

Web Reports Sample

This sample is a PowerBuilder sales application that you deploy to an IIS Web server as a .NET Web Forms project.

Description

The generated application has the following Web Forms pages:

- **Overview** - Displays a Composite Style DataWindow. It contains a report summary of product sales, a graph of the product description versus the dollar amount of product sales, a list of the current product inventory, and a list of orders by sales representative.
- **Sales Order** - This page is a DataWindow displaying customers with nested orders. It lists detail information for a selected customer and the customer's order history. You can select customers from the drop-down list near the top of the page.
- **Expense** - Contains a bar graph of department expenses per quarter for a given year and a pie chart with percentages of overall expenditure per department. The DataWindow on the right provides the actual data for the graph and pie chart. You can change the year of the expense analysis in the drop-down list near the top of the page. When you click the "quarter" hyperlink in the DataWindow, the RowFocusChanged event initiates a postback to the server. The server then reloads the page on the client with the quarter that you clicked detached from ("exploded" on) the pie chart.
- **Employee** - Contains the compensation report for all salaried employees grouped by department. You can select the department from a treeview DataWindow at the left of the page. A toolbar near the top of the page lets you filter, sort, print, or save the current DataWindow data to a text file on the server. You can click the employee ID or name in the current DataWindow to retrieve employee detail information.

Features

This sample demonstrates various DataWindow styles as well as DropDownPictureListBox methods and events in a Web Forms application.

Setting Up Your Environment

Ensure that Visual Studio .NET Framework 2.0 or later is installed on your computer. IIS Web Controls must be installed in your IIS path: \inetpub\wwwroot\.

If the components are not installed, see "Configuring ASP.NET for a PowerBuilder project" in Chapter 1 of the [Deploying Applications and Components to .NET](#) book in the HTML Help.

IIS 5.0 or higher must be installed on the computer and configured to use ASP.NET 2.0.

Detailed Instructions

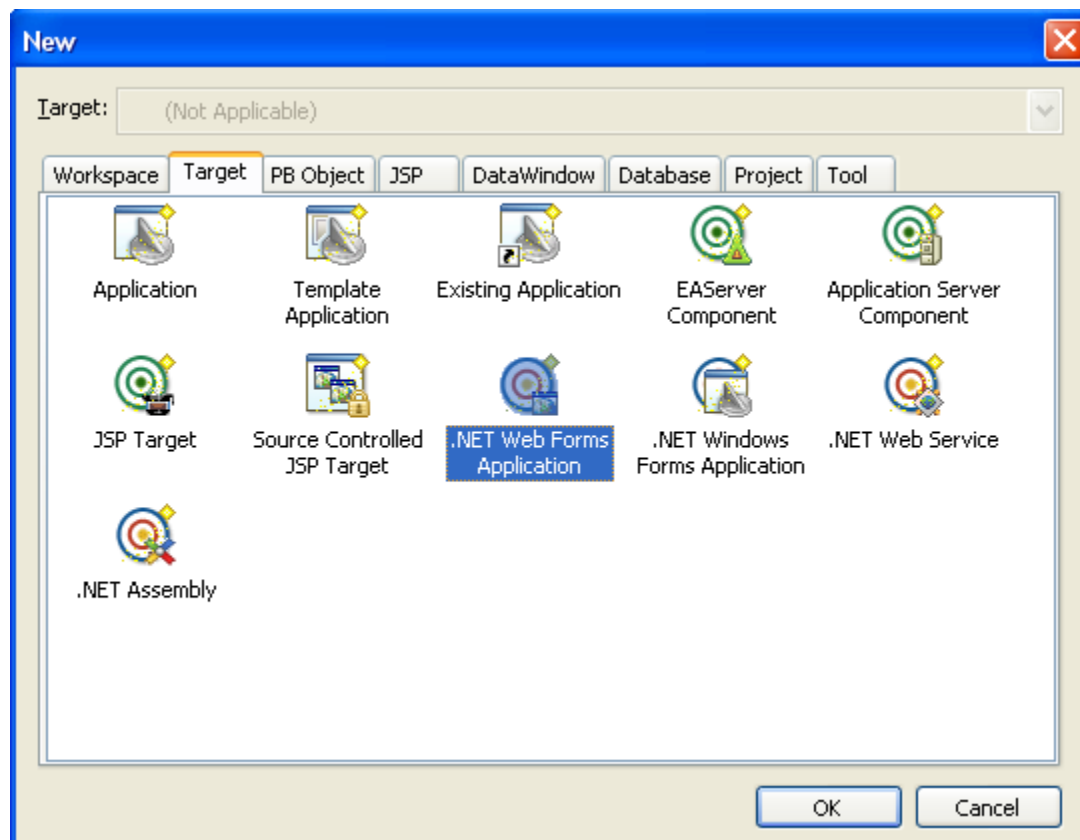
Follow these steps to open the sample in PowerBuilder 11.5:

