



# Objects and Controls

VERSION 4.0

# PowerBuilder

Copyright © 1991-1994 by Powersoft Corporation.

All rights reserved.

First printed and distributed in the United States of America.

Information in this manual may change without notice and does not represent a commitment on the part of Powersoft Corporation.

The software described in this manual is provided by Powersoft Corporation under a Powersoft License agreement. The software may be used only in accordance with the terms of the agreement.

Powersoft Corporation ("Powersoft") claims copyright in this program and documentation as an unpublished work, revisions of which were first licensed on the date indicated in the foregoing notice. Claim of copyright does not imply waiver of Powersoft's other rights.

This program and documentation are confidential trade secrets and the property of Powersoft. Use, examination, reproduction, copying, decompilation, transfer, and/or disclosure to others are strictly prohibited except by express written agreement with Powersoft.

PowerBuilder, Powersoft, and SQL Smart are registered trademarks, and InfoMaker, Powersoft Enterprise Series, PowerMaker, PowerSQL, PowerViewer, and CODE are trademarks of Powersoft Corporation. DataWindow is a proprietary technology of Powersoft Corporation (U.S. patent pending).

1-2-3 is a registered trademark of Lotus Development Corporation. 386 is a trademark of Intel Corporation. ALLBASE/SQL and IMAGE/SQL are trademarks of Hewlett-Packard Company. AT&T Global Information Solutions and TOP END are registered trademarks of AT&T. CICS/MVS, DB2, DB2/2, DRDA, IMS, PC-DOS, and PL/1 are trademarks of International Business Machines Corporation. CompuServe is a registered trademark of CompuServe, Inc. DB-Library, Net-Gateway, SQL Server, and System 10 are trademarks of Sybase Corporation. dBASE is a registered trademark of Borland International, Inc. Graphics Server is a trademark of Bits Per Second Ltd. DEC and Rdb are trademarks of Digital Equipment Corporation. FoxPro, Microsoft, Microsoft Access, MS-DOS, and Multiplan are registered trademarks, and Windows and Windows NT are trademarks of Microsoft Corporation. INFORMIX is a registered trademark of Informix Software, Inc. INTERSOLV, PVCS, and Q+E are registered trademarks of INTERSOLV, Inc. ORACLE is a registered trademark of Oracle Corporation. PaintBrush is a trademark of Zsoft Corporation. PC/SQL-link is a registered trademark, and Database Gateway is a trademark of Micro Decisionware, Inc. Paradox is a registered trademark of Borland International, Inc. SQLBase is a registered trademark of Gupta Corporation. Watcom is a registered trademark of Watcom International Corporation. XDB is a registered trademark of XDB Systems.

December 1994

# Contents

<b>About This Manual .....</b>	<b>vii</b>
<b>1</b>	<b>Attributes, Events, and Functions for Windows ..... 1</b>
	Window..... 2
	Attributes ..... 2
	Events ..... 8
	Functions..... 11
<b>2</b>	<b>Attributes, Events, and Functions for Controls ..... 15</b>
	About controls..... 17
	Types of controls ..... 17
	Controls and their uses ..... 17
	Default control names..... 18
	Naming conventions ..... 19
	CheckBox ..... 20
	Attributes ..... 20
	Events ..... 23
	Functions..... 24
	CommandButton..... 25
	Attributes ..... 25
	Events ..... 28
	Functions..... 29
	DataWindow control..... 30
	Attributes ..... 30
	Events ..... 33
	Functions..... 36
	DropDownListBox ..... 49
	Attributes ..... 49
	Events ..... 53
	Functions..... 54

EditMask .....	57
Attributes.....	57
Events.....	62
Functions .....	63
Graph.....	66
Attributes.....	66
Events.....	69
Functions .....	70
GroupBox.....	74
Attributes.....	74
Functions .....	77
HScrollBar.....	78
Attributes.....	78
Events.....	80
Functions .....	81
Line.....	82
Attributes.....	82
Functions .....	83
ListBox.....	84
Attributes.....	84
Events.....	89
Functions .....	90
MultiLineEdit .....	93
Attributes.....	93
Events.....	98
Functions .....	98
OLE 2.0.....	101
Attributes.....	101
Events.....	104
Functions .....	105
Oval.....	107
Attributes.....	107
Functions .....	108
Picture.....	109
Attributes.....	109
Events.....	112
Functions .....	113
PictureButton.....	114
Attributes.....	114
Events.....	118
Functions .....	119
RadioButton .....	120
Attributes.....	120
Events.....	123
Functions .....	124



Rectangle .....	125
Attributes .....	125
Functions.....	126
RoundRectangle .....	127
Attributes .....	127
Functions.....	128
SingleLineEdit.....	129
Attributes .....	129
Events .....	132
Functions.....	133
StaticText .....	135
Attributes .....	135
Events .....	138
Functions.....	139
User object .....	140
Attributes .....	140
Events .....	144
Functions.....	145
VScrollBar .....	147
Attributes .....	147
Events .....	149
Functions.....	149

<b>3</b>	<b>Attributes, Events, and Functions for Application Objects .....</b>	<b>151</b>
	Application object.....	152
	Attributes .....	152
	Events .....	153
	Functions.....	154
<b>4</b>	<b>Attributes, Events, and Functions for MenuItems .....</b>	<b>155</b>
	MenuItem .....	156
	Attributes .....	156
	Events .....	158
	Functions.....	158
<b>5</b>	<b>Attributes and Functions for MDI Client .....</b>	<b>159</b>
	MDI Client.....	160
	Attributes .....	160
	Functions.....	161

**6**

**Attributes, Events, and Functions for System**

**Objects ..... 163**

- Error ..... 164
  - Attributes ..... 164
  - Events ..... 165
  - Functions ..... 165
- Message ..... 166
  - Attributes ..... 166
  - Events ..... 167
  - Functions ..... 167
- OLEObject ..... 168
  - Events ..... 168
  - Functions ..... 168
- OLEStorage ..... 170
  - Events ..... 170
  - Functions ..... 170
- OLEStream ..... 172
  - Attributes ..... 172
  - Events ..... 172
  - Functions ..... 172
- Pipeline ..... 174
  - Attributes ..... 174
  - Events ..... 174
  - Functions ..... 175
- Transaction ..... 176
  - Attributes ..... 176
  - Events ..... 177
  - Functions ..... 177

**7**

**Attributes for Environment Object ..... 179**

- Environment ..... 180
  - Attributes ..... 180

# About This Manual

## **Subject**

This manual lists attributes, events, and functions for PowerBuilder objects and controls.

This manual is a reference to use when you are writing scripts. For complete information on the objects and controls you can use in PowerBuilder applications, see the *User's Guide*.

## **Audience**

This manual is for programmers using PowerBuilder to create and maintain applications.



## CHAPTER 1

# Attributes, Events, and Functions for Windows

### About this chapter

Windows are the main interface between the user and a PowerBuilder application. Windows can display information, request information from a user, and respond to the user's mouse or keyboard actions.

The definition of a window includes attributes, events, and functions. The attributes determine the style of the window—how it looks. The events are actions in the window; when an event is triggered, the associated script is executed. The functions can trigger events in the window, manipulate or change the window, or provide information about the window.

This chapter describes the attributes, events, and functions that are part of the definition of a PowerBuilder window.

### Contents

<b>Topic</b>	<b>Page</b>
Attributes	2
Events	8
Functions	11

# Window

## Attributes

Every window has a style that determines how it looks to the user. That style is governed by values assigned to the attributes of the window.

Attribute	Data type	Description
BackColor	Long	Specifies the numerical value of the background color: -2 to 16,777,215.  <i>ℳ</i> For more information about color, see the <i>Function Reference</i> .
Border	Boolean	Specifies whether the window has a border: TRUE FALSE
BringToTop	Boolean	Specifies whether PowerBuilder moves the window to the top of the front-to-back order: TRUE FALSE
ColumnsPerPage	Integer	Specifies the number of columns on a page. The default is 0 (10 columns per page).  PowerBuilder multiplies UnitsPerColumn by ColumnsPerPage to determine the number of PowerBuilder units to scroll the window horizontally when the user clicks in the scroll bar.
Control[ ]	WindowObject	Contains the controls in the window. You should not change the contents of this array in a script.

Attribute	Data type	Description
ControlMenu	Boolean	Specifies whether the Control Menu box displays in the title bar: TRUE FALSE
Enabled	Boolean	Specifies whether the window is enabled (can send and receive messages): TRUE FALSE
Height	Integer	Specifies the height of the window, in PowerBuilder units.
HScrollBar	Boolean	Specifies whether a horizontal scroll bar displays: TRUE FALSE
Icon	String	A stock icon or the name of the .ICO file that contains the icon that will be displayed when the window is minimized.
LinesPerPage	Integer	Specifies the number of lines on a page. The default is 0 (10 lines per page).  PowerBuilder multiplies UnitsPerLine by LinesPerPage to determine the number of PowerBuilder units to scroll the window vertically when the user clicks in the scroll bar.
MaxBox	Boolean	Specifies whether a Maximize Box displays in the title bar: TRUE FALSE

Attribute	Data type	Description
MenuID	Menu	<p>Specifies the ID of a menu.</p> <p><b>Note</b> PowerBuilder uses MenuID internally. To change the menu for a window from a script, use the ChangeMenu function; and to display a Popup menu, use the PopMenu function. In both functions, enter the fully qualified name to identify the menu or MenuItem.</p>
MenuName	String	<p>Specifies the name of a menu.</p> <p><b>Note</b> PowerBuilder uses MenuName internally. To change the menu for a window from a script, use the ChangeMenu function; to display a popup menu, use the PopMenu function. In both functions, enter the fully qualified name to identify the menu or MenuItem.</p>
MinBox	Boolean	<p>Specifies whether a Minimize Box displays in the title bar:</p> <p>TRUE FALSE</p>
Pointer	String	<p>Specifies the name of the file containing the pointer that is used for the window.</p>
Resizable	Boolean	<p>Specifies whether the window is resizable:</p> <p>TRUE FALSE</p>
Tag	String	<p>Specifies the tag value assigned to the window.</p>
Title	String	<p>Specifies the text of the window title.</p>



Attribute	Data type	Description
TitleBar	Boolean	Specifies whether a title bar displays: TRUE FALSE  The user can move a window only if it has a title bar.
ToolBarAlignment	ToolBarAlignment (enumerated)	In an MDI frame window, specifies where the toolbar displays.
ToolBarHeight	Integer	In an MDI frame window, specifies the height of the toolbar when it is a floating toolbar.
ToolBarItemsPerRow	Integer	In an MDI frame window, specifies the maximum number of items that display in each row of the toolbar when it is a floating toolbar.
ToolBarTitle	String	In an MDI frame window, specifies the title that displays at the top of the toolbar when it is a floating toolbar.
ToolBarVisible	Boolean	In an MDI frame window, specifies whether the toolbar displays: TRUE FALSE
ToolBarWidth	Integer	In an MDI frame window, specifies the width of the toolbar when it is a floating toolbar.
ToolBarX	Integer	In an MDI frame window, specifies the X coordinate (distance from the left edge of the window, in PowerBuilder units) of the toolbar when it is a floating toolbar.

Attribute	Data type	Description
ToolbarY	Integer	In an MDI frame window, specifies the Y coordinate (distance from the top of the window, in PowerBuilder units) of the toolbar when it is a floating toolbar.
UnitsPerColumn	Integer	<p>Specifies the number of PowerBuilder units you want to scroll right or left when the user clicks the left or right arrow in the horizontal scroll bar in a window or user object. The default is 0 (1/100 of the width of the window). When UnitsPerColumn is 0, PowerBuilder controls horizontal scrolling automatically.</p> <p>PowerBuilder multiplies UnitsPerColumn by ColumnsPerPage to determine the number of PowerBuilder units to scroll the window horizontally when the user clicks in the scroll bar.</p>
UnitsPerLine	Integer	<p>Specifies the number of PowerBuilder units you want to scroll up or down when the user clicks the up or down arrow in the vertical scroll bar in a window or user object. The default is 0 (1/100 of the window height). When UnitsPerLine is 0, PowerBuilder controls vertical scrolling automatically.</p> <p>PowerBuilder multiplies UnitsPerPage by UnitsPerLine to determine the number of PowerBuilder units to scroll the window vertically when the user clicks in the scroll bar.</p>

Attribute	Data type	Description
Visible	Boolean	Specifies whether the window is visible: TRUE FALSE
VScrollBar	Boolean	Specifies whether a vertical scroll bar displays: TRUE FALSE
Width	Integer	Specifies the width of the window, in PowerBuilder units.
WindowState	WindowState (enumerated)	Specifies the state in which you want to run a window.
WindowType	WindowType (enumerated)	Specifies the type of window.
X	Integer	<p>Specifies the X position (distance from left edge of screen) of the window, in PowerBuilder units.</p> <p>The values of the X coordinates in all windows except child windows are measured from the left side of the screen. In child windows, they are measured from the left side of the workspace of the parent window.</p> <p>The workspace is the area between the sides of the window (not including the thickness of the frame, toolbar, or scroll bar, if any) and the top and bottom of the window (not including the thickness of the border or the title bar, menu bar, toolbar, or scroll bar, if any).</p>

Attribute	Data type	Description
Y	Integer	<p data-bbox="874 235 1163 346">Specifies the Y position (distance from the top of the screen) of the window, in PowerBuilder units.</p> <p data-bbox="874 380 1184 608">The values of the Y coordinates in all windows except child windows are measured from the top of the screen. In child windows, they are measured from the top of the workspace of the parent window.</p> <p data-bbox="874 642 1190 922">The workspace is the area between the sides of the window (not including the thickness of the frame, toolbar, or scroll bar, if any) and the top and bottom of the window (not including the thickness of the border or the title bar, menu bar, toolbar, or scroll bar, if any).</p>

## Events

Scripts for events in a window and the controls in the window determine how the window behaves. Scripts control the action that is initiated when an event occurs within the window.

