иерархия группы столбцов расширяется по вертикали. Ячейки добавляются автоматически при создании группы и отображают для группы уникальные значения во время выполнения.

Ячейки в углу табликса создаются в том случае, когда существует и область группы строк, и область группы столбцов. Ячейки в этой области можно объединить, создав метку или внедрив другой элемент отчета.

Ячейки в области текста табликса могут отображать сведения, если ячейка находится в строке или столбце сведений, или статистические данные группы, если ячейка находится в группе строк или столбцов. Область видимости данных в ячейке определяется пересечением самой внутренней группы строк и самой внутренней группы столбцов, в которые входит ячейка.

### Объединение и разбиение ячеек

В области табликса несколько смежных ячеек можно объединить в одну. Это дает возможность, например, создавать ячейки для меток, занимающие несколько строк или столбцов.

В области угла табликса несколько ячеек могут объединяться только в одном направлении за один раз: либо по горизонтали (на несколько столбцов), либо по вертикали (на несколько строк). Чтобы объединить блок ячеек, сначала их необходимо объединять по горизонтали. Объединив нужное число ячеек в каждой из строк, выделите смежные ячейки по вертикали (можно выделить весь столбец) и объедините их.

В области текста табликса ячейки могут быть объединены только по горизонтали. Объединение ячеек по вертикали не поддерживается.

Ранее объединенные ячейки можно разбить. Разбиение ячеек может быть выполнено либо по горизонтали (на несколько столбцов), либо по вертикали (на несколько строк). Чтобы разбить ячейку на блок ячеек, сначала необходимо разбивать их по горизонтали, а затем каждую из получившихся строк — по вертикали.

# Работа с областями данных

Область данных служит для отображения данных из набора данных отчета. Можно перетаскивать области данных из области элементов в область конструктора. В области данных можно группировать, сортировать, фильтровать данные из единичного набора данных и выполнять их статистическую обработку. В отчет можно помещать несколько областей данных. Области данных можно поместить рядом друг с другом в тексте отчета в прямоугольном контейнере или вложить одни области данных в другие. Несколько областей данных создают различные представления одного и того же набора данных или сходные представления разных наборов данных.

## Типы областей данных

Выбранный тип области данных будет зависеть от объема данных, диапазона значений и цели отчета. В следующем списке описываются функциональные возможности всех областей данных.

* **Таблица**. В таблице выводятся данные подробностей или данные, сгруппированные по строкам. Таблица имеет фиксированное число столбцов. Таблица расширяется по странице вниз, насколько требуется. Можно вывести все данные подробностей, строка за строкой, или сгруппировать данные, создав группы строк. Группы строк могут быть вложенными или смежными. Группа строк выводит динамическую строку по странице вниз для каждого значения в группе, вычисляемого во время выполнения. Можно добавить статические строки для меток или итогов. К таблице или отдельным группам можно добавить итоги.
* **Матрица**. В матрице данные группируются по строкам и столбцам. Матрицы обеспечивают функциональность, подобную перекрестным и сводным динамическим представлениям. Матрица должна иметь как минимум одну группу строк и одну группу столбцов. Матрица расширяется поперек страницы для групп столбцов и вниз по странице для групп строк. В клетках матрицы выводятся сводки и статистические значения, областью действия которых являются пересечения групп строк и столбцов. Можно создавать дополнительные вложенные и смежные группы. Число строк и столбцов матрицы зависит от значений в каждой группе, определяемых во время выполнения. Можно добавить статические строки для меток или итогов. К таблице или отдельным группам можно добавить итоги.
* **Список**. Данные в списке выводятся в свободной форме. Например, список можно использовать для создания формы или вывода одновременно таблицы и диаграммы. Чтобы создать макет списка, надо расположить текстовые поля в любом его месте. Строка списка повторяется по одному разу для каждого значения в наборе данных.
* **Диаграмма**. Диаграмма представляет данные в графическом виде. Диаграммы дают возможность визуально представить сводные и статистически обработанные данные.
* **Датчик**. Датчик представляет индикатор, указывающий на конкретное значение в пределах диапазона. Датчики можно использовать в таблице или матрице для вывода относительного значения поля в диапазоне значений области данных — например, в виде KPI. Датчик можно также добавить в область конструктора, чтобы показать одно относительное значение. Индикаторами являются минимальные датчики, обеспечивающие возможность быстрого определения состояния одиночного значения данных.
* **Карта**. Карта отображает данные в географическом контексте. Карты могут использоваться для визуального представления сводных и статистически обработанных данных для каждого элемента карты, например области, дороги или города. Можно визуально отобразить несколько наборов данных, добавив несколько слоев карты, управляемых независимо.