1. Unzip the contents of the associated zip file into a directory on your computer.
2. Make sure that the ODBC DSN "EAS Demo DB V115" for the sample SQL Anywhere database is configured on the Web server. The ODBC DSN "EAS Demo DB V115" is created automatically when you install PowerBuilder 11.5.

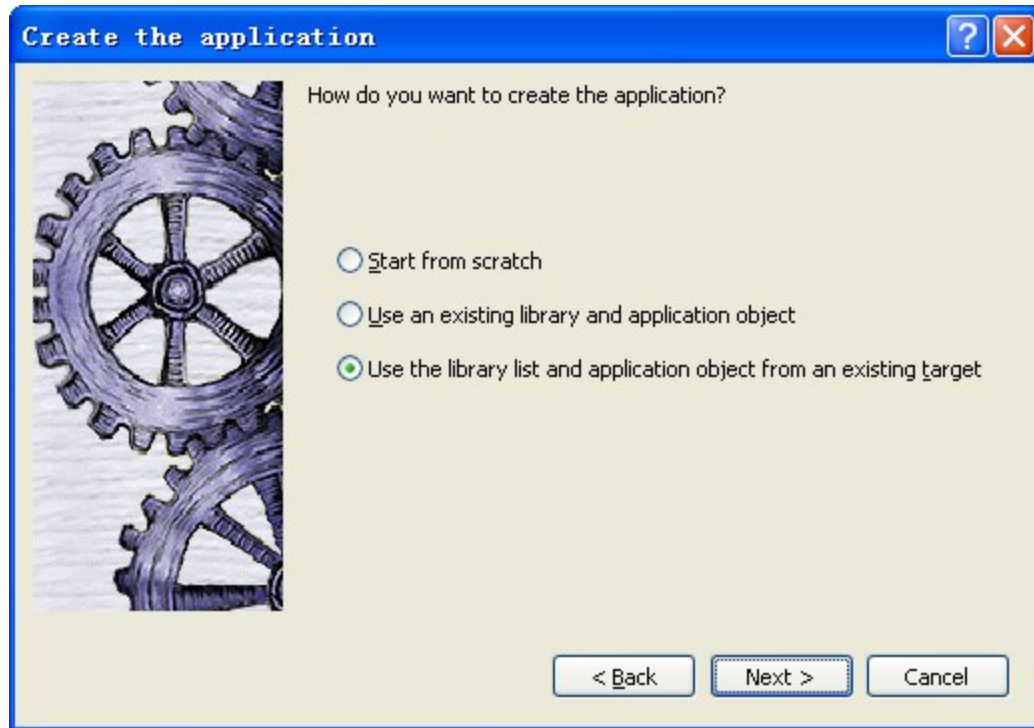
Note: Due to limited access rights of the ASPNET user (Windows XP) or the IIS_WPG user group (Windows 2003), data sources created as User DSNs cannot be loaded. You must create the data sources for your Web Forms application as System DSNs.