Event	Occurs
Activate	Just before the window becomes active. When an Activate event occurs, the first object in the tab order for the window gets focus. If there are no visible objects in the window, the window gets focus.
Clicked	When the user clicks in an unoccupied area of the window (any area with no visible, enabled object).
Close	When the window is closed.

Event	Occurs
CloseQuery	<p>When you remove a window from display (close it). When you close a window, PowerBuilder triggers the CloseQuery event and then inspects the value of Message.ReturnValue. If the Message.ReturnValue is 1, the window cannot be closed.</p> <p><b>Note</b> Closing any window causes PowerBuilder to close all child and popup windows that it opened, and closing an MDI Frame window causes PowerBuilder to close all sheet windows within it. Any window thus being closed can set Message.ReturnValue to cancel the close operation.</p>
Deactivate	When the window becomes inactive.
DoubleClickd	When the user double-clicks in an unoccupied area of the window (any area with no visible, enabled object).
DragDrop	When a dragged control is dropped on the window.
DragEnter	When a dragged control enters the window.
DragLeave	When a dragged control leaves the window.
DragWithin	When a dragged control is within the window.
Hide	Just before the window is hidden.
HotLinkAlarm	After a Dynamic Data Exchange (DDE) server application has sent new (changed) data and the client DDE application has received it.
Key	When the user presses a key and the insertion point is not in a line edit.
MouseDown	When the user presses the left mouse button in an unoccupied area of the window (any area with no visible, enabled object).
MouseMove	When the pointer is moved within the window.
MouseUp	When the user releases the left mouse button in an unoccupied area of the window (any area with no visible, enabled object).
Open	When a script executes the Open function for a window. The event occurs after the window has been opened but before it is displayed.

Event	Occurs
Other	When a Windows message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed in an unoccupied area of the window (any area with no visible, enabled object).
RemoteExec	When a DDE client application has sent a command.
RemoteHotLinkStart	When a DDE client application wants to start a hotlink.
RemoteHotLinkStop	When a DDE client application wants to end a hot link.
RemoteRequest	When a DDE client application requests data.
RemoteSend	When a DDE client application has sent data.
Resize	When the user or a script opens or resizes a window.
Show	When a script executes the Show function for this window. The event occurs just before the window is displayed.
SystemKey	When the insertion point is not in a line edit and the user presses the ALT key or ALT plus another key.
Timer	When a specified number of seconds elapses after the Timer function has been called.
ToolbarMoved	In an MDI frame window, when the user moves the FrameBar or SheetBar.

## Functions

The following functions can trigger events in a window, manipulate or change a window, or provide information about a window. You can also use the PowerShell system functions in scripts for a window. For a list of these functions, see the Object browser.

### Note

Use the Open function to open a window and the Close function to close a window. These functions are system functions and therefore are not listed here.

Function	Data type returned	Description
ArrangeSheets	Integer	Arranges the sheets or icons in the specified MDI frame window.
ChangeMenu	Integer	Changes the menu associated with a window.
ClassName	String	Returns the name assigned to the window.
CloseUserObject	Integer	Removes user object from view, closes it, and executes its Destructor event.
GetActiveSheet	Window	Returns the currently active sheet in the specified MDI frame window.
GetFirstSheet	Window	Obtains the top sheet in the MDI frame.
GetNextSheet	Window	Obtains the sheet that is behind the specified sheet in the MDI frame.
Hide	Integer	Makes the window invisible.
Move	Integer	Places the window in a new location specified by the X and Y arguments.
OpenUserObject	Integer	Displays user object, making its attributes available to scripts.
OpenUserObjectWithParm	Integer	Displays user object, making its attributes available to scripts, and stores parameter in Message object.

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
ParentWindow	Window	Returns the parent window of the window.
PointerX	Integer	Returns the distance from the left edge of the screen to the pointer.
PointerY	Integer	Returns the distance from the top of the screen to the pointer.
PostEvent	Boolean	Adds an event to the end of the message queue for the window.
Print	Integer	Prints the window.
Resize	Integer	Changes the size of the window to the size specified in the width and height arguments.
SetFocus	Integer	Sets the focus to the specified window.
SetMicroHelp	Integer	Sets the MicroHelp text in the specified MDI frame window.
SetPosition	Integer	Specifies the position of the window in the front-to-back order of the application.
SetRedraw	Integer	Turns on or off automatic redrawing of the window after every change.
Show	Integer	Makes the window visible.
TriggerEvent	Integer	Sends an event to a window control and executes the script associated with the event.
TypeOf	Object	Returns the type of the window.



Function	Data type returned	Description
WorkspaceHeight	Integer	Returns the height of the workspace of the specified window.  <b>Note</b> The workspace is the area between the sides of the window (not including the thickness of the frame, toolbar, or scroll bar, if any) and the top and bottom of the window (not including the thickness of the border or the title bar, menu bar, toolbar, or scroll bar, if any).
WorkspaceWidth	Integer	Returns the width of the workspace of the specified window.
WorkspaceX	Integer	Returns the distance from the left edge of the screen to the left edge of the workspace of the specified window.
WorkspaceY	Integer	Returns the distance from the top of the screen to the top of the workspace of the specified window.



## CHAPTER 2

# Attributes, Events, and Functions for Controls

### About this chapter

This chapter provides a brief description of controls and then lists each control and its attributes, events, and functions.

### Contents

<b>Topic</b>	<b>Page</b>
About controls	17
CheckBox	20
CommandButton	25
DataWindow control	30
DropDownListBox	49
EditMask	57
Graph	66
GroupBox	74
HScrollBar	78
Line	82
ListBox	84
MultiLineEdit	93
OLE 2.0	101
Oval	107
Picture	109
PictureButton	114
RadioButton	120
Rectangle	125
RoundRectangle	127

<b>Topic</b>	<b>Page</b>
SingleLineEdit	129
StaticText	135
UserObject	140
VScrollBar	147

## About controls

Controls are a type of object that can be placed in a window or user object. Each control has a definition that usually includes attributes, events, and functions.

## Types of controls

There are two types of controls:

- ◆ Controls that have events
- ◆ Controls that do not have events (these include the GroupBox and the **drawing objects**: Line, Oval, Rectangle, and RoundedRectangle)

## Controls and their uses

Use this control	To
CommandButton PictureButton	Invoke an action
DropDownListBox DataWindow control EditMask Graph ListBox MultiLineEdit Picture SingleLineEdit StaticText	Display data
CheckBox RadioButton	Indicate a choice
OLE 2.0	Embed an OLE 2.0 object
HScrollBar VScrollBar	Display relative data
GroupBox	Group controls

Use this control	To
Line Oval Rectangle RoundRectangle	Add interest or decoration (these are the <b>drawing objects</b> )

**User object controls**

If you created user objects in your application, you can place user object controls in your windows and other user objects.

## Default control names

When you place a control in a window, PowerBuilder assigns it a unique name. The name is the concatenation of the default prefix for the control and the lowest 1- to-4-digit number that makes the name unique. The default prefixes are:

Control	Default prefix
CheckBox	cbx_
CommandButton	cb_
DataWindow	dw_
DropDownListBox	ddl_
EditMask	em_
Graph	gr_
GroupBox	gb_
HScrollBar	hsb_
Line	ln_
ListBox	lb_
MultiLineEdit	mle_
OLE 2 0	ole_
Oval	oval_
Picture	p_
PictureButton	pb_

Control	Default prefix
RadioButton	rb_
Rectangle	r_
RoundRectangle	rr_
SingleLineEdit	sle_
StaticText	st_
User Object	uo_
VScrollBar	vsb_

## Naming conventions

For your applications, you should develop a naming convention that uses suffixes that have meaning in the application. For example:

You could change this name	To this
cb_6	cb_Retrieve
cbx_1	cbx_Retired
dw_1	dw_EmployeeData
sle_2	sle_LName

### Tip

You can change the prefixes (you do so in the Preferences painter). But if you do, Technical Support may find it difficult to recognize the objects in your scripts and help you if you have a problem.

# CheckBox

Checkboxes are small square boxes used to set independent options. When they are selected, they contain an X; when they are not selected, they are empty.

Since CheckBoxes are independent of each other, you can group them without affecting their behavior. Grouping CheckBoxes makes the window easier for the user to understand and use.


Typically, CheckBoxes have two states: on and off. But sometimes you will want to represent a third state, unknown or unspecified. The third state displays as a gray box.

## Attributes

Attribute	Data type	Description
Automatic	Boolean	Specifies whether the control displays an X when the user clicks it: TRUE FALSE
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215.  <i>ℳ</i> For more information about color, see the <i>Function Reference</i> .
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control.
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order: TRUE FALSE
Checked	Boolean	Specifies whether the control is selected: TRUE FALSE



Attribute	Data type	Description
DragAuto	Boolean	Specifies whether PowerBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values: TRUE - When the control is clicked, the control is automatically in Drag Mode. FALSE - When the control is clicked, the control is not automatically in Drag Mode. You have manually put the control into Drag Mode by using the Drag function.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.  When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be clicked): TRUE FALSE
FaceName	String	Specifies the name of the typeface in which the text of the control displays; for example, ARIAL or COURIER.
FontCharSet	FontCharSet (enumerated)	Specifies the font character set used for the text in the control.
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control.
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control.
Height	Integer	Specifies the height of the control, in PowerBuilder units.

Attribute	Data type	Description
Italic	Boolean	Specifies whether the text in the control is italic: TRUE FALSE
LeftText	Boolean	Specifies whether the text displays on the left of the control: TRUE FALSE  This attribute is usually FALSE, and the text appears on the right of the control.
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
TabOrder	Integer	Specifies tab value of the control within the window (0 means the user cannot tab to the control).
Tag	String	Specifies the tag value assigned to the control.
Text	String	Specifies the text that displays next to the control.
TextColor	Long	Specifies the numeric value of the color used for text: -2 to 16,777,215.   For more information about color, see the <i>Function Reference</i> .
TextSize	Integer	Specifies the size of the text in the control, in points. For backward compatibility, the size is stored as a negative number; for example, 10-point text size is stored as -10.
ThirdState	Boolean	Specifies whether the control is in the third state (active and inactive): TRUE FALSE
ThreeState	Boolean	Specifies whether the control has three states: TRUE - Three states FALSE - Two states

Attribute	Data type	Description
Underline	Boolean	Specifies whether the text in the control is underlined: TRUE FALSE
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE
Weight	Integer	Specifies the stroke weight of the text in the control; for example, 400 for normal or 700 for bold.
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.

## Events

Event	Occurs
Clicked	When the control is clicked (selected or unselected).
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When a dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PowerBuilder event.

Event	Occurs
RButtonDown	When the right mouse button is pressed on the control.

## Functions

Function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
Hide	Integer	Makes the control invisible.
Move	Integer	Moves the control to a specified location.
PointerX	Integer	Returns the distance the pointer is from the left edge of the control.
PointerY	Integer	Returns the distance the pointer is from the top of the control.
PostEvent	Boolean	Adds an event to the end of the message queue for the control.
Print	Integer	Prints the control.
Resize	Integer	Changes the size of the control.
SetFocus	Integer	Sets focus to the control.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its attributes.
Show	Integer	Makes the control visible.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

# CommandButton

You use a `CommandButton` to carry out an action. For example, you can use an OK button to confirm a deletion or a Cancel button to cancel the requested deletion.

## Attributes

Attribute	Data type	Description
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order of the window: TRUE FALSE
Cancel	Boolean	Specifies whether the control acts as the Cancel button (the Cancel button receives a Clicked event if the user presses ESC): TRUE FALSE
Default	Boolean	Specifies whether the control is the default control (the default control has a thick border and receives a clicked event if the user presses ENTER without selecting a control): TRUE FALSE
DragAuto	Boolean	Specifies whether PowerBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values: TRUE - When the control is clicked, the control is automatically in Drag Mode. FALSE - When the control is not clicked, the control is not automatically in Drag Mode. You have to manually put the control into Drag Mode by using the Drag function.

