

New Features in PowerBuilder® 10.5

Document ID: DC00357-01-1050-01

Last revised: March 2006

Topic	Page
DataWindow enhancements	2
TreeView DataWindow presentation style	2
RichText enhancements	4
Decimal support in DataWindow expressions	4
New style for drop-down DataWindows in Web DataWindows	6
New HTMLGen.PagingMethod property for Web DataWindows	6
Drop-down calendar DataWindow option	7
Autosize height property on all DataWindow bands	8
New property to suppress group headers on page breaks	9
New property to hide blue outline in Print Preview	9
New dbAlias DataWindow object property	10
New ResetInk, SaveInk, and SaveInkPic DataWindow methods	10
User interface enhancements	10
New icons	10
Menu and toolbar enhancements	10
New and updated PowerBuilder controls	11
Rich Text enhancements	11
DatePicker control	19
DropDownCalendar property for EditMask controls	19
DateSelected event for the MonthCalendar control	20
Language enhancements	20
Importing PowerBuilder extensions	20

Copyright 1991-2006 by Sybase, Inc. All rights reserved. Sybase, the Sybase logo, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Warehouse, Aflaria, Answers Anywhere, Application Manager, AppModes, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-Translator, APT-Library, AvantGo Mobile Delivery, AvantGo Mobile Inspection, AvantGo Mobile Marketing Channel, AvantGo Mobile Sales, AvantGo Pylon, AvantGo Pylon Application Server, AvantGo Pylon Conduit, AvantGo Pylon PIM Server, AvantGo Pylon Pro, Backup Server, BizTracker, ClientConnect, Client-Library, Client Services, Convoy/DM, Copernicus, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DataWindow .NET, DB-Library, dbQueue, Developers Workbench, DirectConnect, DirectConnect Anywhere, Distribution Director, e-ADK, E-Anywhere, e-Biz Impact, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTP, eFulfillment Accelerator, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, EWA, Financial Fusion, Financial Fusion Server, Gateway Manager, GlobalFIX, iAnywhere, iAnywhere Solutions, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InternetBuilder, iScript, Jaguar CTS, jConnect for JDBC, M2M Anywhere, Mach Desktop, Mail Anywhere Studio, Mainframe Connect, Maintenance Express, Manage Anywhere Studio, M-Business Anywhere, M-Business Channel, M-Business Network, M-Business Suite, MDI Access Server, MDI Database Gateway, media.splash, MetaWorks, mFolio, Mirror Activator, MySupport, Net-Gateway, Net-Library, New Era of Networks, ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Biz, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima+, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, Pharma Anywhere, PocketBuilder, Pocket PowerBuilder, Power+, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, QAnywhere, Rapport, RemoteWare, RepConnector, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Report-Execute, Report Workbench, Resource Manager, RFID Anywhere, RW-DisplayLib, RW-Library, Sales Anywhere, SDF, Search Anywhere, Secure SQL Server, Secure SQL Toolkit, Security Guardian, SKILLS, smart.partners, smart.parts, smart.script, SOA Anywhere, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolkit, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLI, STEP, SupportNow, S.W.I.F.T. Message Format Libraries, Sybase Central, Sybase Client/Server Interfaces, Sybase Financial Server, Sybase Gateways, Sybase IQ, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SybFlex, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, Trade-Force, Transact-SQL, Translation Toolkit, UltraLite, UltraLite.NET, UNIBOM, Unilib, Uninsep, Unistring, URK Runtime Kit for UniCode, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server, XcelleNet, and XP Server are trademarks of Sybase, Inc. 10/05

Topic	Page
Decimal datatype support	21
Byte datatype support	21
New OLERuntimeError properties	22
Database connectivity enhancements	22
Database tracing enhancements	22
Adaptive Server Enterprise 15 support	23
ADO.NET support for Sybase Adaptive Server and Oracle	24
Support for Sybase Adaptive Server temporary tables	25
Support for Microsoft SQL Server 2005	25
NCharBind database parameter for Oracle	25
Web Services enhancements	26
About the .NET Web service engine	26
Using the .NET Web service engine	26
Using the EasySoap Web service engine	28
New methods for the SoapConnection object	28

DataWindow enhancements

PowerBuilder® 10.5 includes the new TreeView presentation style, Web DataWindow® enhancements, autosize height for all DataWindow bands, and other enhancements.

TreeView DataWindow presentation style

The presentation style you select for a DataWindow object determines the format PowerBuilder uses to display the DataWindow object in the Design view. PowerBuilder 10.5 includes a new DataWindow wizard that lets you create a DataWindow using a TreeView presentation style. With the TreeView presentation style, you can group hierarchical data and display the data in collapsed or expanded format. The TreeView DataWindow style cannot be used with the Web DataWindow.

You use the TreeView DataWindow wizard to create a TreeView DataWindow object with one TreeView level. You add additional levels to the TreeView by selecting Rows>Create TreeView Level from the menu bar.

This sample TreeView DataWindow uses the department and employee tables in the EAS Demo database and has two TreeView levels. The first level is the department name. The second level is the city where each employee resides:

	Employee ID	First Name	Last Name	Status	Salary	Start Date	Health Insurance	Sex
+	Finance		Total Employees: 5					
-	Marketing		Total Employees: 4					
-	Concord							
	1576	Scott	Evans	Active	\$68,940.00	12/30/1999	<input checked="" type="checkbox"/> Health Insurance	<input type="radio"/> M <input type="radio"/> F
	207	Jane	Francis	Active	\$53,870.00	08/04/1998	<input checked="" type="checkbox"/> Health Insurance	<input type="radio"/> M <input type="radio"/> F
+	Needham							
+	Waltham							
-	R & D		Total Employees: 13					
+	Belmont							
-	Burlington							
	453	Andrew	Rabkin	Active	\$64,500.00	12/13/1992	<input checked="" type="checkbox"/> Health Insurance	<input type="radio"/> M <input type="radio"/> F
	316	Lynn	Pastor	Active	\$74,500.00	10/24/1992	<input checked="" type="checkbox"/> Health Insurance	<input type="radio"/> M <input type="radio"/> F
-	Concord							
	445	Kim	Lull	Active	\$87,900.00	12/13/1992	<input checked="" type="checkbox"/> Health Insurance	<input type="radio"/> M <input type="radio"/> F
+	Houston							
+	Lexington							
-	Lexington							
	529	Dorothy	Sullivan	Active	\$67,890.00	08/03/1993	<input checked="" type="checkbox"/> Health Insurance	<input type="radio"/> M <input type="radio"/> F
+	Milton							
+	Natick							
+	Waltham							
+	Wellesley							
+	Sales		Total Employees: 6					
+	Shipping		Total Employees: 1					

When you use a TreeView DataWindow, you click the state icon to expand or collapse a node. The state icon is a plus (+) sign when the node is collapsed and a minus (-) sign when it is expanded. When a node is expanded, connecting lines display by default to show more detail and indicate how the parent data connects with child data. When a node is collapsed, only the parent data displays; the detail data does not display.