***(для Web-Forms***

***Install-Package Microsoft.ReportingServices.ReportViewerControl.WebForms -Version 150.1404.0)***

**УстановкаReportViewerвWeb-Forms**

# (ИспользованиеэлементауправленияWebFormsReportViewer)

Для просмотра отчетов, развернутых на сервере отчетов, и отчетов, существующих в локальной файловой системе, можно использовать элемент управления WebFormsReportViewer для подготовки отчетов к просмотру в веб-приложении.

Для использования ReporViewer в ASP-проекте нужно выполнить следующие действия

1. Создать веб-сайт Microsoft ASP.NET с помощью Microsoft Visual C# или Microsoft Visual Basic.

- - или -

Открыть существующий веб-сайт ASP.NET и добавить в него новую веб-форму.

1. Найти элемент управления **ScriptManager** в группе **AJAX-расширения** в окне **Панель элементов** и перетащить его в область конструктора веб-формы.

Если окно **Панель элементов** не отображается, к нему можно получить доступ из меню **Вид**, выбрав пункт **Панель элементов**.

1. Найти элемент управления **ReportViewer** в окне **Панель элементов** и перетащите его в область конструктора под элементом управления **ScriptManager**.

В форму будет добавлен элемент управления **ReportViewer** с именем reportViewer1.

## Просмотр отчетов в режиме удаленной обработки

В следующем примере показана подготовка к просмотру отчета, развернутого на сервере отчетов. В этом примере используется отчет SalesOrderDetail, который включен в проект с образцами отчетов **AdventureWorks2012**.

В этом примере используется встроенная проверка подлинности Windows, поэтому сначала необходимо включить олицетворение. Для этого в файл **web.config** необходимо вставить следующую строку:

Копировать

<!--Web.config file. -->

<identity impersonate="true"/>

**Примечание**

По умолчанию олицетворение отключено.

**C#**

Копировать

protected void Page\_Init(object sender, EventArgs e)

{

if (!Page.IsPostBack)

{

// Set the processing mode for the ReportViewer to Remote

reportViewer.ProcessingMode = ProcessingMode.Remote;

ServerReportserverReport = reportViewer.ServerReport;

// Set the report server URL and report path

serverReport.ReportServerUrl =

new Uri("https://<Server Name>/reportserver");

serverReport.ReportPath =

"/AdventureWorks Sample Reports/Sales Order Detail";

// Create the sales order number report parameter

ReportParametersalesOrderNumber = new ReportParameter();

salesOrderNumber.Name = "SalesOrderNumber";

salesOrderNumber.Values.Add("SO43661");

// Set the report parameters for the report

reportViewer.ServerReport.SetParameters(

new ReportParameter[] { salesOrderNumber });

}

}

**VB.NET**

Копировать

Imports Microsoft.Reporting.WebForms

Partial Class \_Default

Inherits System.Web.UI.Page

Protected Sub Page\_Init(ByVal sender As Object, \_

ByVal e As System.EventArgs) Handles Me.Init

If Not Page.IsPostBack Then

'Set the processing mode for the ReportViewer to Remote

reportViewer.ProcessingMode = ProcessingMode.Remote

Dim serverReport As ServerReport

serverReport = reportViewer.ServerReport

'Set the report server URL and report path

serverReport.ReportServerUrl = \_

New Uri("https://<Server Name>/reportserver")

serverReport.ReportPath = \_

"/AdventureWorks Sample Reports/Sales Order Detail"

'Create the sales order number report parameter

Dim salesOrderNumber As New ReportParameter()

salesOrderNumber.Name = "SalesOrderNumber"

salesOrderNumber.Values.Add("SO43661")

'Set the report parameters for the report

Dim parameters() As ReportParameter = {salesOrderNumber}

serverReport.SetParameters(parameters)

End If

End Sub

EndClass

## Просмотр отчетов в режиме локальной обработки

В следующем примере показана подготовка к просмотру отчета, который является частью приложения Windows и не был помещен на сервер отчетов.