Attribute	Data type	Description
DragIcon	String	<p>Contains the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.</p> <p>When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.</p>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be clicked):</p> <p>TRUE FALSE</p>
FaceName	String	<p>Specifies the name of the typeface in which the text of the control displays (for example, ARIAL or COURIER).</p>
FontCharSet	FontCharSet (enumerated)	<p>Specifies the font character set used for the text in the control.</p>
FontFamily	FontFamily (enumerated)	<p>Specifies the font family (type style) used for the text in the control.</p>
FontPitch	FontPitch (enumerated)	<p>Specifies the pitch (spacing) of the font used for the text in the control.</p>
Height	Integer	<p>Specifies the height of the control, in PowerBuilder units.</p>
Italic	Boolean	<p>Specifies whether the text in the control is italic:</p> <p>TRUE FALSE</p>
Pointer	String	<p>Specifies the name of the stock pointer of the file containing the pointer that is used for the control.</p>
TabOrder	Integer	<p>Specifies the tab value of the control within the window (0 means the user cannot tab to the control).</p>
Tag	String	<p>Specifies the tag value assigned to the control.</p>

Attribute	Data type	Description
Text	String	Specifies the text that displays in the control.
TextSize	Integer	Specifies the size of the text in the control, in points. For backward compatibility, the size is stored as a negative number; for example, 10-point text size is stored as -10.
Underline	Boolean	Specifies whether the text in the control is underlined: TRUE FALSE
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE
Weight	Integer	Specifies the stroke weight of the text in the control; for example, 400 for normal or 700 for bold.
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.

## Events

<b>Event</b>	<b>Occurs</b>
Clicked	When the control is clicked.
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When a dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Before the control receives focus (before it is selected and becomes active).
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed on the control.



## Functions

Function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
Hide	Integer	Makes the control invisible.
Move	Integer	Moves the control to a specified location.
PointerX	Integer	Returns the distance the pointer is from the left edge of the control.
PointerY	Integer	Returns the distance the pointer is from the top of the control.
PostEvent	Boolean	Adds an event to the end of the message queue for the control.
Print	Integer	Prints the control.
Resize	Integer	Changes the size of the control.
SetFocus	Integer	Sets focus to the control.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its attributes.
Show	Integer	Makes the control visible.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the control type of the control.

## DataWindow control

You place DataWindow controls in a window or user object and then specify the DataWindow object you want to use within them to display and manipulate data in the window.

### DataWindow objects

A DataWindow object allows users to display, manipulate, and update database or other information. You build DataWindow objects in the DataWindow painter.

*ℳ* For information about DataWindow objects, see the *User's Guide*.

## Attributes

Attribute	Data type	Description
Border	Boolean	Specifies whether the control has a border: TRUE FALSE
BorderStyle	BorderStyle (enumerated)	Specifies the border style of the control.
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order.
ControlMenu	Boolean	Specifies whether the Control Menu box displays in the control title bar: TRUE FALSE
DataObject	String	Specifies the name of the DataWindow or Report object associated with the control.

Attribute	Data type	Description
DragAuto	Boolean	Specifies whether PowerBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values: TRUE - When the control is clicked, the control is automatically in Drag Mode. FALSE - When the control is clicked, the control is not automatically in Drag Mode. You have to manually put the control into Drag Mode by using the Drag function.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.  When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected): TRUE FALSE
Height	Integer	Specifies the height of the DataWindow control, in PowerBuilder units.
HScrollBar	Boolean	Specifies whether a horizontal scroll bar displays in the control when all the data cannot be displayed at one time: TRUE FALSE
HSplitScroll	Boolean	Specifies whether the split bar displays in the control: TRUE FALSE
Icon	String	Specifies the name of the ICO file that contains the icon that displays when the DataWindow control is minimized.

<b>Attribute</b>	<b>Data type</b>	<b>Description</b>
LiveScroll	Boolean	Scrolls the rows in the DataWindow object while the user is moving the scrollbar.
MaxBox	Boolean	Specifies whether a Maximize Box displays in the DataWindow title bar: TRUE FALSE
MinBox	Boolean	Specifies whether a Minimize Box displays in the DataWindow title bar: TRUE FALSE
Resizable	Boolean	Specifies whether the DataWindow control is resizable: TRUE FALSE
TabOrder	Integer	Specifies the tab value of the DataWindow control within the window or user object(0 means the user cannot tab to the control).
Tag	String	Specifies the tag value assigned to the DataWindow control.
Title	String	Specifies the text that displays in the DataWindow title bar.
TitleBar	Boolean	Specifies whether a title bar displays in the DataWindow control: TRUE FALSE  The user can move the DataWindow control only if it has a title bar.
Visible	Boolean	Specifies whether the DataWindow control is visible: TRUE FALSE
VScrollBar	Boolean	Specifies whether a vertical scroll bar displays in the control when not all the data can be displayed at one time: TRUE FALSE
Width	Integer	Specifies the width of the DataWindow control, in PowerBuilder units.

Attribute	Data type	Description
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top edge of the window), in PowerBuilder units.

## Events

Some but not all DataWindow events have action codes that you can use to determine what action will be taken after the event occurs. You set the action codes in the SetActionCode function.

*ℳ* For information about the SetActionCode function, see the *Function Reference*.

Event	Occurs
Clicked	When the user clicks a noneditable field or between fields in the DataWindow.  Action codes: 0 - (Default) Continue processing. 1 - Stop processing.
Constructor	Immediately before the Open event occurs in the window.
DBError	When a database error occurs in the DataWindow.  Action codes: 0 - (Default) Display the error message. 1 - Do not display the error message.
Destructor	Immediately after the Close event occurs in the window.
DoubleClicked	When the user double-clicks a noneditable field or between fields in the DataWindow.
DragDrop	When a dragged control is dropped on the DataWindow control.
DragEnter	When a dragged control enters the DataWindow control.

Event	Occurs
DragLeave	When a dragged control leaves the DataWindow control.
DragWithin	When a dragged control is within the DataWindow control.
EditChanged	When a user types in an edit control in the DataWindow.
GetFocus	Just before the DataWindow control receives focus (before it is selected and becomes active).
ItemChanged	When a field in the DataWindow has been modified and loses focus (for example, the user presses ENTER, the TAB key, or an arrow key or clicks the mouse on another field).  Action codes: 0 - (Default) Accept the data value. 1 - Reject the data value and don't allow focus to change. 2 - Reject the data value but allow the focus to change.
ItemError	When a field has been modified, the field loses focus (for example, the user presses ENTER, TAB, or an arrow key or clicks the mouse on another field), and the field does not pass the validation rules for its column.  Action codes: 0 - (Default) Reject the data value and show an error message box. 1 - Reject the data value with no message box. 2 - Accept the data value. 3 - Reject the data value but allow focus to change.  If the Action Code is 0 or 1 (rejects the data), the field with the incorrect data regains the focus.
ItemFocusChanged	When the current item in the control changes.
LoseFocus	When the DataWindow control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PowerBuilder event.
PrintEnd	When the printing of the DataWindow ends.

Event	Occurs
PrintPage	<p>Before each page of the DataWindow is formatted for printing.</p> <p>Action codes:            0 - Do not skip a page.            1 - Skip a page.</p>
PrintStart	<p>When the printing of the DataWindow starts.</p>
RButtonDown	<p>When the right mouse button is pressed on the control.</p>
Resize	<p>When the user or a script resizes a DataWindow control.</p>
RetrieveEnd	<p>When the retrieval for the DataWindow is complete.</p>
RetrieveRow	<p>After a row has been retrieved.</p> <p>Action codes:            0 - (Default) Continue.            1 - Stop the retrieval.</p>
RetrieveStart	<p>When the retrieval for the DataWindow is about to begin.</p> <p>Action codes:            0 - (Default) Continue.            1 - Do not perform the retrieval.            2 - Do not reset the rows and buffers before retrieving the data from the database.</p>
RowFocusChanged	<p>When the current row changes in the DataWindow.</p>
ScrollHorizontal	<p>When the user scrolls right or left in the DataWindow control with the TAB or arrow keys or the scroll bar.</p>
ScrollVertical	<p>When the user scrolls up or down in the DataWindow control with the TAB or arrow keys or the scroll bar.</p>
SQLPreview	<p>After a Retrieve, Update, or ReselectRow function call and immediately before the SQL statement is submitted to the DBMS.</p>
UpdateEnd	<p>When all the updates from the DataWindow to the database are complete.</p>
UpdateStart	<p>After an Update function call and just before changes in the DataWindow are sent to the database.</p> <p>Action codes:            0 - (Default) Continue.            1 - Do not perform the update.</p>

## Functions

Function	Data type returned	Description
AcceptText	Integer	Applies the contents of the current field to the current item in the DataWindow.
CanUndo	Boolean	Specifies whether the last edit can be undone with the Undo function: TRUE FALSE
CategoryCount	Integer	Returns the number of categories in the specified graph in the DataWindow.
CategoryName	String	Returns the name of the specified category in the specified graph in the DataWindow.
ClassName	String	Returns the name assigned to the DataWindow control.
Clear	Integer	Clears (deletes) the selected text in the current field of the DataWindow object.
ClearValues	Integer	Deletes all items from the value list associated with the specified column in the DataWindow.
Clipboard	Integer	Copies the specified graph in the DataWindow to the clipboard.
Copy	Integer	Copies the selected text in the current field of the DataWindow to the clipboard.
Create	Integer	Creates a DataWindow object using the specified source code and replaces the DataWindow object in the specified DataWindow control with the new DataWindow object.
CrosstabDialog	Integer	Displays the Crosstab Definition dialog box so the user can modify the definition of a crosstab DataWindow during execution.



Function	Data type returned	Description
Cut	Integer	Cuts the selected text from the current field of the DataWindow and stores it in the clipboard.
DataCount	Long	Returns the number of data points in the specified series in the specified graph in the DataWindow.
DBCcancel	Integer	Cancel a database retrieval in progress.
DBErrorCode	Long	Returns the error code (number) generated by a database error.
DBErrorMessage	String	Returns a string containing the text of the error message generated by a database error.
DeletedCount	Long	Returns the number of rows that have been deleted from the DataWindow but have not yet been updated in the associated database table.
DeleteRow	Integer	Deletes the specified row from the DataWindow.
Describe	String	Returns requested information about the structure of the DataWindow object.
Drag	Integer	Starts or ends the dragging of the DataWindow control.
Filter	Integer	Displays specific rows of the DataWindow based on its current filter.
FilteredCount	Integer	Returns the number of rows that are not visible because of the DataWindow's current filter.
Find	Long	Returns the number of the first row that meets the search criteria within a specified search range in the detail area of a DataWindow.

Function	Data type returned	Description
FindCategory	Integer	Returns the number of the specified category in the specified graph in the DataWindow.
FindGroupChange	Long	Searches starting at a specified row for the first break for the specified group in the DataWindow.
FindRequired	Integer	Identifies the required columns that the user has not filled.
FindSeries	Integer	Returns the number of the specified series in the specified graph in the DataWindow.
GetBandAtPointer	String	Returns the string containing the band in which the pointer is currently located followed by a tab character (~t) and the number of the row associated with the band.
GetBorderStyle	Border (enumerated)	Returns a Border enumerated data type indicating the border style of the specified column in the DataWindow.
GetChild	Integer	Stores in the specified variable the name of the child DataWindow in the specified column.
GetClickedColumn	Integer	Returns the number of the column in the DataWindow that the user clicked or double-clicked.
GetClickedRow	Long	Returns the number of the row in the DataWindow that the user clicked or double-clicked.
GetColumn	Integer	Returns the number of the current column in the DataWindow.
GetColumnName	String	Returns the name of the current column in the DataWindow.
GetData	Double	Returns the value of the specified data in the specified series in the specified graph in the DataWindow (superseded by GetDataValue, which is more flexible).

Function	Data type returned	Description
GetDataPieExplode	Integer	Reports the percentage that a pie slice is exploded in a pie graph.
GetDataStyle	Integer	Returns the pattern used to fill the specified data point (Format 1) or the color used for the specified data point (Format 2) in the specified series in the specified graph in the DataWindow.
GetDataValue	Integer	Returns the value of the specified data in the specified series in the specified graph in the DataWindow.
GetFormat	String	Returns the format used for display in the specified column of the DataWindow.
GetItemDate	Date	Returns the date data in the specified row and column of the DataWindow.
GetItemDateTime	DateTime	Returns the datetime data in the specified row and column of the DataWindow.
GetItemDecimal	Decimal	Returns the decimal data in the specified row and column of the DataWindow.
GetItemNumber	Double	Returns the numeric data in the specified row and column of the DataWindow.
GetItemStatus	dwItemStatus (enumerated)	Returns the status of the item at the specified row and column location in the specified buffer.
GetItemString	String	Returns the string data in the specified row and column of the DataWindow.
GetItemTime	Time	Returns the time data in the specified row and column of the DataWindow.
GetMessageText	String	Obtains the message text generated by a crosstab DataWindow

Function	Data type returned	Description
GetNextModified	Long	Returns the number of the first row that was modified in the specified buffer in the specified DataWindow after the specified row.
GetObjectAtPointer	String	Returns the string containing the name of the DataWindow column or graphic object under the pointer in the DataWindow, then a tab character (~t), and then the row number.
GetRow	Long	Returns an integer containing the number of the current row in the DataWindow.
GetSelectedRow	Integer	Returns the number of the first selected row after the specified row number in the DataWindow.
GetSeriesStyle	Integer	Returns the FillPattern enumerated data type representing the pattern used to fill the specified series (Format 1) or the specified color for the specified series (Format 2) for the specified graph in the DataWindow.
GetSQLPreview	String	Returns the current SQL statement for the DataWindow.
GetSQLSelect	String	Returns the current SELECT statement for the DataWindow.
GetText	String	Returns the text in the current field of the DataWindow.
GetTrans	Integer	Returns the values in the DataWindow transaction object.
GetUpdateStatus	Integer	Stores the number of the row that will be updated in a variable and the dwBuffer enumerated data type identifying the buffer containing the row in another variable.
GetValidate	String	Returns the validation rule used in the specified column of the DataWindow.