Properties, events,
and methods

You can set TreeView DataWindow properties to customize the TreeView style, use methods to expand and collapse TreeView nodes, and use events that are fired when a node is expanded or collapsed. For reference information about properties, events and methods, see the following topics in the *DataWindow Reference* or the online Help:

- Properties
 - CollapsedTreeNodeIconName
 - DefaultExpandToLevel
 - ExpandedTreeNodeIconName
 - Indent
 - SelectNodeByMouse
 - ShowConnectLines
 - ShowLeafNodeConnectLines

ShowTreeNodeIcon
StateIconAlignMode
TreeNodeIconName

- Events
 - Expanded
 - Expanding
 - Collapsed
 - Collapsing
 - TreeNodeSelected
 - TreeNodeSelecting
- Methods
 - Expand
 - Collapse
 - ExpandAll
 - CollapseAll
 - ExpandAllChildren
 - CollapseAllChildren
 - ExpandLevel
 - CollapseLevel
 - IsExpanded
 - SelectTreeNode

For more information

For more information, see “TreeView presentation style” in the *User’s Guide*.

RichText enhancements

The RichText DataWindow presentation style and the RichTextEdit control both use a new rich text editor. These changes are described in the section “New and updated PowerBuilder controls” on page 11.

Decimal support in DataWindow expressions

Decimal datatypes are supported in the DataWindow, but in previous releases of PowerBuilder, decimal values were converted to doubles in DataWindow expressions. PowerBuilder 10.5 adds support for decimal values in DataWindow expressions and new DataWindow expression functions to support decimal values. You can also use decimal values as retrieval arguments. In PowerBuilder 10.5, the decimal datatype supports up to 28 digits.

The following arithmetic operators now return a decimal value if both operands have a datatype of decimal:

Operator	Meaning	Example
+	Addition	Subtotal + Tax
-	Subtraction	Price - Discount
*	Multiplication	Quantity*Price
/	Division	Discount/Price

If either operand is not a decimal, the returned value is converted to a double datatype. The exponentiation operator (^) continues to return a double.

Relational operators that operate on numeric values (including =, >, <, <>, >=, and <=) can take decimal operands. The precision of the decimal operand is maintained in comparisons.

The following functions return a decimal datatype if their arguments are decimals: Sum, CumulativeSum, Avg, Median, First, Last, Max, Min, Large, Small, Var, VarP, Mod, Mode, Abs, Case, If.

The following new functions return a decimal result instead of a double: CrosstabAvgDec, CrosstabMaxDec, CrosstabMinDec, and CrosstabSumDec.

The new Dec function converts a string to a decimal. You can also append the letter D in upper- or lowercase to identify a number as a decimal constant in a DataWindow expression. For example, 2.0d and 123.456789012345678901D are treated as decimals.

Decimal and Decimal array have been added to the list of types in the Specify Retrieval Arguments dialog box.

For descriptions of the new functions, see the *DataWindow Reference* or the online Help.

New style for drop-down DataWindows in Web DataWindows

When you tab to a column that uses the drop-down DataWindow edit style, you can use the arrow keys on the keyboard to change its value. If you click the column, the drop-down DataWindow displays so that you can scroll to a different value and click to select it.



Previously, clicking the column in a Web DataWindow displayed all the rows in the DataWindow. The new style allows more accurate rendering of colors and multiple columns.

You set the display properties for the column on the Edit page in the Properties view in the DataWindow painter. The Width of DropDown property sets the width of the drop-down display to a size that is a percentage of the width of the column. For example, 300 sets the display width to three times the column width.

The new behavior uses inline frames (iFrames), which might increase the volume of markup generated. For DataWindow objects that make heavy use of drop-down DataWindows, you might save bandwidth by using the previous behavior, in which the drop-down DataWindows are generated in HTML select elements. To do so, clear the Generate DDDW Frames check box on the Web Generation page with the Format to Configure option set to HTML/XHTML. You can view the generated source from each technique and save it to a file, then compare file sizes to determine which is best for your DataWindow objects.

For more information, see the description of `HTMLGen.property` in the *DataWindow Reference* or the online Help.

New `HTMLGen.PagingMethod` property for Web DataWindows

To set the `HTMLGen.Method` property, select HTML/XHTML in the Format To Configure drop-down list on the Web Generation page in the Properties view in the painter and then select a value from the Paging Method drop-down list. You can also set the property in code.

The `HTMLGen.PagingMethod` property takes a value of the `PagingMethod` enumerated variable. The default is `PostBack!`

Value	Numeric value	Description
PostBack!	0	Paging operations are posted to the server.
CallBack!	1	Paging operations use client-side script callbacks. This option is not available for the HTMLGenerator component.
XMLClientSide!	2	Paging operations are performed on the client by retrieving the full XML result set and performing an XSLT retransformation of the cached stylesheet. You must use the XML rendering format to use this option.

For more information, see the description of HTMLGen.property in the *DataWindow Reference* or the online Help.

