Walkthrough: Using a Database Data Source with the ReportViewer Windows Forms Control in Local Processing Mode

This walkthrough shows how to create a report in a Microsoft Visual Studio 2008 Windows Application Project and how to add a ReportViewer control to a Windows Form so that the report can be viewed by users of the application.

http://i.msdn.microsoft.com/Global/Images/clear.gif Prerequisites

To use this walkthrough, you must have access to the **AdventureWorks** sample database. For more information, see [Walkthrough: Installing the AdventureWorks Database](http://msdn.microsoft.com/en-us/library/aa992075.aspx).

Perform the following steps to add a report to a Visual Studio Windows application project. For this example, you create the application in Microsoft Visual Basic.

http://i.msdn.microsoft.com/Global/Images/clear.gif Create a new Windows application project

1. Open Visual Studio 2008. On the **File** menu, point to **New**, and then select **Project**.
2. In the Project Types pane, choose **Visual Basic**.
3. In the Templates pane, choose **Windows Application** to create a Microsoft Windows application.
4. In the **Name** box, type the name of the project: **ReportWalkthrough**.
5. In the **Location** box, enter the directory in which you want to save your project, or click **Browse** to navigate to it. The Windows Forms Designer opens, showing **Form1.vb** of the project you created.

http://i.msdn.microsoft.com/Global/Images/clear.gif Define a data source connection and DataTable

1. On the **Project** menu, select **Add New Item**.
2. In the **Add New Item** dialog, click **DataSet**.
3. Enter a name for the dataset and click **Add**. This adds a new XML Schema file to the project and opens the DataSet Designer. The default name is **DataSet1.xsd**.
4. In DataSet Designer mode, open the Toolbox, and drag a **TableAdapter** onto the DataSet design surface. This opens the TableAdapter Configuration Wizard.
5. On the **Choose Your Data Connection** page, click **New Connection**.
6. In the **Data Source** dialog box, select **Microsoft SQL Server**. In the **Server name** dialog box, enter the server where **AdventureWorks** is located. Select **AdventureWorks** from the database drop-down list, and click **OK** to continue to the next page of the wizard.
7. On the **Save the Connection String to the Application Configuration File** page, type in the name for the connection string or accept the default **AdventureWorksConnectionString**. Click **Next**.
8. On the **Choose the Command Type** page, select **Use SQL Statements**. Confirm the other radio buttons are not selected.
9. On the **Enter a SQL Statement** page, enter the following Transact-SQL query to retrieve data from the **AdventureWorks** database, and then click **Finish**. You can also click on the **Query Builder** button and use Query Builder to create your query and check it using the **Execute Query** button.

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl11other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl11other');)

SELECT d.name as Dept, s.Name as Shift, e.EmployeeID as EmployeeID

FROM (HumanResources.Department d

INNER JOIN HumanResources.EmployeeDepartmentHistory e

ON d.DepartmentID = e.DepartmentID)

INNER JOIN HumanResources.Shift s

ON e.ShiftID = s.ShiftID

If the query does not return the expected data, you might be using an earlier version of AdventureWorks. For more information about installing the SQL Server 2005 version of AdventureWorks, see [Walkthrough: Installing the AdventureWorks Database](http://msdn.microsoft.com/en-us/library/aa992075.aspx).

1. On the **Choose Methods to Generate** page, accept the defaults **Fill a DataTable** with **Method name**: **Fill** and **Return a DataTable** with **Method name**: **GetData**. Click **Next**.
2. On the **Wizard Results** page, click **Finish**.
3. On the DataSet Designer page, you should see **DataTable1** with the columns resulting from the query. From the **Data** menu, choose **Show Data Sources** and expand the **DataTable1** node to see these columns. You will use the **Data Sources** window and the Dept, Shift, and EmployeeID fields when binding data to your report in the next step.

