DataWindow and UI Enhancements example

This sample demonstrates the use of the enhanced menu and toolbar functionality in PowerBuilder 11 as well as the following features:

- TreeView DataWindow
- DatePicker
- Auto-Size Height for All DataWindow Bands
- Suppress Group Headers

Setting up your environment

Make sure that PowerBuilder 11 has been installed according to the instructions in the accompanying release documentation and that the ODBC DSN EAS Demo DB V110 Unicode has been configured.

Follow these steps to install and run the samples:

1. Start PowerBuilder 11 and open the pbdemos.pbw workspace from the PowerBuilder 11\Code Examples\New DataWindow and UI folder.
2. In the System Tree, right-click the pbdemos target, and then select Run from the pop-up menu.

Note: If you would like to run this code example as a .NET Windows Forms smart client application, see the instructions near the end of this document.

Running the Menu and Toolbar example

You will see a menu with a sky blue background and navy blue text. You will also see a toolbar with a white-to-cream gradient. You may notice the new bitmaps and icons available for your applications. If you want to see the toolbar gradient accentuated, right mouse click anywhere on the toolbar and select “Show Text” from the popup menu. Note the enlarged toolbar icons. Also note that moving your mouse cursor over the toolbar icons allows the toolbar ToolbarHighlightColor property to manifest itself. Right-click on the toolbar once more and experiment with the different toolbar manifestations by selecting the different toolbar position options from the pop-up menu (Left, Right, Top, Bottom, Floating).

Experiment with navigating the drop-down and cascading menus in this sample. Note the titlebars and their text and backcolor options. Note the icon strip background color gradient. Note also the menu and toolbar animation options available.

To experiment with modifying this sample menu/toolbar, terminate the running application, then open m_frame in the menu painter. In the treeview pane/frame of the menu painter, select the root node “m_frame”, and then select the Appearance tabpage in the Properties pane/frame of the menu painter. Note the Menu Style property and the two enumerated values in the dropdownlistbox, traditionalmenu! (self explanatory) and the value that implements the new menu and toolbar capabilities, contemporarymenu!. Modify cosmetic property values as you wish and note their results at runtime.

Running the TreeView DataWindow examples

From the menu select File>New, then any one of the three TreeView DataWindow samples.

There are three samples: Data Explorer, TreeView DataWindow Linking, and Employee Manager.

Data Explorer

The Data Explorer sample displays data in a Windows Explorer-style interface so that the data can be manipulated. Clicking on each name in the TreeView DataWindow on the left displays detailed data in a DataWindow on the right. You can expand or collapse TreeView levels by right-clicking on the TreeView DataWindow and then selecting Expandall or Collapseall.

TreeView DataWindow Linking

The TreeView DataWindow Linking sample uses a TreeView DataWindow on the left for data navigation, linked to four DataWindows on the right for updating the data. The sample demonstrates populating a TreeView DataWindow with data and linking each TreeView level to a separate DataWindow.

Employee Manager

The Employee Manager sample displays employee data in a grid-style TreeView DataWindow. By right-clicking on the TreeView DataWindow, you can expand or collapse all levels, show or hide grid lines, and highlight TreeView nodes or not.

The DataExplorer sample and the TreeView DataWindow Linking sample use the employee, sales_order, sales_order_items, customer, and product tables in the EAS Demo DB database. The TreeView DataWindows (d_sales_report and d_sales_report2) each have three TreeView levels. In the top level (created using the TreeView DataWindow wizard), the sales representative's name displays. In the second and third levels (created using the Rows>Create TreeView Level menu item in the DataWindow painter), the customer's company name displays (in level 2) and the sales order ID displays (in level 3).

The Employee Manager sample uses the employee table in the EAS Demo DB database. The d_emp_demo TreeView DataWindow has three TreeView levels. In the top level (created using the TreeView DataWindow wizard), the department name displays. In the second and third levels (created using the Rows>Create TreeView Level menu item in the DataWindow painter), the sex of the employee (level 2) and the status of the employee (level 3) display.

Using the Data Explorer example

Select an employee name in the TreeView DataWindow. Expanding the TreeView shows the customer company name, the orders for that company, and, in the detail band, the order date, financial code, sales region, and sales representative's ID. You can click on each TreeView level to display details in a DataWindow on the right. The code in the Clicked event uses the GetBandAtPointer method to determine which DataWindow to display. Clicking on some editable items in the detail DataWindow opens a window in which you can manipulate the data. The PopMenu menu object has two menu items that call the CollapseAll and ExpandAll functions to collapse or expand all the nodes in the TreeView.