###### Добавление отчета «Сведения о заказе на продажу» на веб-сайт

1. Откройте веб-сайт, на который должен быть добавлен отчет.
2. В меню **Веб-сайт** выберите пункт **Добавить существующий элемент**.
3. Перейдите в папку, в которой были установлены проекты образцов отчетов AdventureWorks.

Ееместонахождениемпоумолчаниюявляется C:\Program Files\Microsoft SQL Server\100\Samples\Reporting Services\Report Samples\AdventureWorks Sample Reports.

1. Выберитефайл Sales Order Detail.rdlинажмитекнопку **Добавить**.

Файл «Сведения о заказе на продажу.rdl» теперь станет частью проекта.

1. Щелкните правой кнопкой мыши файл SalesOrderDetail.rdl в обозревателе решений и выберите команду **Переименовать**. Переименуйте отчет в **Сведения о заказах на продажу.rdlc** и нажмите клавишу ВВОД.

Если обозреватель решений не отображается, его можно открыть из меню **Вид**, выбрав пункт "Обозреватель решений".

В следующем примере кода создается набор данных для данных о заказе на продажу, а затем подготавливается к просмотру отчет «Сведения о заказе на продажу» в локальном режиме.

**C#**

C#Копировать

protectedvoidPage\_Init(object sender, EventArgs e)

{

if(!Page.IsPostBack)

{

// Set the processing mode for the ReportViewer to Local

reportViewer.ProcessingMode = ProcessingMode.Local;

LocalReportlocalReport = reportViewer.LocalReport;

localReport.ReportPath = "Sales Order Detail.rdlc";

DataSet dataset = newDataSet("Sales Order Detail");

stringsalesOrderNumber = "SO43661";

GetSalesOrderData(salesOrderNumber, ref dataset);

ReportDataSourcedsSalesOrder = newReportDataSource();

dsSalesOrder.Name = "SalesOrder";

dsSalesOrder.Value = dataset.Tables["SalesOrder"];

localReport.DataSources.Add(dsSalesOrder);

GetSalesOrderDetailData(salesOrderNumber, ref dataset);

ReportDataSourcedsSalesOrderDetail = newReportDataSource();

dsSalesOrderDetail.Name = "SalesOrderDetail";

dsSalesOrderDetail.Value = dataset.Tables["SalesOrderDetail"];

localReport.DataSources.Add(dsSalesOrderDetail);

// Create the sales order number report parameter

ReportParameterrpSalesOrderNumber = newReportParameter();

rpSalesOrderNumber.Name = "SalesOrderNumber";

rpSalesOrderNumber.Values.Add("SO43661");

// Set the report parameters for the report

localReport.SetParameters(

newReportParameter[] { rpSalesOrderNumber });

}

}

privatevoidGetSalesOrderData(stringsalesOrderNumber,

refDataSetdsSalesOrder)