Drop-down calendar DataWindow option

The drop-down calendar DataWindow option is available for use on any DataWindow column with an EditMask and a Date, DateTime, or TimeStamp datatype. The DDCalendar EditMask property option allows for separate selections of the calendar month, year, and date. This option can be set in a check box on the Edit tab of the DataWindow painter Properties view when a column with the EditMask edit style is selected. It can also be set in code, as in this example for the birth_date column:

```
dw_1.Modify("birth_date.EditMask.DDCalendar='Yes'")
```

The range for years in Windows applications is 1753 to 3000. You can set the following properties to control the display of the calendar in a script or on the Other page in the Properties view for the column:

Painter option	Property
Drop Align Right	Column.Editmask.ddcal_alignright
CalendarBackColor	Column.Editmask.ddcal_backcolor
CalendarTextColor	Column.Editmask.ddcal_textcolor
CalendarTitleBackColor	Column.Editmask.ddcal_titlebackcolor
CalendarTitleTextColor	Column.Editmask.ddcal_titletextcolor
CalendarTrailingTextColor	Column.Editmask.ddcal_trailingtextcolor

To make sure that dates selected with the drop-down calendar option are displayed with the desired edit mask for Web DataWindows, you should specify that the Client Formatting option be included with the static JavaScript generated and deployed for the DataWindow. To conserve bandwidth, JavaScript for client formatting is not included by default. To include this script, you can select the Client Formatting check box on the Web Generation page of the DataWindow Properties view. If you do not include script for client formatting, the drop-down calendar uses a default edit mask to display the column data based on the client computer's default localization settings.

To navigate in the drop-down calendar, a user can:

- Click the arrows in the top corners to move from month to month
- Click the month to display a list of months, then click a month to select it
- Click the year to display a spin control, then use the spin control's arrows to select a year
- Click a date to select the date and close the calendar
- Press the Esc key to close the calendar without changing the selection

For more information, see the description of EditMask.property in the *DataWindow Reference* or the online Help.

Autosize height property on all DataWindow bands

In PowerBuilder 10.5, the Height.Autosize property can be set on any band of a DataWindow. In earlier PowerBuilder releases, this property was available only for the Detail band.

Restrictions on the Height.Autosize property

- The Height.Autosize property is not supported on DataWindows with Graph, Label, OLE, or Rich Text presentation styles.
 - Nested report overflow to the next page is supported in Detail bands only.
 - Bands cannot be autosized if autosizing would preclude the display of at least one Detail band row per page.
-

Selecting the Height.Autosize property sets the minimum height for the band, allowing it to display the entire content of a picture or a nested report. You can select this property on the General tab of the Properties view for a DataWindow band. At runtime you can modify the Height.Autosize property for a DataWindow band in PowerScript®:

```
dw_2.modify ("DataWindow.header.height.autosize=yes")
dw_2.modify ("DataWindow.footer.height.autosize=no")
```

For more information, see the description of Bandname.property in the *DataWindow Reference* or the online Help.

New property to suppress group headers on page breaks

PowerBuilder 10.5 includes a new property that lets you suppress the group headers that display in a grouped DataWindow object after a page break. You can set this property on group header bands only.

When a group listing straddles a page break, all group headers for which you set this property are suppressed. The suppressed headers do not display at the top of the page. When all group headers are suppressed, the group displays as a continuous listing across the straddled pages. However, if the page break coincides with the start of a new group, only headers above the new group header are suppressed.

You enable the suppress group header property by selecting the Suppress Group Header check box in the Properties view for a group header band. You can verify the property setting and modify it at runtime with the Describe and Modify methods:

```
string s
s = dw_1.describe("datawindow.header.1.suppress")
dw_1.modify("datawindow.header.1.suppress=no")
```

For more information, see the description of Bandname.property in the *DataWindow Reference* or the online Help.

New property to hide blue outline in Print Preview

By default, a blue line displays in Print Preview mode to show the location of the margins. You can hide this line by clearing the Print Preview Shows Outline check box on the Print Specifications page in the Properties view in the DataWindow painter.

For more information, see the description of Print.Preview.property in the *DataWindow Reference* or the online Help.

New dbAlias DataWindow object property

The dbAlias DataWindow object property allows you to get column names that use the alias for a table name in a script. Use this property if you have defined an alias for a table name in the SELECT statement for a DataWindow object and you want to obtain the column name with the table alias so that you can manipulate the update WHERE clause dynamically.

For more information, see the description of dbAlias in the *DataWindow Reference* or the online Help.

New ResetInk, SaveInk, and SaveInkPic DataWindow methods

The SaveInk method saves overlay ink to a file or blob from an InkPicture control in a DataWindow. The SaveInkPic method saves a picture, and optionally overlay ink, to a file. ResetInk clears the ink from the control.

For more information, see ResetInk, SaveInk and SaveInkPic in the *DataWindow Reference* or the online Help.

User interface enhancements

PowerBuilder 10.5 uses new icons and offers a contemporary style for menus and toolbars.

New icons

The icons in the PowerBuilder user interface have been updated, and the new icons are available for use in your applications. When you migrate an application to PowerBuilder 10.5, any PowerBuilder stock icons it uses are updated automatically.

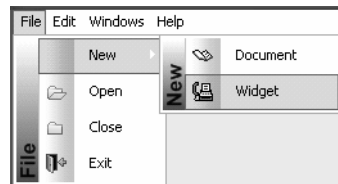
Menu and toolbar enhancements

In PowerBuilder, you use the Menu painter to create menus and toolbars. A toolbar is associated with a menu, and its toolbar buttons act as shortcuts for choosing items from the menu.

Menu styles You can customize the display of menus and toolbars in applications that you create with PowerBuilder. Menus that you import or migrate from earlier versions of PowerBuilder use the Traditional menu style by default.

Menus with a Contemporary style have a three-dimensional menu appearance similar to those in Microsoft Office 2003 and Visual Studio 2005, and can include bitmap and menu title bands.

Toolbar style properties Toolbars have style properties that you can change at design time on the top-level menu object. You can modify these properties only if you select `contemporarytoolbar!` as the toolbar style for the top-level menu object.



In previous versions of PowerBuilder, you could use toolbars only in MDI windows. In PowerBuilder 10.5, you can use a toolbar in a window of type `Main!` as well.

For more information about menus and toolbars, see Chapter 14, “Working with Menus and Toolbars,” in the *User’s Guide*.

New and updated PowerBuilder controls

RichText support has been changed, and a DatePicker control, a DropDownCalendar property for EditMask controls, and a new event for the MonthCalendar control have been added.