Using the TreeView DataWindow Linking example

Select an employee name in the TreeView DataWindow. If you expand the TreeView, you display the customer company name, the orders for that company, and, in the detail band, the icon and name for each item in the order. You can click on each TreeView level and in the detail band to display details in one of four DataWindows on the right. The code in the Clicked event uses GetBandAtPointer to determine which DataWindow to display.

Using the Employee Manager example

The PopMenu menu object has six menu items that call the CollapseAll and ExpandAll functions to collapse or expand the TreeView nodes, call the modify function to set the datawindow.grid.line property to show or hide grid lines, and change the return code of the TreeNodeSelecting event to affect the SelectTreeNode function.

For more information

For more information, see the chapter on TreeView DataWindows in the User's Guide in the HTML Help.

Running the DatePicker example

From the menu select File>New>DatePicker.

Using the DatePicker example

Note the format of the date in the DatePicker control labeled "Long Date". Drop the control down. Click on the left and right arrowheads in the titlebar to change the month. Click on the month and/or the year in the titlebar. Note the alternative ways to select the month and/or year. Perform the same operations on the DatePicker control labeled Short Date if you wish.

On the control labeled Time, select the month, day or year and note how the spin control modifies the value.

On the control labeled Custom Format, note that you can specify a DateTime format different from that specified in the users' Regional Settings on their workstations. Select the month, day or year and note how the spin control modifies the value.

Running the DataWindow Band Auto-Size Height example

From the menu select File>New>Band Auto-size Height.

Using the DataWindow Band Auto-Size Height example

The DropDownListBox at the top of the window changes the DataWindow object assigned to the main DataWindow control in the sample window. This allows you to examine the Auto-Size Height behavior within the context of different presentation styles.

As you select different DataWindow objects, the various items in the AutoSize Height group box change their enabled/disabled configuration as is appropriate to each DataWindow object. The Method group box allows you to examine the behavior of each example using dot notation or Describe and Modify.

The best example is that using d_composite. The primary motivation behind adding this feature was to allow non-detail bands to automatically resize for the variable number of rows that might be displayed by nested reports (or child DataWindows) that are placed in those bands. In the AutoSize Height group box, uncheck the Header check box to "turn off" the feature for that DataWindow and note that the nested report (child DataWindow) disappears from view. Check the box once more to redisplay the nested report.

Use the DropDownListBox to examine the variations in this new feature in various DataWindow objects.

Note also that there is a Print Preview Margin Border checkbox near the top of the window. Deselect the checkbox and you will notice that the blue margin outline border disappears. You can now display DataWindows in Print Preview mode without having the blue line display.

Running the Suppress Group Headers example

From the menu, select File>New>Suppress Group Headers. Notice that the DropDownListBox at the top of the window allows you to change the DataWindow object assigned to the main DataWindow control in the sample window. This allows you to examine the Suppress Group Headers behavior of several grouped DataWindows.

As you select different DataWindow objects, the items in the Suppress Header group box change their enabled/disabled configuration as appropriate for each DataWindow object.

The first example uses d_group_1. The goal of this feature is to enable you to suppress group headers on page breaks in a grouped DataWindow. In the DataWindow Preview view, scroll to page 2. There you will find that the group header that displays at the top of the page in earlier PowerBuilder releases is now suppressed. In the Suppress Header group box, clear the header.1 check box to disable the feature for that DataWindow. Note that the group header appears. Check the check box again to suppress the group header.

Continue using the DropDownListBox to examine the variations in this new feature in a variety of DataWindow objects.

.NET Windows Forms Smart Client example

This code example also includes a .NET Windows Forms target where the project object is set up to publish the application as a smart client. To try this feature out you must first make sure that your environment is configured to support Microsoft .NET 2.0, IIS and ASP .NET 2.0. Please refer to the Deploying Applications and Components to .NET manual in the online HTML Help for more information on configuring your environment.

After you have configured your environment to support .NET Windows Forms applications and to publish smart client applications, you can deploy the pbdemos_winform target using the project painter. Open the p_pbdemos_winform project object and examine the property values on the various tabs. When you are ready, click on the Deploy Project toolbar button. Once the deployment has completed you can then click on the Publish Application toolbar button. After the publication is complete you can run the example as a smart client application by clicking on the Run button that appears on the publish page in the web browser.