Web Services Sample

The sample in this package demonstrates the use of the .NET Web Services client functionality in PowerBuilder 11.5.

Description

This sample uses the .NET Web Services client proxy and underlying .NET implementation classes to invoke a variety of publicly accessible Web Services.

Setting up your environment

Make sure the .NET Framework 2.0 (or 3.0/3.5) and .NET Framework 2.0 (or 3.0/3.5) SDK have been installed on your computer. Make sure that PowerBuilder 11.5 has been installed according to the instructions in the accompanying release documentation.

Follow these steps to install and run the samples:

- Start PowerBuilder 11.5 and open the webservices.pbw workspace.
- Redeploy the entire workspace to regenerate the proxies and assemblies on your local machine. (This should be necessary only if you encounter runtime errors which are probably caused by web services changes or missing runtime assemblies.)

  Note: If during the deployment process a message "Can not access the WSDL or the WSDL file is invalid " appears in the Output window, it is likely that the Web Service is not operating correctly and the site hosting the WSDL cannot be contacted. You can examine the Output window to determine which proxy project failed to deploy.

Running the sample

This code example has been enhanced in PowerBuilder 11.5 to allow you to run the application as a Win32 (standard PowerBuilder) application, as a .NET Web Forms application, or as a .NET Windows Forms application. All three targets reference the same .pbls. Depending on the target that is active, you will see a different project object in the ws.pbl in the System Tree.

Win32 (P-Code) Implementation

To run this application as a standard PowerBuilder p-code application, simply click on the Select and Run toolbar button (the running man with an arrow button) and select the ws target from the list of available targets. If you wish to create a compiled version of this application, deploy the p_ws_exe project object.

.NET Web Forms

You must deploy ps_webform1 project in the ws_webform target before running the .NET Web Forms implementation of this application. Before you can deploy a PowerBuilder application as a .NET Web Forms application you must make sure your environment has been configured to use IIS and ASP .NET 2.0. Please refer to the Deploying Applications and Components to .NET manual in the compiled HTML file for more information.

.NET Windows Forms

You must deploy ps_winform project in the ws_winform target before running the .NET Windows Forms implementation of this application. Before you can deploy a PowerBuilder application as a .NET Windows Forms application you must make sure your environment has been configured to use Microsoft .NET 2.0 (or 3.0/3.5). Please refer to the Deploying Applications and Components to .NET manual in the compiled HTML file for more information.

Using the Web Services Sample

The left pane of the window provides a list (using the TreeView DataWindow presentation style) of various public Web Services culled from the Xmethods site (http://www.xmethods.net). The services are organized by style (RPC or Document style) and then by the method of implementation for the respective service. The list of services, including the information that appears on the General tab and the WSDL URL, is stored in the external DataWindow object d_svclist. All services with a svcstatus field value of “Y” will be shown in the TreeView DataWIndow; the others are filtered out.

Each service is associated with a custom class visual user object that is used on the second tab page in the display. This tab contains the custom user interface needed to invoke the requested service. In most cases, input fields are pre-populated so that all you need to do is press the Invoke button to execute the service and see the results.

If you would like to add a new service into the example, simply add a new row to the external DataWindow d_svclist containing the information for that service. Then extend the uo_ancestor object to create the UI needed to invoke your service. The name of your inherited user object should be stored in the svcuo field of the DataWindow row that you inserted for the new service.

All of the Web Services referenced in this example were found on the XMethods site; however, since that site has frequent changes, one or more of the services may not be operational at the time you run this sample. If that is the case, exception handling within the example should allow for a graceful failure.

For more information

For more information, select New Features from the Contents page of the online Help.