{

stringsqlSalesOrder =

"SELECT SOH.SalesOrderNumber, S.Name AS Store, " +

" SOH.OrderDate, C.FirstName AS SalesFirstName, " +

" C.LastName AS SalesLastName, E.Title AS " +

" SalesTitle, SOH.PurchaseOrderNumber, " +

" SM.Name AS ShipMethod, BA.AddressLine1 " +

" AS BillAddress1, BA.AddressLine2 AS " +

" BillAddress2, BA.City AS BillCity, " +

" BA.PostalCode AS BillPostalCode, BSP.Name " +

" AS BillStateProvince, BCR.Name AS " +

" BillCountryRegion, SA.AddressLine1 AS " +

" ShipAddress1, SA.AddressLine2 AS " +

" ShipAddress2, SA.City AS ShipCity, " +

" SA.PostalCode AS ShipPostalCode, SSP.Name " +

" AS ShipStateProvince, SCR.Name AS " +

" ShipCountryRegion, CC.Phone AS CustPhone, " +

" CC.FirstName AS CustFirstName, CC.LastName " +

" AS CustLastName " +

"FROM Person.Address SA INNER JOIN " +

" Person.StateProvince SSP ON " +

" SA.StateProvinceID = SSP.StateProvinceID " +

" INNER JOIN Person.CountryRegion SCR ON " +

" SSP.CountryRegionCode = SCR.CountryRegionCode " +

" RIGHT OUTER JOIN Sales.SalesOrderHeader SOH " +

" LEFT OUTER JOIN Person.Contact CC ON " +

" SOH.ContactID = CC.ContactID LEFT OUTER JOIN" +

" Person.Address BA INNER JOIN " +

" Person.StateProvince BSP ON " +

" BA.StateProvinceID = BSP.StateProvinceID " +

" INNER JOIN Person.CountryRegion BCR ON " +

" BSP.CountryRegionCode = " +

" BCR.CountryRegionCode ON SOH.BillToAddressID " +

" = BA.AddressID ON SA.AddressID = " +

" SOH.ShipToAddressID LEFT OUTER JOIN " +

" Person.Contact C RIGHT OUTER JOIN " +

" HumanResources.Employee E ON C.ContactID = " +

" E.ContactID ON SOH.SalesPersonID = " +

" E.EmployeeID LEFT OUTER JOIN " +

" Purchasing.ShipMethod SM ON SOH.ShipMethodID " +

" = SM.ShipMethodID LEFT OUTER JOIN Sales.Store" +

" S ON SOH.CustomerID = S.CustomerID " +

"WHERE (SOH.SalesOrderNumber = @SalesOrderNumber)";

SqlConnection connection = new

SqlConnection("Data Source=(local); " +

"Initial Catalog=AdventureWorks; " +

"Integrated Security=SSPI");

SqlCommand command =

newSqlCommand(sqlSalesOrder, connection);

command.Parameters.Add(

newSqlParameter("SalesOrderNumber",

salesOrderNumber));

SqlDataAdaptersalesOrderAdapter = new

SqlDataAdapter(command);

salesOrderAdapter.Fill(dsSalesOrder, "SalesOrder");

}

privatevoidGetSalesOrderDetailData(stringsalesOrderNumber,

refDataSetdsSalesOrder)

{

stringsqlSalesOrderDetail =

"SELECT SOD.SalesOrderDetailID, SOD.OrderQty, " +

" SOD.UnitPrice, CASE WHEN " +

" SOD.UnitPriceDiscount IS NULL THEN 0 " +

" ELSE SOD.UnitPriceDiscount END AS " +

" UnitPriceDiscount, SOD.LineTotal, " +

" SOD.CarrierTrackingNumber, " +

" SOD.SalesOrderID, P.Name, P.ProductNumber " +

"FROM Sales.SalesOrderDetail SOD INNER JOIN " +

" Production.Product P ON SOD.ProductID = " +

" P.ProductID INNER JOIN " +

" Sales.SalesOrderHeader SOH ON " +

" SOD.SalesOrderID = SOH.SalesOrderID " +

"WHERE (SOH.SalesOrderNumber = @SalesOrderNumber) " +

"ORDER BY SOD.SalesOrderDetailID";

using (SqlConnection connection = new

SqlConnection("Data Source=(local); " +

"Initial Catalog=AdventureWorks; " +

"Integrated Security=SSPI"))

{

SqlCommand command =

newSqlCommand(sqlSalesOrderDetail, connection);

command.Parameters.Add(

newSqlParameter("SalesOrderNumber",

salesOrderNumber));

SqlDataAdaptersalesOrderDetailAdapter = new

SqlDataAdapter(command);

salesOrderDetailAdapter.Fill(dsSalesOrder,

"SalesOrderDetail");

}

}

**VB.NET**

VBКопировать

ImportsSystem.Data

ImportsSystem.Data.SqlClient

ImportsMicrosoft.Reporting.WebForms

PartialClass \_Default

InheritsSystem.Web.UI.Page

ProtectedSubPage\_Init(ByVal sender AsObject, \_

ByVal e AsSystem.EventArgs) HandlesMe.Init

IfNotPage.IsPostBackThen

'Set the processing mode for the ReportViewer to Local

reportViewer.ProcessingMode = ProcessingMode.Local

DimlocalReportAsLocalReport

localReport = reportViewer.LocalReport

localReport.ReportPath = "Sales Order Detail.rdlc"