Rich Text enhancements

RTF is a standard for specifying formatting instructions and document content in ASCII documents. An editor that supports RTF interprets a document’s instructions and displays the text with formatting.

PowerBuilder 10.5 uses a new rich text editor to support the RichTextEdit control and the RichText DataWindow presentation style. The new editor brings a modern look and includes some new features, including the ability to name and use formatting styles. Most of the properties and functions of rich text objects in previous versions of PowerBuilder continue to be supported by the new rich text editor. When you import rich text objects from previous versions of PowerBuilder, any obsolete properties and functions are ignored.

The new rich text editor supports a subset of the RTF specification version 1.6.

The new rich text editor features are described in the following sections:

- Rich text editor user interface changes
- RichTextEdit control changes
- RichText DataWindow presentation style changes

Rich text editor user interface changes

Rulers, toolbars, and status bars

Like the earlier rich text editor, the new editor includes a toolbar and ruler that you can hide or display at design time or runtime. If you enable the PopMenu property on a RichTextEdit control or rich text DataWindow, end users of your PowerBuilder applications can also opt to display or hide the toolbar or ruler.

The new editor includes a status bar that you can enable at design time or runtime, or that the users of your rich text applications can opt to display or hide. The rich text editor no longer includes a separate tab bar. End users of a rich text control can set tabs for the control in the Tabs dialog box or directly on the ruler.

Rich text editor toolbar items

The toolbar for the rich text editor has a contemporary flat-style look. It includes drop-down lists for formatting style and zoom percent, and toolbar buttons for adding bullets or sequential numbers to paragraphs in the rich text control.



Users can set document properties in text format dialog boxes that they open by double-clicking on blank areas of the toolbar.

To open this dialog box	Double-click this area on the toolbar
Styles	Space in front of the Style drop-down list or the space trailing it
Font	Space in front of any of the font-related fields—font face drop-down list, font size drop-down list, or character attributes (bold, italic, underline)—or the space trailing them
Paragraph	Space in front of or trailing the alignment buttons
Bullets and Numbering	Space in front of or trailing the bullet and numbering buttons
Tabs	Space in front of or trailing the tab and paragraph buttons

Viewing page margins

The new editor lets you view headers and footers at the same time as the text body. When you opt to include headers and footers, the editor displays each page of text with borders for the top and bottom margins, and with gaps between adjoining pages. This gives the rich text control or DataWindow object a three-dimensional appearance.

You can decide whether to allow headers and footers at design time only. If you do not allow headers and footers, the pages of text display as a continuous sheet without borders or margins, although you can still see separate pages in the print preview mode.

Dialog box changes

The Styles dialog box and the Bullets and Numbering dialog box are new to PowerBuilder 10.5. Users can create, modify, or delete styles in the Styles dialog box. The Font dialog box now includes line position items (normal, superscript, subscript). The Paragraph dialog box adds indent and distance-setting capabilities. In earlier versions of PowerBuilder, the Tabs dialog box was included in the Paragraph dialog box, but it is now separate.

RichTextEdit control changes

Modified properties The table that follows lists new and modified properties of the PowerBuilder 10.5 RichTextEdit control:

Property	Datatype	Description
StatusBar	Boolean	<p>New property that specifies whether a status bar displays below the editing area of a rich text control. Values are:</p> <ul style="list-style-type: none"> • True Status bar is visible. • False Status bar is hidden. <p>You can set the value of this property in the painter at design time or in PowerScript at runtime. If the pop-up menu on a rich text control is enabled, end users can turn the status display on or off from the pop-up menu.</p>
ControlCharsVisible	Boolean	<p>New property replacing the following obsolete properties: ReturnsVisible, SpacesVisible, and TabsVisible. It specifies whether control characters are visible in the rich text control. Values are:</p> <ul style="list-style-type: none"> • True Control characters are visible. • False Control characters are hidden.
HeaderFooter	Boolean	<p>Although this property remains substantially the same as in previous versions of PowerBuilder, setting this to “true” causes the header and footer to display in the same view as the body of the rich text control. (In the previous rich text editor, you could not display the header and footer in the same view with the text body.) You must still set this property at design time only.</p>

Property	Datatype	Description
InputFieldBackColor	Long	Although this property remains substantially the same as in previous versions of PowerBuilder, the new rich text editor allows a user to override the default background color on individual input fields. The changes that a user makes for an input field in a font properties dialog box do not affect the default background color of other input fields in a rich text document.
Pointer	Enumerated value	The new rich text editor supports most but not all of the stock pointers available in the old rich text editor. The AppStarting!, Help!, and No! pointers are no longer supported. Users can still select one of the following pointers: Arrow!, Cross!, HourGlass!, Hyperlink!, IBeam!, Size!, SizeNESW!, SizeNS!, SizeWE!, SizeNWSE!, SizeWE!, and UpArrow!. If no value (or an unsupported value) is specified, the default pointer is "IBeam!" when the rich text control is editable and "Arrow!" when the control is read-only.

Obsolete properties The following properties are not supported in the new rich text editor: ReturnsVisible, SpacesVisible, TabBar, TabsVisible, and UndoDepth. If you set these properties on a RichTextEdit control, they are ignored by the new editor. The maximum undo depth in the new rich text editor is 50. This value cannot be changed at either design time or runtime.