Function	Data type returned	Description
GetValue	String	Returns the specified item in the value list for the specified column.
GroupCalc	Integer	Recalculates the breaks in the groups in the DataWindow.
Hide	Integer	Makes the control invisible.
ImportClipboard	Long	Copies data from the clipboard to the DataWindow.
ImportFile	Long	Copies data from a file to the DataWindow.
ImportString	Long	Copies data from a string to the DataWindow.
InsertRow	Long	Inserts a new initialized row before the specified row in the DataWindow.
IsSelected	Boolean	Returns TRUE if the specified row in the DataWindow is selected; returns FALSE if the row is not selected or is greater than the number of rows in the DataWindow.
LineCount	Integer	Returns the number of lines in the current field in the DataWindow.
ModifiedCount	Long	Returns the number of rows that have been modified in the DataWindow but have not yet been updated in the associated database table.
Modify	String	Uses the specification contained in a string to modify the DataWindow object.
Move	Integer	Moves the specified DataWindow control to a specified location.
ObjectAtPointer	grObjectType	Returns the number of the series the pointer is over and the number of the data point in the graph in the DataWindow and identifies the object type.

Function	Data type returned	Description
OLEActivate	Integer	Activates OLE for the OLE object in the specified row and column of the DataWindow.
Paste	Integer	Inserts the contents of the clipboard at the cursor position in the current field of the DataWindow.
PointerX	Integer	Returns the distance the pointer is from the left edge of the DataWindow control.
PointerY	Integer	Returns the distance the pointer is from the top of the DataWindow control.
Position	Integer	Obtains the cursor position in the current field of the DataWindow.
PostEvent	Boolean	Adds an event to the end of the message queue for the DataWindow control.
Print	Integer	Sends the contents of the DataWindow to the current printer.
PrintCancel	Integer	Cancel the print job and deletes the spool file (if any) when the contents of the DataWindow is sent to print.  Use Format 1 when the PrintDataWindow function was used to send it to print. Use Format 2 when Format 1 of the Print function was used to send it to print.
ReplaceText	Integer	Replaces the selected text in the current field of the DataWindow with the specified string.
ReselectRow	Integer	Accesses the database to reselect all columns that can be updated and refreshes all timestamp columns in a row in the DataWindow.

Function	Data type returned	Description
Reset	Integer	Format 1 completely clears the contents of the DataWindow object without deleting the rows from the underlying table. Format 2 deletes all items from the value list of a column.
ResetDataColors	Integer	Resets the color of a data point to the color specified for the series.
ResetTransObject	Integer	Stops the DataWindow object from using a programmer-defined transaction object (thereafter, the DataWindow uses its internal transaction object).
ResetUpdate	Integer	Resets the update flags for the DataWindow.
Resize	Integer	Changes the width and height of the DataWindow control.
Retrieve	Long	Causes the DataWindow to retrieve rows from the database.
RowCount	Long	Returns the number of rows currently available in the DataWindow (all the rows retrieved minus any deleted rows plus any inserted rows minus any rows that have been filtered out).
RowsCopy	Integer	Copies a range of rows from one DataWindow control to another or from one buffer to another within a single DataWindow control.
RowsDiscard	Integer	Discards a range of rows. The rows cannot be restored without retrieving from the database.
RowsMove	Integer	Clears a range of rows from a DataWindow control and inserts the rows in another DataWindow control or another buffer of the same DataWindow control.

Function	Data type returned	Description
SaveAs	Integer	Saves the data represented in the specified graph in the DataWindow to the specified file name in the specified format.
SaveAs	Integer	Saves the contents of the DataWindow to a file in the specified format with or without column headings at the beginning.
Scroll	Integer	Scrolls the DataWindow the specified direction the specified number of lines.
ScrollNextPage	Long	Scrolls forward by the number of rows showing in the DataWindow.
ScrollNextRow	Long	Scrolls the DataWindow to the next row. ScrollNextRow changes the current row but does not change the current column.
ScrollPriorPage	Long	Scrolls backward by the number of rows showing in the DataWindow.
ScrollPriorRow	Long	Scrolls to the previous row. The ScrollPriorRow function changes the current row in the DataWindow but does not change the current column.
ScrollToRow	Integer	Causes the control to scroll to the specified row. ScrollToRow changes the current row in the DataWindow but does not change the current column.
SelectedLength	Integer	Determines the total number of characters and spaces (length) in the specified text in the current field of the DataWindow.
SelectedLine	Integer	Obtains the number of the line where the cursor is located in the DataWindow control.
SelectedStart	Integer	Returns the location of the first character in the selected text in the current field of the DataWindow.



Function	Data type returned	Description
SelectedText	String	Determines what text (if any) is selected in the current field of the DataWindow.
SelectRow	Integer	Selects or deselects the specified row of the DataWindow.
SelectText	Integer	Selects text in the current field of the DataWindow beginning at the specified position and continuing for the specified number of characters and highlights the text when the DataWindow control has focus.
SeriesCount	Integer	Returns the number of series in the specified graph in the DataWindow.
SeriesName	String	Returns the name of the specified series in the specified graph in the DataWindow.
SetActionCode	Integer	Defines the action a DataWindow takes following an event.   For information on action codes, see the DataWindow control "Events" section earlier in this chapter.
SetBorderStyle	Integer	Sets the border style of the specified column in the DataWindow.
SetColumn	Integer	Makes the specified column the current column in the DataWindow.
SetDataPieExplode	Integer	Explodes a pie slice in a pie graph.
SetDataStyle	Integer	Sets the fill pattern (Format 1) or the color (Format 2) for the specified datapoint in the specified series in the specified graph in the DataWindow.
SetDetailHeight	Integer	Sets the height of each row in a specified range.

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
SetFilter	Integer	Defines the filter criteria for the DataWindow. The actual filtering is performed by the Filter function.
SetFocus	Integer	Sets focus to the DataWindow control.
SetFormat	Integer	Sets the display format for the specified column of the DataWindow.
SetItem	Integer	Sets the value of the specified row and column of the specified DataWindow.
SetItemStatus	Integer	Sets the status of a row in a specified column of the DataWindow in the specified buffer.
SetPosition	Integer	Specifies the position of a DataWindow graphic object in the DataWindow.
SetPosition	Integer	Specifies whether the DataWindow control always displays on top in the front-to-back order within the window.
SetRedraw	Integer	Controls automatic redrawing of the DataWindow control after each change in its attributes or contents.
SetRow	Integer	Makes the specified row the current row in the DataWindow.
SetRowFocusIndicator	Integer	Sets the current row indicator for the DataWindow.
SetSeriesStyle	Integer	Sets the fill pattern (Format 1) or the color (Format 2) for the specified series in the specified graph in the DataWindow.
SetSort	Integer	Defines the sort criteria for the DataWindow. The actual sorting is performed by the Sort function.
SetSQLPreview	Integer	Sets the current SQL statement for the DataWindow.

Function	Data type returned	Description
SetSQLSelect	Integer	Changes the current SELECT statement for the DataWindow.
SetTabOrder	Integer	Changes the tab value of the specified column in the DataWindow.
SetText	Integer	Replaces the text in the field at the current row and column of the DataWindow with the specified text.
SetTrans	Integer	Sets values in the DataWindow's internal transaction object.
SetTransObject	Integer	Sets the transaction object for the DataWindow and provides control over the transaction, including the ability to commit from a script.
SetValidate	Integer	Changes the validation rule used for the specified column of the DataWindow.
SetValue	Integer	Sets the value of the specified item in the value list or the code table of the specified column of the DataWindow.
ShareData	Integer	Shares data between a primary DataWindow control and a secondary DataWindow control.
ShareDataOff	Integer	Turns off sharing for the DataWindow control. If that DataWindow control is the primary DataWindow control, all secondary DataWindow controls are disconnected and their DataWindow objects no longer contain data.
Show	Integer	Makes the DataWindow control visible.
Sort	Integer	Sorts the rows of the DataWindow based on its current sort criteria.

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
TextLine	String	Obtains the text on the cursor line in the current field of the DataWindow.
TriggerEvent	Integer	Triggers a specified event in the DataWindow control and executes the script for the event.
TypeOf	Object	Returns the type of the control.
Undo	Integer	Cancels the last edit in the current field of the DataWindow.
Update	Integer	Sends to the database all inserts, deletes, and updates of the DataWindow.


## DropDownListBox

A DropDownListBox combines the features of a ListBox and a SingleLineEdit. In some DropDownListBoxes, the user can select an item by entering the name of the item in the edit box. In other DropDownListBoxes, the user cannot modify the edit box and must click the item or enter the first character of the item to select it.

### Making the list display


In the development environment, if the list portion of the DropDownListBox is not displayed, the user must click the down arrow at the end of the edit box to display it.

## Attributes

Attribute	Data type	Description
Accelerator	Integer	The ASCII value of the accelerator key you want to assign as the accelerator for the control.
AllowEdit	Boolean	Specifies whether the user can enter text in the edit box portion of the control: TRUE FALSE
AutoHScroll	Boolean	Specifies whether the edit box portion of the control scrolls horizontally automatically when data is entered or deleted: TRUE FALSE
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215.  For more information about color, see the <i>Function Reference</i> .
Border	Boolean	Specifies whether the control has a border: TRUE FALSE

Attribute	Data type	Description
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control.
BringToTop	Boolean	Specifies whether PowerBuilder will move the control to the top of the front-to-back order in the window: TRUE FALSE
DragAuto	Boolean	Specifies whether PowerBuilder will put the control automatically into Drag Mode. DragAuto has these boolean values: TRUE - When the control is clicked, the control is automatically in Drag Mode. FALSE - When the control is not clicked, the control is not automatically in Drag Mode. You have to manually put the control into Drag Mode by using the Drag function.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.  When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected): TRUE FALSE
FaceName	String	Specifies the name of the typeface in which the text of the control displays (for example, ARIAL or COURIER).
FontCharSet	FontCharSet (enumerated)	Specifies the font character set used for the text in the control.
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control.

Attribute	Data type	Description
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control.
Height	Integer	Specifies the height of the control, in PowerBuilder units.
HScrollBar	Boolean	Specifies whether a horizontal scroll bar is displayed in the control: TRUE FALSE
Italic	Boolean	Specifies whether the text in the control is italic: TRUE FALSE
Item[ ]	String array	Specifies the contents of the ListBox portion of the DropDownListBox.
Limit	Integer	Specifies the maximum number of characters (0 to 32,767) the user can enter in the SingleLineEdit portion of the DropDownListBox (0 means unlimited).
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer that will be used for the control.
ShowList	Boolean	Specifies whether the option list always displays in the ListBox portion of the DropDownListBox when the control displays: TRUE FALSE  This attribute is usually set to FALSE (displays only when the user clicks the down arrow).
Sorted	Boolean	Specifies whether the ListBox portion of the DropDownListBox is automatically sorted in ascending order: TRUE FALSE
TabOrder	Integer	Specifies the tab value of the control within the window (0 means the user cannot tab to the control).
Tag	String	Specifies the tag value assigned to the control.

Attribute	Data type	Description
Text	String	Specifies the text in the control.
TextColor	Long	Specifies the numeric value of the color used for text: -2 to 16,777,215.   For more information about color, see the <i>Function Reference</i> .
TextSize	Integer	Specifies the size of the text in the control, in points. For backward compatibility, the size is stored as a negative number; for example, 10-point text size is stored as -10.
Underline	Boolean	Specifies whether the text in the control is underlined: TRUE FALSE
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE
VScrollBar	Boolean	Specifies whether a vertical scroll bar is displayed in the control: TRUE FALSE
Weight	Integer	Specifies the stroke weight of the text in the control; for example, 400 for normal or 700 for bold.
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.



## Events

Event	Occurs
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DoubleClicked	When the control is double-clicked (selected and activated).
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When a dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
LoseFocus	When the control loses focus (becomes inactive).
Modified	When the control loses focus, the text has been changed, and the ENTER or TAB key is pressed.
Other	When a Windows message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed on the control.
SelectionChanged	When an item is selected in the ListBox portion of the DropDownListBox.