Dim dataset AsNewDataSet("Sales Order Detail")

DimsalesOrderNumberAsString = "SO43661"

'Get the sales order data

GetSalesOrderData(salesOrderNumber, dataset)

'Create a report data source for the sales order data

DimdsSalesOrderAsNewReportDataSource()

dsSalesOrder.Name = "SalesOrder"

dsSalesOrder.Value = dataset.Tables("SalesOrder")

localReport.DataSources.Add(dsSalesOrder)

'Get the sales order detail data

GetSalesOrderDetailData(salesOrderNumber, dataset)

'Create a report data source for the sales

'order detail data

DimdsSalesOrderDetailAsNewReportDataSource()

dsSalesOrderDetail.Name = "SalesOrderDetail"

dsSalesOrderDetail.Value = \_

dataset.Tables("SalesOrderDetail")

localReport.DataSources.Add(dsSalesOrderDetail)

'Create a report parameter for the sales order number

DimrpSalesOrderNumberAsNewReportParameter()

rpSalesOrderNumber.Name = "SalesOrderNumber"

rpSalesOrderNumber.Values.Add("SO43661")

'Set the report parameters for the report

Dimparameters() AsReportParameter = {rpSalesOrderNumber}

localReport.SetParameters(parameters)

EndIf

EndSub

PrivateSubGetSalesOrderData(ByValsalesOrderNumberAsString, \_

ByRefdsSalesOrderAsDataSet)

DimsqlSalesOrderAsString = \_

"SELECT SOH.SalesOrderNumber, S.Name AS Store, "& \_

" SOH.OrderDate, C.FirstName AS SalesFirstName, "& \_

" C.LastName AS SalesLastName, E.Title AS "& \_

" SalesTitle, SOH.PurchaseOrderNumber, "& \_

" SM.Name AS ShipMethod, BA.AddressLine1 "& \_

" AS BillAddress1, BA.AddressLine2 AS "& \_

" BillAddress2, BA.City AS BillCity, "& \_

" BA.PostalCode AS BillPostalCode, BSP.Name "& \_

" AS BillStateProvince, BCR.Name AS "& \_

" BillCountryRegion, SA.AddressLine1 AS "& \_

" ShipAddress1, SA.AddressLine2 AS "& \_

" ShipAddress2, SA.City AS ShipCity, "& \_

" SA.PostalCode AS ShipPostalCode, SSP.Name "& \_

" AS ShipStateProvince, SCR.Name AS "& \_

" ShipCountryRegion, CC.Phone AS CustPhone, "& \_

" CC.FirstName AS CustFirstName, CC.LastName "& \_

" AS CustLastName "& \_

"FROM Person.Address SA INNER JOIN "& \_

" Person.StateProvince SSP ON "& \_

" SA.StateProvinceID = SSP.StateProvinceID "& \_

" INNER JOIN Person.CountryRegion SCR ON "& \_

" SSP.CountryRegionCode = SCR.CountryRegionCode "& \_

" RIGHT OUTER JOIN Sales.SalesOrderHeader SOH "& \_

" LEFT OUTER JOIN Person.Contact CC ON "& \_

" SOH.ContactID = CC.ContactID LEFT OUTER JOIN"& \_

" Person.Address BA INNER JOIN "& \_

" Person.StateProvince BSP ON "& \_

" BA.StateProvinceID = BSP.StateProvinceID "& \_

" INNER JOIN Person.CountryRegion BCR ON "& \_

" BSP.CountryRegionCode = "& \_

" BCR.CountryRegionCode ON SOH.BillToAddressID "& \_

" = BA.AddressID ON SA.AddressID = "& \_

" SOH.ShipToAddressID LEFT OUTER JOIN "& \_

" Person.Contact C RIGHT OUTER JOIN "& \_

" HumanResources.Employee E ON C.ContactID = "& \_

" E.ContactID ON SOH.SalesPersonID = "& \_

" E.EmployeeID LEFT OUTER JOIN "& \_

" Purchasing.ShipMethod SM ON SOH.ShipMethodID "& \_

" = SM.ShipMethodID LEFT OUTER JOIN Sales.Store"& \_

" S ON SOH.CustomerID = S.CustomerID "& \_

"WHERE (SOH.SalesOrderNumber = @SalesOrderNumber)"