To launch a SQL Anywhere database automatically from a Web Forms application, the ASPNET user (Windows XP) or the IIS_WPG user group (Windows 2003) must be granted at least read and execution rights to the directory indicated by the SQLANY10 environment variable. The ASPNET user or the IIS_WPG user group must also have full control privileges to the directory that contains the database.

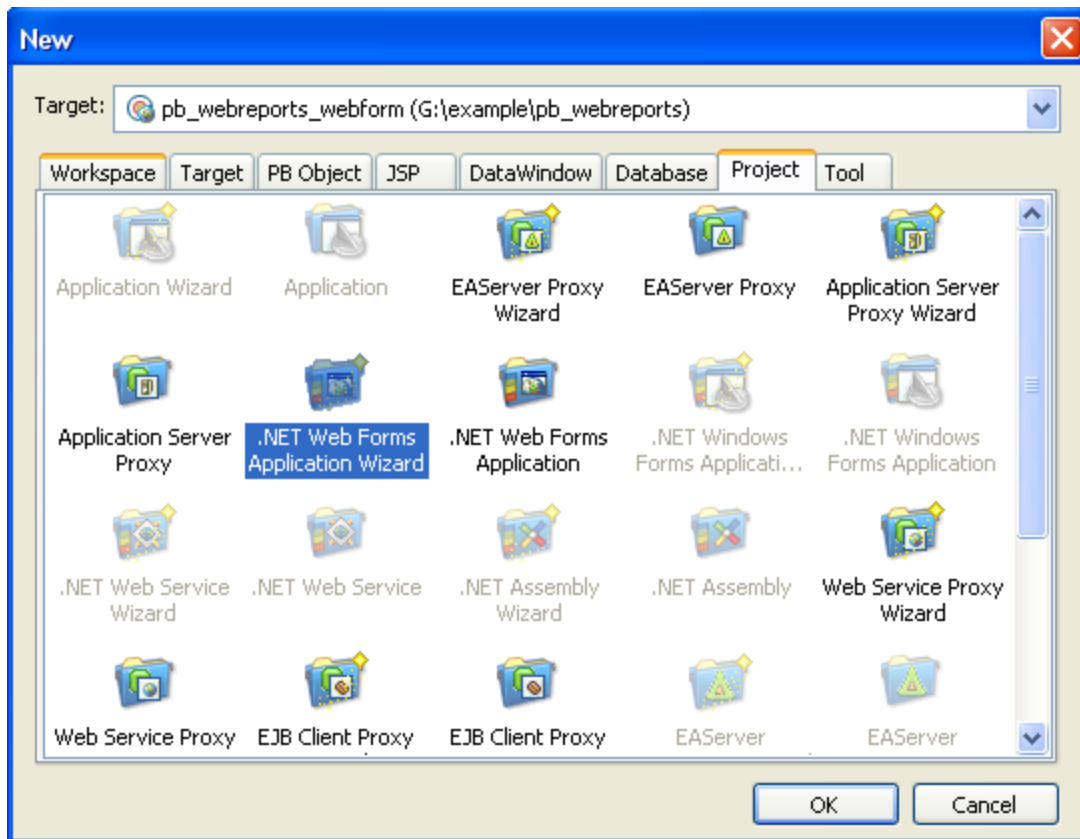
3. In PowerBuilder 11.5, select File>Open Workspace.
4. In the Open Workspace dialog box, navigate to the directory where sample files are unzipped, select pb_webreports.pbw, and click Open.
5. Select File>New to open the New dialog box, then select .Net Web Forms Application on the Target page.