\*

## Functions

Function	Data type returned	Description
AddItem	Integer	Adds a new item to the end of the ListBox portion of the control.
ClassName	String	Returns the name assigned to the control.
Clear	Integer	Clears the selected text from the control (but does not place it in the clipboard).
Copy	Integer	Copies (but does not delete) the selected text from the control to the clipboard.
Cut	Integer	Cuts (deletes) the selected text (if any) from the control and places it in the clipboard.
DeleteItem	Integer	Deletes the item indicated by the index from the ListBox portion of the control.
DirList	Boolean	Populates the ListBox portion of the DropDownListBox with a list of the files of the specified type that match the specified file pattern.
DirSelect	Boolean	Retrieves the current selection from the specified control and puts it in the specified variable.
Drag	Integer	Starts or ends the dragging of the control.
FindItem	Integer	Finds the first item in the ListBox portion of the control (after the specified index) that begins with a specified string.
Hide	Integer	Makes the control invisible (hidden).


Function	Data type returned	Description
InsertItem	Integer	Adds a new item to the ListBox portion of the DropDownListBox before the item indicated by the index.
Move	Integer	Moves the control to a specified location.
Paste	Integer	Inserts the contents of the clipboard (if any) at the cursor location in the control.
PointerX	Integer	Returns the distance the pointer is from the left edge of the control.
PointerY	Integer	Returns the distance the pointer is from the top of the control.
Position	Integer	Returns the position of the cursor in the control.
PostEvent	Boolean	Adds an event to the end of the message queue for control.
Print	Integer	Prints the control.
ReplaceText	Integer	Replaces the selected text in the control with the specified string.
Reset	Integer	Deletes all items from the control.
Resize	Integer	Changes the size of the control.
SelectedLength	Integer	Returns the length of the selected text in the control.
SelectedStart	Integer	Returns the starting position of the selected text (if any) in the control.
SelectedText	String	Returns a string containing the selected text (if any) from the control (the Allow attribute must be TRUE).

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
SelectItem	Integer	Format 1 finds and selects the first item in the control (after the specified index) that begins with the specified string; Format 2 finds and selects the specified item number.
SelectText	Integer	Selects the text in the control specified by the starting position and length; when the control has focus, highlights the text.
SetFocus	Integer	Sets focus in the first item in the box.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its attributes.
Show	Integer	Makes the control visible
Text	String	Returns the text of the item in the ListBox portion of the DropDownListBox that is identified by the index.
TotalItems	Integer	Returns the total number of items in the ListBox portion of the DropDownListBox.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

## EditMask

An EditMask is a box similar to a MultiLineEdit in which the user can enter and edit one or more lines of text. The text is formatted using the edit mask specified for the control. For example, you might use an EditMask to format a telephone number or date automatically as the user enters it.

## Attributes

Attribute	Data type	Description
Accelerator	Integer	Specifies the ASCII value of the key you want to assign as the accelerator key for the control.
Alignment	Alignment	Specifies the alignment of text in the control.
AutoHScroll	Boolean	Specifies whether PowerBuilder automatically scrolls left or right when data is entered into or deleted from the control: TRUE FALSE
AutoSkip	Boolean	Specifies whether to skip to the next control when the last character in the edit mask has been entered: TRUE FALSE
AutoVScroll	Boolean	Specifies whether PowerBuilder automatically scrolls up or down when data is entered into or deleted from the control: TRUE FALSE
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215.   For more information about color, see the <i>Function Reference</i> .

Attribute	Data type	Description
Border	Boolean	Specifies whether the control has a border: TRUE FALSE
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control.
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order in the window: TRUE FALSE
DisplayData	String	Specifies the data that initially displays in the control.
DisplayOnly	Boolean	Specifies whether the text in the control is display-only and cannot be changed by the user: TRUE FALSE
DragAuto	Boolean	Specifies whether PowerBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values: TRUE - When the control is clicked, the control is automatically in Drag Mode. FALSE - When the control is clicked, the control is not automatically in Drag Mode. You have to manually put the control into Drag Mode by using the Drag function.

Attribute	Data type	Description
DragIcon	String	<p>Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.</p> <p>When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.</p>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected):</p> <p>TRUE FALSE</p>
FaceName	String	<p>Specifies the name of the typeface in which the text of the control displays (for example, ARIAL or COURIER).</p>
FontCharSet	FontCharSet (enumerated)	<p>Specifies the font character set used for the text in the control.</p>
FontFamily	FontFamily (enumerated)	<p>Specifies the font family (type style) used for the text in the control.</p>
FontPitch	FontPitch (enumerated)	<p>Specifies the pitch (spacing) of the font used for the text in the control.</p>
Height	Integer	<p>Specifies the height of the control, in PowerBuilder units.</p>
HScrollBar	Boolean	<p>Specifies whether a horizontal scroll bar displays in the control when all the data cannot be displayed at one time:</p> <p>TRUE FALSE</p>
HideSelection	Boolean	<p>Specifies whether selected text stays selected (highlighted) even when the control does not have focus:</p> <p>TRUE (text does not stay highlighted) FALSE (text stays highlighted)</p>
Increment	Double	<p>Specifies the increment of the spin.</p>

Attribute	Data type	Description
Italic	Boolean	Specifies whether the text in the control is italic: TRUE FALSE
Limit	Integer	Specifies the maximum number of characters (0 to 32,767) that can be entered in the control (0 means unlimited).
Mask	String	Specifies the mask to use to format and edit data in the control.
MaskDataType	MaskDataType (enumerated)	Specifies the data type of the control.
MinMax	String	Specifies the minimum and maximum values for the spin.
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
Spin	Boolean	Specifies whether to scroll through the spin values: TRUE FALSE
TabOrder	Integer	Specifies the tab value of the control within the window (0 means the user cannot tab to the control).
TabStop[ ]	Integer	Specifies the positions of tab stops in the control.
Tag	String	Specifies the tag value assigned to the control.
Text	String	Specifies the text that displays in the control.
TextColor	Long	Specifies the color to be used for the text in the control.  <i>ℳ</i> For more information about color, see the Function Reference.
TextCase	TextCase (enumerated)	Specifies the case used to display text the user enters



Attribute	Data type	Description
TextSize	Integer	Specifies the size of the text in the control, in points. For backward compatibility, the size is stored as a negative number; for example, 10-point text size is stored as -10.
UnderLine	Boolean	Specifies whether the text in the control is underlined: TRUE FALSE
UseCodeTable	Boolean	Specifies whether PowerBuilder uses the code table for the column to validate data: TRUE FALSE
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE
VScrollBar	Boolean	Specifies whether a vertical scroll bar displays in the control when not all the data can be displayed at one time: TRUE FALSE
Weight	Integer	Specifies the stroke weight of the text in the control; for example, 400 for normal or 700 for bold.
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.

## Events

<b>Event</b>	<b>Occurs</b>
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters a target control.
DragLeave	When a dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
LoseFocus	When the control loses focus (becomes inactive)
Modified	When a control has been changed and loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

## Functions

Function	Data type returned	Description
CanUndo	Boolean	Returns TRUE if the Undo function can be used to undo the last edit in the control and returns FALSE if it cannot.
ClassName	String	Returns the name assigned to the control.
Clear	Integer	Clears the selected text (if any) from the control but does not place it in the clipboard.
Copy	Integer	Copies (but does not delete) the selected text (if any) from the control to the clipboard.
Cut	Integer	Cuts (deletes) the selected text (if any) from the control and places it in the clipboard.
Drag	Integer	Starts or ends the dragging of the control.
GetData	Integer	Obtains the unformatted data in the control.
Hide	Integer	Makes the control invisible.
LineCount	Integer	Returns the number of lines in the EditMask in the window.
LineLength	Integer	Returns the length of the line in which the cursor is positioned.
Move	Integer	Moves the control to the specified location.
Paste	Integer	Inserts the contents of the clipboard at the insertion point in the specified control.
PointerX	Integer	Returns the distance the pointer is from the left edge of the control.
PointerY	Integer	Returns the distance the pointer is from the top of the control.


<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
Position	Integer	Returns the position of the insertion point in the control.
PostEvent	Boolean	Adds the specified event to the end of the message queue for the control.
Print	Integer	Prints the control.
ReplaceText	Integer	Replaces the currently selected text with the specified string. If no text is selected, inserts the text at the insertion point.
Resize	Integer	Changes the width and height of the control.
Scroll	Integer	Moves the contents of the control up or down the specified number of lines.
SelectedLength	Integer	Returns the total number of characters and spaces (length) in the selected text in the control.
SelectedLine	Integer	Returns the number of the line where the insertion point is located in the control.
SelectedStart	Integer	Returns the position of the first character in the selected text in the control.
SelectedText	String	Determines what if any text is selected in the control.
SelectText	Integer	Selects text in the control beginning at the specified position and continuing for the specified number of characters.
SetFocus	Integer	Sets focus to the control.
SetMask	Integer	Specifies the contents and data type of the edit mask for the control.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its attributes.
Show	Integer	Makes the control visible.
TextLine	String	Returns the entire text of the line in which the insertion point is currently located.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.
Undo	Integer	Cancels the last edit in the control.

# Graph

A graph is a representation of a series of data points (values). The graph can have a single series of values or multiple series.

## Attributes

Attribute	Data type	Description
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215.  For more information about color, see the <i>Function Reference</i> .
Border	Boolean	Specifies whether the control has a border.
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order of the window or DataWindow.
Category	grAxis	Specifies the attributes of the category axis of the graph.
CategorySort	grSortType	Specifies how the categories are sorted.
Depth	Integer	Specifies the percent the depth is of the width of the graph.

Attribute	Data type	Description
DragAuto	Boolean	Specifies whether PowerBuilder will put the graph automatically into DragMode. DragAuto has these boolean values: TRUE - When the control is clicked, the control is automatically in Drag Mode. FALSE - When the control is clicked, the control is not automatically in Drag Mode. You have to manually put the control into Drag Mode by using the Drag function.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags a control (the ICO file). The default icon is a box the size of the control.  When the user drags a control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Elevation	Integer	Specifies the angle of front-to-back elevation.
Enabled	Boolean	Specifies whether the control is enabled (can be selected): TRUE FALSE
FocusRectangle	Boolean	Specifies whether a dotted rectangle (the focus rectangle) will frame the control when it has focus: TRUE FALSE
GraphType	grGraphType (enumerated)	Specifies the type of the graph.
Height	Integer	Specifies the height of the control, in PowerBuilder units.
Legend	grLegendType (enumerated)	Specifies the alignment of the text in the graph legend.

Attribute	Data type	Description
LegendDispAttr	grDispAttr	Specifies the type style for the text in the graph legend, including the text style, size, color, and rotation.
OverlapPercent	Integer	Specifies the percent of the width of the data markers that different series in a graph overlap.
Perspective	Integer	Specifies the distance the graph is from the front of the window.
PieDispAttr	grDispAttr	Specifies attributes of the text in pie graph labels, including the text style, size, color, and rotation.
Pointer	String	Contains the name of the stock pointer or the file containing the pointer used for the graph.
Rotation	Integer	Specifies how much to rotate the graph from left to right.
Series	grAxis	Specifies the series in the graph.
SeriesSort	grSortType	Specifies how the series are sorted.
ShadeColor	Long	Specifies the color used for the shading in the graph.
Spacing	Integer	Specifies the space between data markers in the graph as a percent.
TabOrder	Integer	Specifies the tab value of the control in the tabbing sequence (0 means the user cannot tab to the control).
Tag	String	Specifies the tag value of the control.
TextColor	Long	Specifies the color to be used for the text in the control.
Title	String	Specifies the text of the title for the graph.
TitleDispAttr	grDispAttr	Specifies the type style for the text in the graph legend, including the text style, size, color, and rotation.
Values	grAxis	Specifies the values of the value axis of the graph.



Attribute	Data type	Description
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE
Width	Integer	Specifies the stroke weight of the text in the control; for example, 400 for normal or 700 for bold.
X	Integer	Specifies the X position (the distance from the left edge of the parent window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the parent window), in PowerBuilder units.

## Events

Event	Occurs
Clicked	When the control is clicked (selected or unselected).
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DoubleClicked	When the control is double-clicked (selected or unselected).
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

## Functions

Function	Data type returned	Description
AddCategory	Integer	Adds a category to the graph.
AddData	Long	Adds a value to the end of the specified series for the graph.
AddSeries	Integer	Adds a series to the graph and assigns the series a number.
CategoryCount	Integer	Counts the categories in the graph.
CategoryName	String	Obtains the name of the specified category in the graph.
ClassName	String	Returns the name assigned to the control.
Clipboard	Integer	Copies the graph in bitmap (BMP) format to the clipboard.
DataCount	Long	Returns the number of data points in a specified series in the graph.
DeleteCategory	Integer	Deletes the specified category and the data values in the series from the graph.
DeleteData	Integer	Deletes the data value in the specified data point in the specified series in the graph.
DeleteSeries	Integer	Deletes the specified series and the data values in the series from the graph.
Drag	Integer	Starts or ends dragging of the control.
FindCategory	Integer	Obtains the number of the specified category in the graph.
FindSeries	Integer	Obtains the number PowerBuilder assigned to the specified series when it was added to the graph.
GetData	Double	Obtains the value of the specified data point in the specified series in the graph (superseded by GetDataValue, which is more flexible).
GetDataPieExplode	Integer	Reports the percentage that a pie slice is exploded in a pie graph.