http://i.msdn.microsoft.com/Global/Images/clear.gif Design the report

1. On the **Project** menu, select **Add New Item**.
2. In the **Add New Item** dialog, click **Report**.
3. Enter a name for the report file. By default, the report name is **Report1.rdlc**. Click **Add**. Report Designer opens and displays the dotted surface that represents the report page.
4. Open the Toolbox. Click on a textbox and then on the form.
5. Enter a report title in the textbox: **# of Employees per Shift per Dept**.
6. From the Toolbox, drag a **Matrix** report item onto the report below the textbox.
7. From the Data Sources window, expand **DataTable1** to see the columns Dept, Shift and EmployeeID. Drag the Deptfield onto the **Rows** textbox in the first column, second row of the matrix.
8. Drag the Shift field onto the **Columns** textbox in the second column, first row of the matrix. While the textbox is selected, set the **TextAlign** property to **Right**. Click the **B** icon on the **Report Formatting** toolbar to use bold font.
9. Select the matrix data region by clicking anywhere in the matrix. Then right-click and select the matrix by name (the default is **matrix1**). An outline will appear around the matrix report item. Right-click on the outline and select **Properties**. Check that the title of the dialog box that opens is **Matrix Properties**.
10. Click the **Groups** tab. In the **Columns** section, click **Edit**.
11. In the **Sorting and Grouping** dialog box, click the **Sorting** tab.
12. Click in the first box under **Expression** to activate the textbox and then click the dropdown arrow and select **=Fields!Dept.Value**. This ensures the report data will be sorted on the Department name.
13. From the **Data Sources** window, drag the EmployeeID field onto the **Data** textbox in the second column, second row of the matrix. Right-click on this textbox and select **Properties**.
14. In the **Properties** window on the **General** tab, click the expression (*fx*) button to invoke the expression editor.
15. Edit the default **Sum** aggregate function and change it to **Count**. You can invoke the expression editor by right-clicking on the textbox and choosing **Properties**. The expression should read:

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl16other');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl16other');)

=Count(Fields!EmployeeID.Value)

http://i.msdn.microsoft.com/Global/Images/clear.gif Add a ReportViewer control to the application

1. In Solution Explorer, right-click **Form1.vb** and choose **View Design** to open your form in Design mode.
2. In the **Form** properties window, click the **+** to expand the **Size** property. Set the Form **Height** to 700.
3. From the Data section of the Visual Studio toolbox, click the **ReportViewer** icon and click the form. Adjust your form width and ReportViewer control width as needed.
4. Open the smart tags panel of the ReportViewer control by clicking the triangle on the top right corner. Click the **Choose Report** drop-down list and select **Report1.rdlc**. Selecting a report causes instances of data sources used in the report to be created automatically. Code is generated to instantiate a [DataSet](http://msdn.microsoft.com/en-us/library/system.data.dataset.aspx) (the ADO.NET container for a [DataTable](http://msdn.microsoft.com/en-us/library/system.data.datatable.aspx)), a **TableAdapter** component, and a [BindingSource](http://msdn.microsoft.com/en-us/library/system.windows.forms.bindingsource.aspx) object corresponding to each data source used in the report.
5. In the open smart tags panel, choose **Dock in parent container**.

http://i.msdn.microsoft.com/Global/Images/clear.gif Build and run the application

1. On the **Build** menu, click **Build ReportWalkthrough**. As part of the build process, the report is compiled. Some errors (such as a syntax error in an expression used in the report) are noted by adding them to the **Task List**.
2. Press **F5** to run the application and view the report in your form.

http://i.msdn.microsoft.com/Global/Images/clear.gif See Also

**Reference**

[Microsoft.Reporting.WinForms.ReportViewer.Drillthrough](http://msdn.microsoft.com/en-us/library/microsoft.reporting.winforms.reportviewer.drillthrough.aspx)  
[Microsoft.Reporting.WinForms.LocalReport.SubreportProcessing](http://msdn.microsoft.com/en-us/library/microsoft.reporting.winforms.localreport.subreportprocessing.aspx)  
[Microsoft.Reporting.WebForms.ReportViewer.Drillthrough](http://msdn.microsoft.com/en-us/library/microsoft.reporting.webforms.reportviewer.drillthrough.aspx)  
[Microsoft.Reporting.WebForms.LocalReport.SubreportProcessing](http://msdn.microsoft.com/en-us/library/microsoft.reporting.webforms.localreport.subreportprocessing.aspx)

**Concepts**

[Using the ReportViewer Tasks Smart Tags Panel](http://msdn.microsoft.com/en-us/library/ms252072.aspx)

**Other Resources**

[Samples and Walkthroughs](http://msdn.microsoft.com/en-us/library/ms251686.aspx)