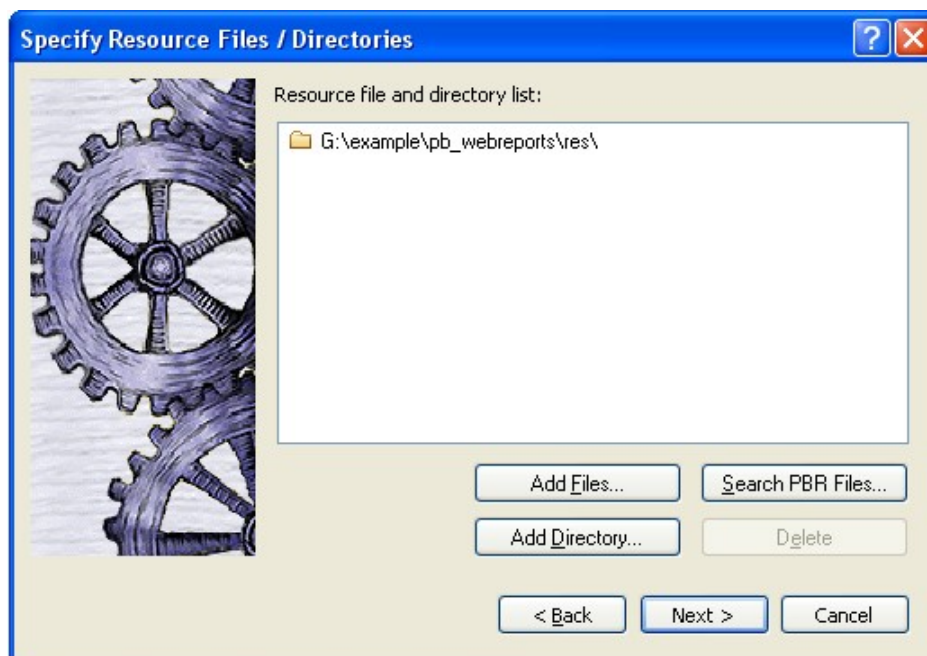
6. Since the application has been created already, choose the option "Use the library list and application object from an existing target" in the wizard.



7. Choose the target pb_webreports from the workspace, then specify a new target name such as pb_webreports_webform.pbt. This is the default name for the new target, so you do not need to change the new target name.
8. Choose the default setting to create the project later, click Next, and then click Finish to generate the new target. You start the project wizard in the next step from the Project page of the New dialog box.
9. Make sure the new Web Forms target is selected in the Target drop-down list near the top of the Project page of the New dialog box. Select *.Net Web Forms Application Wizard* on the Project page and click OK.



10. Accept the wizard default settings, but on the Specify Resource Files/Directories page, add a folder named `res` to the Resource File and Directory list box. You can click Add Directory to browse to this folder. It is located under the directory where you extracted the sample zip file.



After you click Finish in the project wizard, PowerBuilder creates a .NET Web Forms project in a Web Forms target library and opens the project in the Project painter

The Employee page of the sample application lets you save DataWindow data to a text file on the server. If you to download this file from or view it on the server, you must render the File Manager icon in your Web Forms application. You can render the File Manager icon by changing the PBFileManager global property to true on the Configuration tab of the Project painter.

For more information about the File Manager and the File Manager icon, see the [Deploying Applications and Components to .NET](#) book in the HTML Help.

11. Select Design>Deploy Project from the Project painter to deploy the project. Alternatively, when all painters are closed, including the Project painter, you can right-click a .NET Web Forms project in the System Tree and select Deploy from its pop-up menu.

The Output window displays the progress of the deployment and provides a list of application functions, events, and properties that are not supported in the Web Forms version of the application. Most of these unsupported features and warnings are benign and do not prevent users from running the application as Web Forms.

If the 2.0 version is the only version of the Microsoft .NET Framework installed on the server, or if you configured the server to use the 2.0 version for all Web sites by default, you can run the application immediately after you deploy it. By default, when you run the application from the design-time environment, PowerBuilder opens the Internet Explorer browser to the project deployment address specified on the Run tab of the Project painter.

12. Run the application from PowerBuilder by selecting Design>Run Project from the Project painter menu or selecting the Run Project toolbar icon from the Project painter toolbar. The System Tree pop-up menu for the .NET Web Forms project also has a Run Project menu item.

When the application starts, the main page displays the Products & Sales information. You can continue testing this example by toggling to other pages of the Web application. These pages are described in the Description section of this Readme.