Function	Data type returned	Description
GetDataStyle	Integer	Stores the Fill Pattern enumerated data type representing the pattern used to fill the specified data point in the specified series in the graph.
GetDataValue	Integer	Returns the value of the specified data in the specified series in the specified graph.
GetSeriesStyle	Integer	Stores the FillPattern enumerated data type representing the pattern used to fill the specified series in the graph.
Hide	Integer	Makes the control invisible.
ImportClipboard	Long	Copies the contents of the clipboard to the graph starting in the specified column.
ImportFile	Long	Copies the contents of the specified file to the graph starting in the specified column.
ImportString	Long	Imports the contents from the specified string to the graph starting in the specified column.
InsertCategory	Integer	Inserts a new category before an existing category in the graph.
InsertData	Long	Inserts a new data point into the graph before a specified data point and moves existing data points to the right.
InsertSeries	Integer	Inserts a new series before an existing series in the graph.
ModifyData	Integer	Changes the value of the specified data point in the specified series in the graph to the specified value, and optionally modifies the data for the specified tick mark.
Move	Integer	Moves the control to a specified location.

Function	Data type returned	Description
ObjectAtPointer	GrObjectType	Stores the number of the series the pointer is over in the graph and the number of the specified data point in reference values, and identifies the object type.
PointerX	Integer	Returns the distance the pointer is from the left edge of the control.
PointerY	Integer	Returns the distance the pointer is from the top of the control.
PostEvent	Boolean	Adds an event to the end of the message queue for the control.
Print	Integer	Prints the control.
Reset	Integer (enumerated)	Deletes data in the graph as specified.
ResetDataColors	Integer	Resets the color of a data point to the series color.
Resize	Integer	Changes the size of the control.
SaveAs	Integer	Saves the contents of the graph to a file in the specified format.
SeriesCount	Integer	Determines how many series there are in the graph.
SeriesName	String	Obtains the name of the specified series in the graph.
SetDataPieExplode	Integer	Explodes a pie slice in a pie graph.
SetDataStyle	Integer	Format 1 sets the fill pattern for a data point in a series in a graph.  Format 2 stores the color used for the specified color type (background, shade, or text) for the specified data point in the specified series in the graph.
SetFocus	Integer	Sets the focus to the control.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.
SetRedraw	Integer	Controls automatic redrawing of the control after a change in its attributes.

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
SetSeriesStyle	Integer	Format 1 sets the fill pattern for a series in a graph.  Format 2 sets the color of a color type (background, shade, or text) in a graph.
Show	Integer	Makes the control visible.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

## GroupBox

A GroupBox is a box used to group related controls. For example, you can use a GroupBox to group a series of RadioButtons or CommandButtons. The user cannot select the GroupBox but can select controls within the GroupBox. If the GroupBox contain RadioButtons, the user can select only one RadioButton in the GroupBox at a time.

When you hide or show a GroupBox, PowerBuilder does not automatically hide or show the controls in the GroupBox.

### Events

A GroupBox cannot be selected and has no events.

## Attributes

Attribute	Data type	Description
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215.  <i>ℳ</i> For more information about color, see the <i>Function Reference</i> .
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control.
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order of the window.
DragAuto	Boolean	Specifies whether PowerBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values: TRUE - When the control is clicked, the control is automatically in Drag Mode. FALSE - When the control is clicked, the control is not automatically in Drag Mode. You have to manually put the control into Drag Mode by using the Drag function.

Attribute	Data type	Description
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.  When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected): TRUE FALSE
FaceName	String	Specifies the name of the typeface in which the text of the control displays (for example, ARIAL or COURIER).
FontCharSet	FontCharSet (enumerated)	Specifies the font character set used for the text in the control.
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control.
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control.
Height	Integer	Specifies the height of the control, in PowerBuilder units.
Italic	Boolean	Indicates whether the text in the control is italic: TRUE FALSE
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
TabOrder	Integer	Specifies the tab value of the control within the window (0 means the user cannot tab to the control).
Tag	String	Specifies the tag value assigned to the control.

Attribute	Data type	Description
Text	String	Specifies the text that displays in the control title.
TextColor	Long	Specifies the numeric value of the color used for text: -2 to 16,777,215.  <i>ℳ</i> For more information about color, see the <i>Function Reference</i> .
TextSize	Integer	Specifies the size of the text in the control, in points. For backward compatibility, the size is stored as a negative number; for example, 10-point text size is stored as -10.
Underline	Boolean	Indicates whether the text in the control is underlined: TRUE FALSE
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE
Weight	Integer	Specifies the stroke weight of the text in the control; for example, 400 for normal or 700 for bold.
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuUilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.



## Functions

Function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
Hide	Integer	Makes the control invisible.
Move	Integer	Moves the control to a specified location.
PointerX	Integer	Returns the distance the pointer is from the left edge of the control.
PointerY	Integer	Returns the distance the pointer is from the top of the control.
PostEvent	Boolean	Adds an event to the end of the message queue for the control.
Print	Integer	Prints the control.
Resize	Integer	Changes the size of the control.
SetFocus	Integer	Sets focus to the specified control.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its attributes.
Show	Integer	Makes the control visible.
TriggerEvent	Integer	Triggers the specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

## HScrollBar

An HScrollBar is a horizontal bar with arrows at either end and a scroll box. The scroll box usually indicates a relative or changing amount (for example, the percentage of a task that has been completed).

**Note**

The HScrollBar control is not the horizontal scroll bar that displays to allow the user to scroll through information in a control.

## Attributes

Attribute	Data type	Description
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order: TRUE FALSE
DragAuto	Boolean	Specifies whether PowerBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values: TRUE - When the control is clicked, the control is automatically in Drag Mode. FALSE - When the control is clicked, the control is not automatically in Drag Mode. You have to manually put the control into Drag Mode by using the Drag function.

Attribute	Data type	Description
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.  When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Height	Integer	Specifies the height of the control, in PowerBuilder units.
MaxPosition	Integer	Specifies the value of the Position attribute when the scroll box is at the right edge of the control.
MinPosition	Integer	Specifies the value of the Position attribute when the scroll box is at the left edge of the control.
Pointer	String	Specifies the name of the stock pointer or file containing the pointer used for the control.
Position	Integer	Specifies a value between MinPosition and MaxPosition specifying the position of the scroll box.
StdHeight	Boolean	Specifies whether PowerBuilder uses the standard height for the control: TRUE FALSE
TabOrder	Integer	Specifies the tab value of the control within the window (0 means the user cannot tab to the control).
Tag	String	Specifies the tag value assigned to the control.
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE
Width	Integer	Specifies the width of the control, in PowerBuilder units.

Attribute	Data type	Description
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.

## Events

Event	Occurs
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When a dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
LineLeft	When the left arrow of the control is clicked.
LineRight	When the right arrow of the control is clicked.
LoseFocus	When the control loses focus (becomes inactive).
Moved	When the scroll box is moved (use the Position attribute to determine the new location).
Other	When a Windows message occurs that is not a PowerBuilder event.
PageLeft	When the open space to the left of the scroll box is clicked.
PageRight	When the open space to the right of the scroll box is clicked.
RButtonDown	When the right mouse button is pressed on the control.

## Functions

Function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
Hide	Integer	Makes the control invisible.
Move	Integer	Moves the control to a specified location.
PointerX	Integer	Returns the distance the pointer is from the left edge of the control.
PointerY	Integer	Returns the distance the pointer is from the top of the control.
PostEvent	Boolean	Adds an event to the end of the message queue for control.
Print	Integer	Prints the control.
Resize	Integer	Changes the size of the control.
SetFocus	Integer	Sets the focus to the control.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its attributes.
Show	Integer	Makes the control visible.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.


# Line

A line drawing object is a single, straight solid or dashed line.

## Events

A line cannot be selected and has no events.

## Attributes

Attribute	Data type	Description
BeginX	Integer	Specifies the X position of one end of the line (the distance from the left edge of the window), in PowerBuilder units.
BeginY	Integer	Specifies the Y position of one end of the line (the distance from the top of the window), in PowerBuilder units.
EndX	Integer	Specifies the X position of the other end of the line (the distance from the left edge of the window), in PowerBuilder units.
EndY	Integer	Specifies the Y position of the other end of the line (the distance from the top of the window), in PowerBuilder units.
LineColor	Long	Specifies the numeric value of the line color: -2 to 16,777,215.   For information about specifying color, see the <i>Function Reference</i> .
LineStyle	LineStyle (enumerated)	Specifies the style of the line.
LineThickness	Integer	Specifies the thickness of the line, in PowerBuilder units. If LineThickness is greater than one pixel (about three PowerBuilder units), the LineStyle is Continuous!.

Attribute	Data type	Description
Tag	String	Specifies the tag value assigned to the control.
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE

## Functions

Function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Hide	Integer	Makes the control invisible.
Move	Integer	Moves the control to a specified location (changes the BeginX, BeginY, EndX, and EndY attributes).
Resize	Integer	Changes the length of the control (changes the settings of the BeginX, BeginY, EndX, and EndY attributes).
Show	Integer	Makes the control visible.
TypeOf	Object	Returns the type of the control.

# ListBox

A ListBox displays available options or values. If more options or values exist than can display in the ListBox at one time or the text exceeds the width of the ListBox, the ListBox will have one or two (vertical or horizontal) scroll bars.

## Attributes

Attribute	Data type	Description
Accelerator	Integer	Specifies the ASCII value of the key you want to assign as the accelerator key for a control.
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215.  <i>ℳ</i> For more information about color, see the <i>Function Reference</i> .
Border	Boolean	Specifies whether the control has a border: <b>TRUE</b> <b>FALSE</b>
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control.
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order: <b>TRUE</b> <b>FALSE</b>
DisableNoScroll	Boolean	Specifies behavior of scroll bar: <b>TRUE</b> - The scroll bar will always be visible, but will be disabled when all the items can be accessed without it <b>FALSE</b> - The scroll bar will be displayed only if it is necessary (based on the number of items and the height of the listbox)



Attribute	Data type	Description
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values:</p> <p>TRUE - When the control is clicked, the control is automatically in Drag Mode.</p> <p>FALSE - When the control is clicked, the control is not automatically in Drag Mode. You have to manually put the control into Drag Mode by using the Drag function.</p>
DragIcon	String	<p>Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.</p> <p>When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.</p>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected):</p> <p>TRUE</p> <p>FALSE</p>

Attribute	Data type	Description
ExtendedSelect	Boolean	<p>Specifies whether users can select multiple items in the listbox at one time:</p> <p>TRUE FALSE</p> <p>When ExtendedSelect = TRUE, users can select multiple items by clicking on an item and dragging the mouse up or down to select items; using Click/Shift+Click to select a sequential group of items; or using Control+Click on multiple items.</p> <p><b>Note:</b> The MultiSelect attribute allows users to select multiple items in a listbox by simply clicking on the items. If MultiSelect=TRUE and ExtendedSelect=TRUE, then the behavior of ExtendedSelect takes precedence.</p>
FaceName	String	Specifies the name of the typeface in which the text of the control displays (for example, ARIAL or COURIER).
FontCharSet	FontCharSet (enumerated)	Specifies the font character set used for the text in the control.
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control.
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control.
Height	Integer	Specifies the height of the control, in PowerBuilder units.
HScrollBar	Boolean	<p>Specifies whether a horizontal scroll bar displays:</p> <p>TRUE FALSE</p>
Italic	Boolean	<p>Specifies whether the text in the control is italic:</p> <p>TRUE FALSE</p>

Attribute	Data type	Description
Item[ ]	String	Specifies the items in the control.
MultiSelect	Boolean	Specifies whether users can select multiple items in the ListBox at one time: TRUE FALSE
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
Sorted	Boolean	Specifies whether the items in the ListBox are sorted: TRUE FALSE
TabOrder	Integer	Specifies the tab value of the control (0 means the user cannot tab to the control).
TabStop[ ]	Integer array	Specifies the positions of the tab stops in the ListBox. The tab stops are in character positions, and the tab stop delimiter is a space. If you assign a value to only the first tab stop, TabStop[1], the tab stops are equally spaced using the number of character positions specified for the first tab stop. If more than one tab stop is entered, tab stops are located in the positions specified. You can define 16 tab stops in the control; the default array is TabStop[8], with a tab stop every eight character positions.
Tag	String	Specifies the tag value assigned to the control.
TextColor	Long	Specifies the numeric value of the color used for text: -2 to 16,777,215.  <i>ℳ</i> For more information about color, see the <i>Function Reference</i> .
TextSize	Integer	Specifies the size of the text in the control, in points. For backward compatibility, the size is stored as a negative number; for example, 10-point text size is stored as -10.