Modified functions

The following table lists functions that have been modified in PowerBuilder 10.5:

Function	Description
InsertDocument	<p>This function supports the following additional file types:</p> <ul style="list-style-type: none"> • FileTypeHTML! – The file being opened is in HTML format (HTM or HTML) • FileTypeDoc! – The file being opened is in Microsoft Word format (DOC) <p>Also, you can replace, but not add to, existing headers and footers with headers and footers from a document that you insert using this function. To replace an existing document, including its headers and footers, you must set the <i>clearflag</i> argument to “true”.</p>
PasteRTF	<p>Due to a minor quirk in the new rich text editor, the PasteRTF function always ignores the last paragraph tag at the end of an RTF string that you paste in a control when there is more than one paragraph tag.</p>
SaveDocument	<p>This function supports the following additional file types:</p> <ul style="list-style-type: none"> • FileTypeDoc! – Save the file in Microsoft Word format • FileTypeHTML! – Save the file in HTML format • FileTypePDF! – Save the file in PDF format
ScrollNextPage and ScrollPriorPage	<p>These functions work in the RichTextEdit control edit mode only when the HeaderFooter property of a rich text control is selected. They work in print preview mode regardless of the HeaderFooter property setting and they work for the RichText DataWindow control in edit mode whether or not the DataWindow has header or footer bands.</p>
SelectedLength, SelectText, and SelectTextAll	<p>These functions count carriage returns and line feeds as single characters rather than as separate characters. Other functions, such as LineLength, SelectTextLine, SelectTextWord, and TextLine do not include carriage returns or line feeds in character counts.</p>
SelectedLine	<p>Obtains the number of the line that contains the insertion point in an editable control. The insertion point now moves to the next line if the current line contains a carriage return.</p>

Function	Description
SetTextStyle	<p>A new overloaded syntax removes the boolean arguments for superscript and subscript formatting. Users can set superscript or subscript styles from the properties dialog box that they access by double-clicking blank areas of the toolbar. The overloaded syntax is:</p> <pre data-bbox="663 388 1184 465">integer rtename.SetTextStyle (boolean bold, boolean underline, boolean italic, boolean strikeouts)</pre>
ShowHeadFoot	<p>A new overloaded syntax adds a boolean argument for specifying whether the insertion point (caret) for editing the header/footer panel is in the header or the footer section. Values are:</p> <ul data-bbox="633 604 1009 661" style="list-style-type: none"> • True Caret is in the header section. • False Caret is in the footer section. <p>The second argument is ignored if the first argument is false. The second argument defaults to “true” if a value is not provided. The overloaded syntax for this function is:</p> <pre data-bbox="663 774 1184 821">integer rtename.ShowHeadFoot (boolean editheadfoot, boolean headerfooter)</pre>

Modified events The InputFieldSelected and the PictureSelected events are no longer triggered when the user presses the Enter key. The PrintHeader and PrintFooter events are obsolete. They are no longer triggered under any circumstance.

Setting a default font The Properties view for a RichTextEdit control now includes a Font tab page where you can set default font characteristics for the control. When the control first displays at runtime, and you include the toolbar with a RichTextEdit control, the toolbar indicates the default font characteristics that you selected on the Font tab page at design time. Although the application user can change fonts at runtime, or you can use PowerScript to change the font style, you can set the default font at design time only.

Migration issue When you migrate applications created in older versions of PowerBuilder, the InputFieldsVisible property in RichTextEdit controls and in RichText DataWindow objects is automatically set to “false” in the migrated applications. You must set this property to “true” to see data in the input fields. You must set this property and the InputFieldNamesVisible property to “true” to see text labels for the input fields in a rich text control.

RichText DataWindow presentation style changes

The RichText DataWindow presentation style uses the new rich text editor. Property and method changes to the RichTextEdit control also apply to the RichText DataWindow.

In PowerBuilder 10.5, the Preview view of the DataWindow painter for a RichText DataWindow is “WYSIWYG” (what you see is what you get) and read-only. When you set content in Detail view for the DataWindow header and footer, the Preview view displays the header and footer at the same time as the DataWindow row details. In previous versions of PowerBuilder, you were required to open the Print Preview view to preview the document body and page margins simultaneously.

With the focus in the Preview view of the DataWindow painter, you can still select the File>Print Preview menu item to preview the document before it is sent to the printer. Since the preview is WYSIWYG, the panel to control margins in the Print Preview view has been removed. In previous versions of PowerBuilder, the panel for the page margins also included buttons to scroll through the pages of the DataWindow document. In PowerBuilder 10.5, you must press the Page Up and Page Down keys to change pages in the Print Preview view.

When you place focus in the Preview view for a DataWindow with headers and footers, you also enable the Design>Header/Footer menu item. Selecting this menu item opens the Select Band To Activate dialog box. Radio buttons in this dialog box prompt you to select a band to activate. After you click OK, focus changes to the band in the Preview view that corresponds to the radio button that you select.

Input fields in the new rich text editor are always multiline-capable and never size-fixed, so the MultiLine and FixedSize properties for rich text input fields are obsolete. They are no longer assignable from the DataWindow painter.

At runtime, the RichText DataWindow also displays in WYSIWYG format—with the headers and footers displaying in the same window as the document body. The toolbar and its related dialog boxes, and the ruler and status bar are also available at runtime. Otherwise, the runtime functionality remains unchanged.

DatePicker control

The DatePicker control provides an easy way for a user to select a single date. The user can choose to edit the date in the control or to select a date from a drop-down calendar. The calendar is similar to the MonthCalendar control, which can be used to select a range of dates. You can also specify that the DatePicker control has an up-down control instead of a drop-down calendar.

You can also use a drop-down calendar in Date or DateTime EditMask controls and DataWindow columns that display dates. See “DropDownCalendar property for EditMask controls” and “Drop-down calendar DataWindow option” for more information.

To add a DatePicker control to a window or visual user object, select it from the Insert>Control menu and drop it on the window or object. The DatePicker control has several properties that allow you to work with times as well as dates, but to pick dates only, you can use the drop-down calendar.

You can set initial properties for the appearance and behavior of the control in the Properties view. Properties that apply to the drop-down calendar portion of the control are similar to the properties that apply to the MonthCalendar control and display on the Calendar page in the Properties view. For example, you can choose which day of the week displays as the first day in the week, whether the current date is circled, and whether a “Today Section” showing the current date displays at the bottom of the calendar.

For more information

For more information, see “DatePicker” in the chapter on using controls in the *User’s Guide*. For a complete list of DatePicker properties, events, and functions, see DatePicker control in *Objects and Controls* in the online Help.

DropDownCalendar property for EditMask controls

You can use a drop-down calendar that is similar to the DatePicker control in EditMask controls that have a Date or DateTime edit mask. The user can choose to edit the date in the control or to select a date from a drop-down calendar. The range for years in Windows applications is 1753 to 3000.

To specify that an EditMask control use a drop-down calendar to display and set dates, select the Drop-down Calendar check box on the Mask page in the Properties view. You can set display properties for the calendar on the Calendar page. Users navigate and select dates within the calendar as they do in the calendar in a DatePicker control.