Using connection AsNewSqlConnection( \_

"Data Source=(local); "& \_

"Initial Catalog=AdventureWorks; "& \_

"Integrated Security=SSPI")

Dim command AsNewSqlCommand(sqlSalesOrder, connection)

Dim parameter AsNewSqlParameter("SalesOrderNumber", \_

salesOrderNumber)

command.Parameters.Add(parameter)

DimsalesOrderAdapterAsNewSqlDataAdapter(command)

salesOrderAdapter.Fill(dsSalesOrder, "SalesOrder")

EndUsing

EndSub

PrivateSubGetSalesOrderDetailData( \_

ByValsalesOrderNumberAsString, \_

ByRefdsSalesOrderAsDataSet)

DimsqlSalesOrderDetailAsString = \_

"SELECT SOD.SalesOrderDetailID, SOD.OrderQty, "& \_

" SOD.UnitPrice, CASE WHEN "& \_

" SOD.UnitPriceDiscount IS NULL THEN 0 "& \_

" ELSE SOD.UnitPriceDiscount END AS "& \_

" UnitPriceDiscount, SOD.LineTotal, "& \_

" SOD.CarrierTrackingNumber, "& \_

" SOD.SalesOrderID, P.Name, P.ProductNumber "& \_

"FROM Sales.SalesOrderDetail SOD INNER JOIN "& \_

" Production.Product P ON SOD.ProductID = "& \_

" P.ProductID INNER JOIN "& \_

" Sales.SalesOrderHeader SOH ON "& \_

" SOD.SalesOrderID = SOH.SalesOrderID "& \_

"WHERE (SOH.SalesOrderNumber = @SalesOrderNumber) "& \_

"ORDER BY SOD.SalesOrderDetailID"

Using connection AsNewSqlConnection( \_

"Data Source=(local); "& \_

"Initial Catalog=AdventureWorks; "& \_

"Integrated Security=SSPI")

Dim command AsNewSqlCommand(sqlSalesOrderDetail, \_

connection)

Dim parameter AsNewSqlParameter("SalesOrderNumber", \_

salesOrderNumber)

command.Parameters.Add(parameter)

DimsalesOrderDetailAdapterAsNewSqlDataAdapter(command)

salesOrderDetailAdapter.Fill(dsSalesOrder, \_

"SalesOrderDetail")

EndUsing

EndSub

EndClass

См. также:

После добавления элемента управления в форму появляется смарт-тег **Задачи ReportViewer** и выводится приглашение выбрать отчет. Если требуемый отчет развернут на сервере отчетов, выберите параметр **<Отчет по серверу>** из раскрывающегося списка **Выбор отчета**. После выбора параметра **<Отчет по серверу>** на экране появляются два дополнительных свойства: **URL-адрес сервера отчетов** и **Путь к отчету**. Свойство **URL-адрес сервера отчетов** определяет адрес сервера отчетов, а свойство **Путь к отчету** — полный путь к подлежащему отрисовке отчету.

Для просмотра отчета в локальном режиме необходимо выбрать параметр **Создать новый отчет**, чтобы запустить конструктор отчетов, или выбрать отчет, который уже является частью существующего проекта. После выбора отчета обязательно введите имя RDLC-файла отчета в свойстве **ReportPath** элемента управления ReportViewer. Это свойство находится в узле **LocalReport** в области **Свойства**.

При подготовке отчета к просмотру есть возможность скрыть один или несколько элементов на панели инструментов ReportViewer. Например, можно скрыть кнопку печати. Для скрытия элементов панели инструментов присвойте значение **False** указанным ниже свойствам в области **Свойства**.

* **ShowBackButton**
* **ShowExportControls**
* **ShowFindControls**
* **ShowPageNavigationControls**
* **ShowPrintButton**
* **ShowRefreshButton**
* **ShowZoomControl**