Attribute	Data type	Description
Underline	Boolean	Specifies whether the text in the control is underlined: TRUE FALSE
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE
VScrollBar	Boolean	Specifies whether a vertical scroll bar is displayed on the right of the ListBox: TRUE FALSE
Weight	Integer	Specifies the stroke weight of the text in the control; for example, 400 for normal or 700 for bold.
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.

## Events

<b>Event</b>	<b>Occurs</b>
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DoubleClicked	When the control is double-clicked (selected and activated).
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When a dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed on the control.
SelectionChanged	When an item in the control is selected.

## Functions

Function	Data type returned	Description
AddItem	Integer	Adds a new item to the end of the ListBox. If the Sorted attribute of the control is TRUE, the items are sorted again after the item is added.
ClassName	String	Returns the name assigned to the control.
DeleteItem	Integer	Deletes the item indicated by the index from the ListBox.
DirList	Boolean	Populates the ListBox with a list of the files of the specified type that match the specified file pattern.
DirSelect	Boolean	Returns the current selection for the control and puts it in the specified variable.
Drag	Integer	Starts or ends the dragging of a control.
FindItem	Integer	Finds the first item in the ListBox (after the specified index) that begins with the specified string.
Hide	Integer	Makes the control invisible.
InsertItem	Integer	Adds a new item to the ListBox before the item indicated by the index. If the Sorted attribute of the control is TRUE, the items are sorted again after the item is added.
Move	Integer	Moves the control to a specified location.
PointerX	Integer	Returns the distance the pointer is from the left edge of the control.
PointerY	Integer	Returns the distance the pointer is from the top of the control.
PostEvent	Boolean	Adds an event to the end of the message queue for the control.
Print	Integer	Prints the control.
Reset	Integer	Removes all items from the control.
Resize	Integer	Changes the size of the control.

Function	Data type returned	Description
SelectedIndex	Integer	Returns the index of the item in the ListBox that is currently selected. If more than one item is selected, it returns the index of the first selected item.
SelectedItem	String	Returns the text of the first selected item.
SelectItem	Integer	Format 1 finds and highlights the first item in the control after the specified index that begins with the specified string. SelectItem works only if the MultiSelect attribute is FALSE.  Format 2 finds and selects the specified item number.
SetFocus	Integer	Sets the focus to the control.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its attributes.
SetState	Integer	Sets the state (highlighted or not highlighted) of the item indicated by the specified index. SetState works only for multiselect controls (that is, those for which the MultiSelect attribute is TRUE).
SetTop	Integer	Scrolls the items in the control so that the item indicated by the specified index is at the top of the control.
Show	Integer	Makes the control visible.
State	Integer	Returns 1 if the item specified by the specified index is selected (highlighted) and 0 if the item is not selected.
Text	String	Returns the text of the item in the control that is identified by the specified index.
Top	Integer	Returns the index number of the item currently at the top of the control.
TotalItems	Integer	Returns the total number of items in the control.
TotalSelected	Integer	Returns the total number of items selected in the control.


<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.



## MultiLineEdit

A MultiLineEdit is a box in which the user can enter and edit more than one line of text. You typically use a MultiLineEdit as an input field.

### Attributes

Attribute	Data type	Description
Accelerator	Integer	Specifies the ASCII value of the key you want to assign as the accelerator key for a control.
Alignment	Alignment (enumerated)	Specifies the text alignment in the control.
AutoHScroll	Boolean	Specifies whether the control automatically scrolls horizontally when data is entered or deleted: TRUE FALSE
AutoVScroll	Boolean	Specifies whether the control automatically scrolls vertically when data is entered or deleted: TRUE FALSE
BackColor	Long	When AutoVScroll is FALSE, the control wraps. Specifies the numeric value of the background color: -2 to 16,777,215.
Border	Boolean	 For more information about color, see the <i>Function Reference</i> . Specifies whether the control has a border: TRUE FALSE
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control.

Attribute	Data type	Description
BringToTop	Boolean	<p>Specifies whether PowerBuilder moves the control to the top of the front-to-back order in the window:</p> <p>TRUE FALSE</p>
DisplayOnly	Boolean	<p>Specifies whether the text is display-only and cannot be changed by the user:</p> <p>TRUE FALSE</p>
DragAuto	Boolean	<p>Specifies whether PowerBuilder will put the control automatically into Drag Mode. DragAuto has these boolean values:</p> <p>TRUE - When the control is clicked, the control is automatically in Drag Mode. FALSE - When the control is clicked, the control is not automatically in Drag Mode. You have to manually put the control into Drag Mode by using the Drag function.</p>
DragIcon	String	<p>Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.</p> <p>When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.</p>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected):</p> <p>TRUE FALSE</p>
FaceName	String	<p>Specifies the name of the typeface in which the text of the control displays (for example, ARIAL or COURIER).</p>

Attribute	Data type	Description
FontCharSet	FontCharSet (enumerated)	Specifies the font character set used for the text in the control.
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control.
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control.
Height	Integer	Specifies the height of the control, in PowerBuilder units.
HideSelection	Boolean	Specifies whether selected text stays selected (highlighted) even when the control does not have focus: TRUE - Text does not stay highlighted FALSE - Text stays highlighted
HScrollBar	Boolean	Specifies whether a horizontal scroll bar displays: TRUE FALSE
IgnoreDefaultButton	Boolean	Specifies whether the Clicked event for the window's Default command button is triggered when user presses ENTER. TRUE - Do not trigger Clicked event; add new line in MultiLineEdit control. FALSE - Trigger Clicked event; do not add new line in MultiLineEdit control (default).
Italic	Boolean	Specifies whether the text in the control is italic: TRUE FALSE
Limit	Integer	Specifies the maximum number of characters (0 to 32,767) that can be entered in the control (0 means unlimited).
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.

Attribute	Data type	Description
TabOrder	Integer	Specifies the tab value of the control within the window (0 means the user cannot tab to the control).
TabStop[ ]	Integer	Specifies the positions of the tab stops in the control. The tab stops are in character positions, and the tab stop delimiter is a space. If you assign a value to only the first tab stop, TabStop[1], the tab stops are equally spaced using the number of character positions specified for the first tab stop. If more than one tab stop is entered, tab stops are located in the positions specified. You can define 16 tab stops in the control; the default array is TabStop[8], with a tab stop every eight character positions.
Tag	String	Specifies the tag value assigned to the control.
Text	String	Specifies the text that displays in the control.
TextCase	TextCase (enumerated)	Specifies the case in which text entered in the control displays.
TextColor	Long	Specifies the numeric value of the color used for text: -2 to 16,777,215.  <i>For more information about color, see the <i>Function Reference</i>.</i>
TextSize	Integer	Specifies the size of the text in the control, in points. For backward compatibility, the size is stored as a negative number; for example, 10-point text size is stored as -10.
Underline	Boolean	Specifies whether the text in the control is underlined: TRUE FALSE
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE

Attribute	Data type	Description
VScrollBar	Boolean	Specifies whether a vertical scroll bar is displayed on the right of the control: TRUE FALSE
Weight	Integer	Specifies the stroke weight of the text in the control; for example, 400 for normal or 700 for bold.
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.

## Events

<b>Event</b>	<b>Occurs</b>
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When a dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
LoseFocus	When the control loses focus (becomes inactive).
Modified	When a control has been changed and loses focus.
Other	When a Windows message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

## Functions

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
CanUndo	Boolean	Returns TRUE if the Undo function can be used to undo the last edit in the control and returns FALSE if it cannot.
ClassName	String	Returns the name assigned to the control.
Clear	Integer	Clears the selected text (if any) from the control (but does not place it in the clipboard).
Copy	Integer	Copies (but does not delete) the selected text (if any) from the control to the clipboard.
Cut	Integer	Cuts (deletes) the selected text (if any) from the control to the clipboard.
Drag	Integer	Starts or ends the dragging of the control.

Function	Data type returned	Description
Hide	Integer	Makes the control invisible.
LineCount	Integer	Returns the number of lines in the MultiLineEdit.
LineLength	Integer	Returns the length of the line in which the insertion point is positioned.
Move	Integer	Moves the control to a specified location.
Paste	Integer	Inserts the contents of the clipboard (if any) at the insertion point in the control.
PointerX	Integer	Returns the distance the pointer is from the left edge of the control.
PointerY	Integer	Returns the distance the pointer is from the top of the control.
Position	Integer	Returns the position of the insertion point in the control.
PostEvent	Boolean	Adds the specified event to the end of the event queue for the specified object.
Print	Integer	Prints the control.
ReplaceText	Integer	Replaces the currently selected text (if any) with the specified string. If no text is selected, the ReplaceText function inserts the text at the insertion point.
Resize	Integer	Changes the size of the control.
Scroll	Integer	Moves the contents of the control up or down by the specified number of lines.
SelectedLength	Integer	Returns the length of the selected text (if any) in the control.
SelectedLine	Integer	Returns the number of the line in which the insertion point is currently located.
SelectedStart	Integer	Returns the starting position of the selected text (if any) in the control.
SelectedText	String	Returns a string with the selected text (if any) from the control.
SelectText	Integer	Selects the text specified by the starting position and length.
SetFocus	Integer	Sets focus to the specified control.

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its attributes.
Show	Integer	Makes the control visible.
TextLine	String	Returns the entire text of the line in which the insertion point is currently located.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.
Undo	Integer	Cancels the previous editing function performed in the control.



## OLE 2.0

An OLE 2.0 control contains an object, such as a spreadsheet or word processing document, that was created by an OLE 2.0-aware application. The PowerBuilder application's user can activate the object and edit it in the application in which it was created (the *server* application).

### Attributes

Attribute	Data type	Description
Activation	omActivation	Specifies how the OLE object will be activated.
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. <i>↪</i> For more information about color, see the <i>Function Reference</i> .
Border	Boolean	Specifies whether the control has a border.
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order: TRUE FALSE
ClassLongName	String	The long name for the server application associated with the OLE object in the control (read-only).
ClassShortName	String	The short name for the server application associated with the OLE object in the control (read-only).
ContentsAllowed	omContents-Allowed	Specifies whether the OLE object in the control must be embedded or linked or whether either method is allowed.
DisplayName	String	User-readable name for your OLE control. This name is displayed in OLE dialog boxes and windows that show the object's name. If you don't

Attribute	Data type	Description
DisplayType	omDisplayType	<p>specify a value, the name of the control (such as ole_1) is used for DisplayName.</p> <p>Specifies how the OLE object will be displayed in the control. The control can display the actual contents or an icon to represent the object.</p>
DocFileName	String	<p>The name of an OLE storage file or a data file of the server application that has been opened for the control (read-only).</p>
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values:</p> <p>TRUE - When the control is clicked, the control is automatically in Drag Mode.</p> <p>FALSE - When the control is clicked, the control is not automatically in Drag Mode. You have manually put the control into Drag Mode by using the Drag function.</p>
DragIcon	String	<p>Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.</p> <p>When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.</p>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected):</p> <p>TRUE</p> <p>FALSE</p>

Attribute	Data type	Description
FocusRectangle	Boolean	Specifies whether a dotted rectangle (the focus rectangle) will frame the control when it has focus: TRUE FALSE
Height	Integer	Specifies the height of the control, in PowerBuilder units.
IsDragTarget	Boolean	Specifies whether an OLE object can be dropped on the control: TRUE FALSE
LinkItem	String	The name of an item within the server application's data file to which the control is linked (read-only).
LinkUpdateOptions	omLinkUpdateOptions	Specifies how a linked object in the control will be updated. If automatic, the link will be updated when the object is opened and whenever the object changes in the server application. If manual, the link will not be updated.
Object	omObject	The link information that connects the control to the server's data.
ObjectData	Blob	If the object is embedded, the object itself is stored as a blob in the ObjectData attribute.
ParentStorage	OMStorage	Specifies the parent storage (read-only).
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
TabOrder	Integer	Specifies tab value of the control within the window (0 means the user cannot tab to the control).
Tag	String	Specifies the tag value assigned to the control.
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE

<b>Attribute</b>	<b>Data type</b>	<b>Description</b>
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.

## Events

<b>Event</b>	<b>Occurs</b>
Clicked	When the control is clicked (selected or unselected).
Close	Just before a window is removed from display (closed).
Constructor	Immediately before the Open event occurs in the window.
DataChange	When the server application notifies the control that data has changed.
Destructor	Immediately after the Close event occurs in the window.
DoubleClicked	When the control is double-clicked (and possibly activated).
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When a dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Just before the control receives focus (and possibly becomes activated).
LoseFocus	When the control loses focus (becomes inactive).
Other	When an operating environment message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed in the control
Rename	When the server application notifies the control that the

Event	Occurs
	object has been renamed.
Save	When the server application notifies the control that the data has been saved.
ViewChange	When the server application notifies the control that the view shown to the user has changed.

## Functions

Function	Data type returned	Description
Activate	Integer	Activates the object server either in place or offsite.
ClassName	String	Returns the name assigned to the control.
Clear	Integer	Releases the OLE object and deletes references to it without updating storage.
Copy	Integer	Copies the contents of the control to the clipboard.
Cut	Integer	Copies the contents of the control to the clipboard and clears the control.
DoVerb	Integer	Execute the specified verb.
Drag	Integer	Puts the object into drag mode.
Hide	Integer	Makes the control invisible.
InsertClass	Integer	Inserts a new object created from a class ID.
InsertFile	Integer	Inserts a new object where the source is a template file.
InsertObject	Integer	Presents the user with a standard dialog box and inserts based on the selection.
LinkTo	Integer	Links to a file and (optionally) an item within the file.
Move	Integer	Moves the control to a specified location.