For a description of the DropDownCalendar property, see DropDownCalendar in the chapter on properties in *Objects and Controls*.

DateSelected event for the MonthCalendar control

The new DateSelected event for the MonthCalendar control fires *only* when a user explicitly selects a date using the mouse. The existing DateChanged event fires whenever a date is changed—when a user selects a date using the mouse, when the date is changed in a script, and when the user uses the arrow keys on the keyboard to move to another date, or the arrows on the control to scroll to another month.

For more information, see the DateSelected event in the *PowerScript Reference* or the online Help.

Language enhancements

In PowerBuilder 10.5, you can import PowerBuilder extensions from the pop-up menu for a PBL in the System Tree. The Decimal datatype supports more digits, a new Byte datatype is available, and there are new properties for the OLERuntimeError system object.

Importing PowerBuilder extensions

In PowerBuilder 10.5, you can import PowerBuilder extensions into a library in your application using a pop-up menu item, and authors of PowerBuilder extensions no longer need to supply a PBD file with the extension file.

A PowerBuilder extension is an extension to PowerBuilder functionality created using the PowerBuilder Native Interface (PBNI). Some PowerBuilder extensions are provided with PowerBuilder, including the PBDOM, EJB client, SOAP client for Web services, and UDDIProxy extensions. You can also find PowerBuilder extensions on the PowerBuilder Samples and Utilities Web site at <http://PowerBuilder.CodeXchange.sybase.com>.

In previous releases of PowerBuilder, the author of a PowerBuilder extension had to build the extension file (a *.pbx* or *.dll* file) in a C++ development environment, then use the pbx2pbd utility to create a PBD file. The user of the extension had to put the extension file in the application's search path and add the PBD file to the target's library list.

In PowerBuilder 10.5, the author needs to supply only the extension file, and the user can import the descriptions of objects in the extension file into an existing library using the Import PB Extension item on the library's pop-up menu in the System Tree. The PBX or DLL file must be in the application's search path at runtime.

In the System Tree, the imported extension displays as a user object in the library and can be used like any other user object.

Importing the PBX or DLL file converts the description of the extension into PowerScript and adds it to the PBL as source code. The new keyword `native` in the source code indicates that the PowerBuilder type was defined in an extension.

Decimal datatype support

In previous releases of PowerBuilder, the Decimal datatype in PowerScript supported up to 18 digits. In PowerBuilder 10.5, the Decimal datatype supports up to 28 digits.

Byte datatype support

PowerBuilder 10.5 adds support for the Byte datatype. The Byte datatype is the same as the Uint datatype except that it has a more restricted range: from 0 to 255. For interaction with Java components or Web services, the PowerScript Byte datatype maps directly to Byte datatypes in the Java and C# or .NET languages.

PowerBuilder 10.5 also adds the following functions for working with Byte datatypes:

Function	Description
Byte (<i>stringorblob</i>)	Converts a numeric datatype, including a string whose value is a number, into a Byte datatype, or obtains a Byte value stored in a blob
GetByte (<i>blobvariable, n, b</i>)	Extracts data of type Byte from a blob variable
SetByte (<i>blobvariable, n, b</i>)	Sets data of type Byte for a blob variable

For more information, see `Byte`, `GetByte`, and `SetByte` in the *PowerScript Reference* or the online Help.

New OLERuntimeError properties

OLERuntimeError is a system object that inherits from the RuntimeException object (which inherits, in turn, from the Throwable object). OLERuntimeError is thrown automatically when an OLE error occurs that is not handled by an `ocx_error`, `ExternalException`, or `Error` event script. Information from the `ExternalException` and `Error` events is propagated to the OLERuntimeError object.

In PowerBuilder 10.5, the OLERuntimeError object has four properties that are not present in its ancestor objects:

OLERuntimeError property	Datatype	Description
Description	String	Description of the error intended for the end user
HelpContext	UnsignedLong	Help context ID of the topic in a Help file that contains information about the error
HelpFile	String	Complete path and file name of a Help file with information about the error
Source	String	Name of the exception source—typically, an application name

Database connectivity enhancements

Database connectivity enhancements include database tracing enhancements, support for Adaptive Server® Enterprise version 15, and ADO.NET support for Adaptive Server and Oracle.

Database tracing enhancements

PowerBuilder 10.5 includes accuracy and ease-of-use enhancements to the main database trace utility, *PBTRAI05.DLL*, and provides a second trace utility, *PBTRS105.DLL*. The newer utility logs the date and time (rather than the time elapsed) of each SQL statement issued to the database. This utility can use a timestamp from the time zone of the DBMS server computer, allowing clients in separate time zones to merge trace output files in a meaningful fashion.

These enhancements were introduced in PowerBuilder 10.2. For more information, see the section on “Using the Database Trace tool” in the *Connecting to Your Database* book.

Adaptive Server Enterprise 15 support

The native interface for Adaptive Server® Enterprise (*pbsyc105.dll*) can be used with Adaptive Server version 15 and has been enhanced to support several new features in version 15 of Adaptive Server.

Scrollable cursors	Adaptive Server allows both scrollable and nonscrollable cursors, which can be either semi-sensitive or insensitive. “Scrollable” means that you can scroll through the cursor result set by fetching any, or many, rows, rather than one row at a time; you can also scan the result set repeatedly. The <code>CursorType</code> database parameter lets you specify the type of cursor to use. For more information, see <code>CursorType</code> in the PowerBuilder online Help.
Unitext support	PowerBuilder supports the variable-length <code>UniText</code> datatype, which can hold up to 1,073,741,823 Unicode characters (2,147,483,646 bytes). You can use <code>UniText</code> anywhere you use the text datatype, with the same semantics. <code>Unitext</code> columns are stored in UTF-16 encoding, regardless of the Adaptive Server default character set. The <code>Unitext</code> datatype uses UTF-16 encoding to support Unicode text and maps to the PowerScript string datatype.
Bigint support	PowerBuilder supports the 64-bit integer signed <code>BigInt</code> datatype, which maps to the PowerScript <code>longlong</code> datatype, and the unsigned <code>BigInt</code> datatype, which maps to the PowerScript decimal datatype.
Unsigned integer datatypes	In addition to the unsigned <code>BigInt</code> datatype, PowerBuilder also supports two other unsigned integer datatypes introduced in Adaptive Server 15.0. In this release, the unsigned integer datatypes are supported when connecting to Adaptive Server using the SYC and JDBC drivers.

These datatypes allow you to extend the range of the positive numbers for the existing integer types without increasing the required storage size. That is, the signed versions of these datatypes extend both in the negative direction and the positive direction. However, the unsigned versions extend only in the positive direction. Table 1-2 describes the range of the signed and unsigned versions of these datatypes.

Table 1: Ranges for signed and unsigned integer datatypes

Datatype	Range of signed datatypes	Range of unsigned datatypes
<code>BigInt</code>	Integers between -2^{63} and $2^{63} - 1$ (from -9,223,372,036,854,775,808 to +9,223,372,036,854,775,807, inclusive)	Integers between 0 and 18,446,744,073,709,551,615
<code>Int</code>	Integers between -2^{31} and $2^{31} - 1$ (-2,147,483,648 and 2,147,483,647), inclusive	Integers between 0 and 4,294,967,295
<code>SmallInt</code>	Integers between -2^{15} and $2^{15} - 1$ (-32,768 and 32,767), inclusive	Integers between 0 and 65535

Using Adaptive Server 15

To use Adaptive Server 15, you must install Open Client version 15 on the client computer and set the Release parameter to 15 to establish an Open Client 15 client context. You can specify the Release parameter on the Connection page of the Database Profile Setup dialog box for Adaptive Server or in code:

```
SQLCA.DBParm="Release='15' "
```

The Open Client 15 client context can be used with Adaptive Server 11.x, 12.x, and 15.x.

In PowerBuilder 10.5, the SYC driver links to the appropriate version of the client libraries dynamically and the Open Client context is released when all connections are closed. If you open multiple connections, the first Open Client context established is used for all connections. If you need to establish a new Open Client context in the development environment, close all open connections and establish a new connection with the Release parameter set to the context you require.

ADO.NET support for Sybase Adaptive Server and Oracle

PowerBuilder 10.5 provides native access to Adaptive Server Enterprise and Oracle using ADO.NET. You can connect to a database using the ADO.NET data providers and perform operations using native syntax. You can create a new connection or share an existing connection object for transaction management.

PowerBuilder now supports the following ADO.NET namespaces and data providers:

Data Provider	Namespace
.NET Framework Data Provider for OLE DB	System.Data.OleDb
.NET Framework Data Provider for SQL Server	System.Data.SqlClient
Oracle Data Provider for .NET (ODP.NET)	Oracle.DataAccess.Client
Sybase ADO.NET Data Provider for Adaptive Server Enterprise (ASE)	Sybase.Data.AseClient

Support is provided in the *PBADO105.DLL* file and the *Sybase.PowerBuilder.Db.dll* and *Sybase.PowerBuilder.DbExt.dll* .NET assemblies. The ADO.NET drivers distributed with PowerBuilder are built with a specific version of the ADO.NET assemblies provided by Sybase and Oracle. You need to deploy *Sybase.PowerBuilder.DbExt.dll* with your application if you are using Oracle 10g or Adaptive Server 15 or later.

For more information, see Chapter 5, “Using the ADO.NET Interface,” in *Connecting to Your Database*.

Support for Sybase Adaptive Server temporary tables

PowerBuilder 10.5 allows you to create and edit temporary tables in the Database painter, SQL painter, or DataWindow painter when you use the PowerBuilder SYC native driver to connect to an Adaptive Server Enterprise database. Temporary tables persist for the duration of a database connection, residing in a special database called “tempdb”.

You add a temporary table to the tempdb database by assigning a name that starts with the # character when you create a new table in a PowerBuilder painter. (Temporary tables must start with the # character.)

After you create a temporary table, you can create indexes and a primary key for the table. If you define a unique index or primary key, you can execute INSERT, UPDATE, and DELETE statements for a temporary table. Selecting Edit Data from the pop-up menu of a temporary table retrieves data that you store in that table.

For more information, see the section on “Creating and editing Sybase Adaptive Server temporary tables” in the *User’s Guide*.

Support for Microsoft SQL Server 2005

The PowerBuilder 10.5 ODBC, OLE DB, and ADO.NET drivers have been tested with Microsoft SQL Server 2005. No new features in SQL Server 2005 are supported.

NCharBind database parameter for Oracle

The NCharBind database parameter for Oracle9i and Oracle 10g improves performance by allowing you to specify whether string variables should be treated as Char or NChar datatypes. The default is to treat string variables as Char datatypes. You can set this parameter on the Transaction tab page in the Database Profile Setup dialog box for Oracle9i and Oracle 10g. For more information, see NCharBind in the *Connection Reference* or the online Help.

Web Services enhancements

PowerBuilder 10.5 can use the .NET Web service engine to construct SOAP requests and parse the SOAP messages returned from a Web service.

For more information on building Web service applications, see Chapter 31, “Building a Web Services Client,” in *Application Techniques*.

About the .NET Web service engine

The .NET Web service engine can handle the XML-type array datatype. It also supports the transmission of security constraints in the header sections in the SOAP message envelope.

The .NET Web service engine supports the latest Web service standards. To use this engine, you must have the *wsdl.exe* Web service tool on the development computer. This tool is required to parse WSDL files and generate C# code for a .NET assembly. The *wsdl.exe* file installs with the .NET SDK. It is not required on deployment computers, although deployment computers must have the .NET 2.0 Framework to consume a Web service that depends on the .NET Web service engine.

Using the .NET Web service engine

Generating a .NET assembly

If you select the .NET Web service engine in the Web Service Proxy wizard, the wizard generates a .NET assembly (DLL) in addition to a proxy object. To use the Web service at runtime, you must deploy the wizard-generated DLL along with your application.

You can also select the .NET Web service engine in the Project painter for a new Web service proxy. If you select the .NET Web service engine on the Web Service tab of the Properties dialog box for the Web Service Proxy Generator, PowerBuilder attempts to generate an assembly DLL after you click Apply or OK. You cannot use the Properties dialog box to change the Web service engine for a proxy that you have already generated with the Web Service Proxy wizard.

Naming the DLL You can name the DLL generated by the Web Service Proxy wizard or by the Project painter in the Proxy Assembly Name text box. You do not need to include the DLL extension. The name of the wizard-generated assembly is *Web_service.DLL*, where *Web_service* is the name you provide in the Proxy Assembly Name field. If you do not provide a name, the assembly takes the name of the Web service to be consumed by the DLL. The assembly is generated in the current target directory.

Deploying the DLL You must deploy the DLL created for your Web service project to the directory where you deploy the client executable. You must also copy the *Sybase.PowerBuilder.WebService.Runtime.dll* and the *Sybase.PowerBuilder.WebService.RuntimeRemoteLoader.dll* system assemblies to this directory.

New extension class library

Although you use the same SOAP connection and exception-handling classes for the .NET Web service engine as for the EasySoap Web service engine, the classes that reference the .NET Web service engine require a different extension file or library.

For more information, see “Importing objects from an extension file” in *Application Techniques*.

Adding security information to the SOAP client header

You can use the SetOptions function of the SoapConnection class to include security information in the SOAP client header. Alternative methods provide a preferred way of adding security information.

For information on the alternative methods, see “New methods for the SoapConnection object” on page 28.

You can include the following options in the Options argument of the SetOptions function for a .NET Web service client:

.NET security option	Description
Domain	A string value for the Web domain to which the user belongs. This could be a domain name, such as “sybase.com”, or a computer name.
UseWindowsAuthentication	A “yes” or “no” value to determine whether to use “Integrated Windows Authentication.” The value you enter can be a boolean or a string. If this option is set to “yes,” you do not need to set the UserID, Password, or Domain options.
AuthenticationMode	A string value for the authentication mode you want to use. This can be “basic” or “digest”. AuthenticationMode values are described on the Microsoft MSDN Web site at http://msdn2.microsoft.com/en-us/library/w67h0dw7.aspx .
CertificateFile	A string value for the certificate file or files that you want to send from the Web service client to the server. The string value could include local files with a full path and URLs to remote certificate files. You must use a semicolon as a separator for multiple files.

With the .NET Web service engine, you cannot use the SoapLog or ConnectionCache parameters in the Options argument of the SetOptions method.

Using the EasySoap Web service engine

Selecting the EasySoap Web service engine

You can still use the EasySoap Web service engine, which does not support some of the advanced features of the .NET Web service engine. The EasySoap Web service engine is retained for backward compatibility and for use with EAServer components deployed to UNIX computers.

You set the Web service engine that you want to use on the first page of the Web Service Proxy Wizard or on the Web Service tab of the Property sheet for a Web service project. The Use .NET Engine check box is cleared by default for new Web service projects. You must not select the check box if you are developing a Web service application that you intend to deploy to UNIX computers.

New methods for the SoapConnection object

Methods for .NET SOAP connection options

Instead of using the SetOptions method for setting all SOAP connection options in a single argument, you can use separate methods to apply the specific options that your Web service application requires. The following methods can be used to supplement or replace the SetOptions method for .NET Web services:

SoapConnection method	Description	Syntax
RemoveAuthentication	Clears Basic, Digest, Domain, and Integrated Windows Authentication information from the SoapConnection object.	<i>conn.RemoveAuthentication ()</i>
SetBasicAuthentication	Sets client identification information for Web services that require Basic authentication.	<i>conn.SetBasicAuthentication (string domain, string userID, string password)</i>
SetClientCertificateFile	Sets the certificate file or files to use to connect to a Web service. Filename can be a list of files separated by semicolons.	<i>conn.SetClientCertificateFile (string filename)</i>
SetTimeout	Sets the client-side timeout value for a SOAP connection.	<i>conn.SetTimeout (long seconds)</i>

SoapConnection method	Description	Syntax
UseIntegratedWindows Authentication	Determines whether the SoapConnection object uses Integrated Windows Authentication to connect to a Web service.	<code>conn.UseIntegratedWindowsAuthentication</code> (boolean <i>useIWA</i>)

All SoapConnection methods return a value of the long datatype. They return 0 for success and 50 for failure. The *conn* variable in the method syntaxes holds an instantiated SoapConnection object.

Methods for EasySoap SOAP connection options

The following methods can be used to supplement or replace the SetOptions method for EasySoap Web services:

SoapConnection method	Description	Syntax
SetSoapLogFile	Sets the name of a file for logging raw SOAP messages	<code>conn.SetSoapLogFile</code> (string <i>filename</i>)
SetTimeout	Sets the client-side timeout value for a SOAP connection	<code>conn.SetTimeout</code> (long <i>seconds</i>)
UseConnectionCache	Determines whether a connection cache is used for the Web service connection	<code>conn.UseConnectionCache</code> (boolean <i>cache</i>)

All SoapConnection methods return a value of the long datatype. They return 0 for success and 50 for failure. The *conn* variable in the method syntaxes holds an instantiated SoapConnection object.

For more information on using the new SoapConnection methods, see the *PowerBuilder Extension Reference* in the online Help or the *Release Bulletin*.

Alternative method for setting a proxy server connection

If your application users connect to the Internet through a proxy server, you can use the SetProxyServer method instead of the SetProxyServerOptions method to assign proxy server connection parameters.

The SetProxyServer method has two syntaxes:

- `conn.SetProxyServer` (string *address*, string *userID*, string *password*)
- `conn.SetProxyServer` (string *hostname*, long *port*, string *userID*, string *password*)

All SoapConnection methods return a value of the long datatype. They return 0 for success and 50 for failure. The *conn* variable in the method syntaxes holds an instantiated SoapConnection object.

The first syntax has three string arguments. For this syntax, you set the value of the proxy server port in the *address* argument along with the host name of the proxy server. You can also include a proxy endpoint in the *address* argument using the format: *http://hostname:port/path*.

The second syntax has arguments for the proxy server address, port, user ID, and password. The port argument takes a long datatype and the other arguments take string datatypes. These are the same components of the single string argument that you use with the older `SetProxyServerOptions` method.