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
Open	Integer	Opens a document file and loads the object into the control or opens a substorage within the previously opened storage and loads an object.
Paste	Integer	Pastes the contents of the clipboard into the control.
PasteLink	Integer	Pastes a link to the contents of the clipboard into the control.
PasteSpecial	Integer	Presents the user with a dialog box allowing them to select Paste or PasteLink.
PointerX	Integer	Returns the distance the pointer is from the left edge of the control.
PointerY	Integer	Returns the distance the pointer is from the top of the control.
PostEvent	Boolean	Adds an event to the end of the message queue for the control.
Print	Integer	Prints the control.
Resize	Integer	Changes the size of the control.
Save	Integer	Saves an object previously loaded from a storage.
SaveAs	Integer	Saves the contained object as a member in the requested storage or saves the contained object to the requested storage file.
SelectObject	Integer	Sets the internal state of the control (updates menu).
SetFocus	Integer	Sets the focus to the control.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its attributes
Show	Integer	Makes the control visible.
TriggerEvent	Integer	Triggers a specific event for the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

## Oval

An oval is a filled or outlined round or elliptical drawing object that you typically use for design effects (for example, you can put a `CommandButton` or a picture in an oval). The grouping does not affect the behavior of the controls in the oval.

### Events

An oval cannot be selected and has no events.

## Attributes

Attribute	Data type	Description
FillColor	Long	Specifies the numeric value of the color used to fill the control: -2 to 16,777,215.  <i>ℳ</i> For information on color, see the <i>Function Reference</i> .
FillPattern	FillPattern (enumerated)	Specifies the hatch pattern used to fill the control.  FDiagonal! is lines going from the lower left to the upper right. BDiagonal! is lines going from the upper left to the lower right.
Height	Integer	Specifies the height of the control, in PowerBuilder units.
LineColor	Long	Specifies the numeric value of the line color: -2 to 16,777,215.  <i>ℳ</i> For information on color, see the <i>Function Reference</i> .
LineStyle	LineStyle (enumerated)	Specifies the style of the line used to draw the control.
LineThickness	Integer	Specifies the thickness of the line used to draw the control, in PowerBuilder units. If LineThickness is greater than one pixel (about four PowerBuilder units), the LineStyle is Continuous!.

---

<b>Attribute</b>	<b>Data type</b>	<b>Description</b>
Tag	String	Specifies the tag value assigned to the control.
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.

## Functions

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
ClassName	String	Returns the name assigned to the control.
Hide	Integer	Makes the control invisible.
Move	Integer	Moves the control to a specified location.
Resize	Integer	Changes the size of the control.
Show	Integer	Makes the control visible.
TypeOf	Object	Returns the type of the control.



## Picture

A picture is a bitmap image (a BMP or RLE file) or an image in a Windows metafile (WMF file). You can create the image in another application or use a scanner to create it. The picture must be stored in a Microsoft Windows 3.x BMP, RLE, or WMF format.

## Attributes

Attribute	Data type	Description
Border	Boolean	Specifies whether the control has a border: TRUE FALSE
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control.
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order of the window.
DragAuto	Boolean	Specifies whether PowerBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values: TRUE - When the control is clicked, the control is automatically in Drag Mode. FALSE - When the control is clicked, the control is not automatically in Drag Mode. You have to manually put the control into Drag Mode by using the Drag function.

Attribute	Data type	Description
DragIcon	String	<p>Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.</p> <p>When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.</p>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected):</p> <p>TRUE FALSE</p>
FocusRectangle	Boolean	<p>Specifies whether a dotted rectangle (focus rectangle) frames the picture when it has focus.</p>
Height	Integer	<p>Specifies the height of the control, in PowerBuilder units.</p>
Invert	Boolean	<p>Specifies whether the control displays with its colors inverted:</p> <p>TRUE FALSE</p>
OriginalSize	Boolean	<p>Specifies whether the width and height attributes of a bitmap image (picture) will be set to their original values:</p> <p>TRUE FALSE</p> <p>In the Window painter, setting OriginalSize to TRUE overrides the existing width and height. You cannot change this attribute in a script.</p>
PictureName	String	<p>Specifies the name of the file that contains the picture. The file extension .BMP, .RLE, or .WMF is required.</p>
Pointer	String	<p>Specifies the name of the stock pointer or the file containing the pointer used for the control.</p>

Attribute	Data type	Description
TabOrder	Integer	Specifies the tab value of the picture within the window (0 means the user cannot tab to the control).
Tag	String	Specifies the tag value assigned to the control.
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (distance from the top of the window), in PowerBuilder units.

## Events

<b>Event</b>	<b>Occurs</b>
Clicked	When the control is clicked (selected).
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DoubleClicked	When the control is double-clicked (selected and activated).
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When a dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

## Functions

Function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
Draw	Integer	Draws a picture in the parent window at a specified location.
Hide	Integer	Makes the control invisible.
Move	Integer	Moves the control to a specified location.
PointerX	Integer	Returns the distance the pointer is from the left edge of the control.
PointerY	Integer	Returns the distance the pointer is from the top of the control.
PostEvent	Boolean	Adds an event to the end of the message queue for the control.
Print	Integer	Prints the control.
Resize	Integer	Changes the size of the control.
SetFocus	Integer	Sets the focus to the control.
SetPicture	Integer	Constructs a new bitmap for the control.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its attributes.
Show	Integer	Makes the control visible.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

# PictureButton

A PictureButton displays a picture (a bitmap or Windows metafile) and, like a CommandButton, is used to carry out an action. For example, you can use a button with a picture of a file to save a file or a button with a picture of a stop sign to cancel a requested deletion.

## Attributes

Attribute	Data type	Description
Alignment	HTextAlign	Specifies how the text in the control is aligned.
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order of the window.
Cancel	Boolean	Specifies whether the control acts as the Cancel button (the Cancel button receives a Clicked event if the user presses ESC): TRUE FALSE
Default	Boolean	Specifies whether the control is the default PictureButton (the default PictureButton has a thick border and receives a clicked event if the user presses ENTER without selecting an control): TRUE FALSE
DisabledName	String	Specifies the name of the picture (bitmap image) that displays when the control is disabled. If the string has no extension, PowerBuilder adds .BMP, .RLE, or .WMF.

Attribute	Data type	Description
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values:</p> <p>TRUE - When the control is clicked, the control is automatically in Drag Mode</p> <p>FALSE - When the control is clicked, the control is not automatically in Drag Mode (you have to manually put the control into Drag Mode by using the Drag function)</p>
DragIcon	String	<p>Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.</p> <p>When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.</p>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected):</p> <p>TRUE</p> <p>FALSE</p>
FaceName	String	<p>Specifies the name of the typeface in which the text of the control displays (for example, ARIAL or COURIER).</p>
FontCharSet	FontCharSet (enumerated)	<p>Specifies the font character set used for the text in the control.</p>
FontFamily	FontFamily (enumerated)	<p>Specifies the font family (type style) used for the text in the control.</p>
FontPitch	FontPitch (enumerated)	<p>Specifies the pitch (spacing) of the font used for the text in the control.</p>
Height	Integer	<p>Specifies the height of the control, in PowerBuilder units.</p>

Attribute	Data type	Description
Italic	Boolean	Specifies whether the text in the control is italic: TRUE FALSE
OriginalSize	Boolean	Specifies whether the width and height attributes of a bitmap image (picture) are set to their original values: TRUE FALSE  In the Window painter, setting OriginalSize to TRUE overrides the existing width and height. You cannot change this attribute in a script.
PictureName	String	Specifies the name of the file that contains the picture. The file extension .BMP or .RLE is required.
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
TabOrder	Integer	Specifies the tab value of the control within the window (0 means the user cannot tab to the control).
Tag	String	Specifies the tag value assigned to the control.
Text	String	Specifies the text that displays in the control.
TextSize	Integer	Specifies the size of the text in the control, in points. For backward compatibility, the size is stored as a negative number; for example, 10-point text size is stored as -10.
Underline	Boolean	Specifies whether the text in the control is underlined: TRUE FALSE
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE



<b>Attribute</b>	<b>Data type</b>	<b>Description</b>
VTextAlign	VTextAlign	Specifies how the text in the control is aligned.  All these values except MultiLine! assume there is only one line of text.
Weight	Integer	Specifies the stroke weight of the text in the control; for example, 400 for normal or 700 for bold.
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from top of the window), in PowerBuilder units.

## Events

<b>Event</b>	<b>Occurs</b>
Clicked	When the control is clicked.
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When a dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

## Functions

Function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of a control.
Hide	Integer	Makes the control invisible.
Move	Integer	Moves the control to a specified location.
PointerX	Integer	Returns the distance the pointer is from the left edge of the control.
PointerY	Integer	Returns the distance the pointer is from the top of the control.
PostEvent	Boolean	Adds an event to the end of the message queue for control.
Print	Integer	Prints the control..
Resize	Integer	Changes the size of the control.
SetFocus	Integer	Sets focus to the specified control.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its attributes.
Show	Integer	Makes the control visible.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

## RadioButton

A RadioButton is a small round button that is used to turn an option on and off. When the option is on, the button has a dark center. When the option is off, the center is blank. RadioButtons are often grouped in a GroupBox.


### Tip

When RadioButtons are in a GroupBox, the user can select only one button in the group, and the group usually has a default button.

## Attributes

Attribute	Data type	Description
Automatic	Boolean	Specifies whether the control becomes dark when it is clicked: TRUE FALSE
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. <i>↳ For more information about color, see the <a href="#">Function Reference</a>.</i>
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control.
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order in the window: TRUE FALSE
Checked	Boolean	Specifies whether the item is selected (the center is dark): TRUE FALSE

Attribute	Data type	Description
DragAuto	Boolean	Specifies whether PowerBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values: <b>TRUE</b> - When the control is clicked, the control is automatically in Drag Mode. <b>FALSE</b> - When the control is clicked, the control is not automatically in Drag Mode. You have to manually put the control into Drag Mode by using the Drag function.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.  When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected): <b>TRUE</b> <b>FALSE</b>
FaceName	String	Specifies the name of the typeface in which the text of the control displays (for example, HELV or COURIER).
FontCharSet	FontCharSet (enumerated)	Specifies the font character set used for the text in the control.
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control.
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control.
Height	Integer	Specifies the height of the control, in PowerBuilder units.

Attribute	Data type	Description
Italic	Boolean	Specifies whether the text in the control is italic: TRUE FALSE
LeftText	Boolean	Specifies whether the text displays to the left of the control: TRUE FALSE
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
TabOrder	Integer	Specifies the tab value of the control (0 means the user cannot tab to the control). In a GroupBox, the up and down arrow keys are used to move among RadioButtons in a specified sequence.  To permit tabbing in a GroupBox, change the tab value of the GroupBox to 0 and assign nonzero tab values to the RadioButtons (the default tab value for the RadioButtons in a GroupBox is 0).
Tag	String	Specifies the tag value assigned to the control.
Text	String	Specifies the text that displays next to the control.
TextColor	Long	Specifies the numeric value of the color used for text: -2 to 16,777,215.   For more information about color, see the <i>Function Reference</i> .
TextSize	Integer	Specifies the size of the text in the control, in points. For backward compatibility, the size is stored as a negative number; for example, 10-point text size is stored as -10.
Underline	Boolean	Specifies whether the text in the control is underlined: TRUE FALSE

Attribute	Data type	Description
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE
Weight	Integer	Specifies the stroke weight of the text in the control; for example, 400 for normal or 700 for bold.
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.

## Events

Event	Occurs
Clicked	When the control is clicked (selected or unselected).
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When a dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

## Functions

Function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
Hide	Integer	Makes the control invisible.
Move	Integer	Moves the control to a specified location.
PointerX	Integer	Returns the distance the pointer is from the left edge of the control.
PointerY	Integer	Returns the distance the pointer is from the top of the control.
PostEvent	Boolean	Adds an event to the end of the message queue for control.
Print	Integer	Prints the control.
Resize	Integer	Changes the size of the control.
SetFocus	Integer	Sets the focus to the control.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its attributes.
Show	Integer	Makes the control visible.
TriggerEvent	Integer	Triggers a specified event for the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.



# Rectangle

A rectangle is a filled or outlined rectangular form within a window and is typically used for design purposes. For example, you can put a `CommandButton` or a picture in a rectangle, or you can use a rectangle behind and slightly offset from another control to create a shadow effect. When you use a rectangle to group controls, the grouping does not affect the behavior of the controls in the rectangle.

## Events

A rectangle cannot be selected and has no events.

## Attributes

Attribute	Data type	Description
FillColor	Long	Specifies the numeric value of the color used to fill the control: -2 to 16,777,215.  ↪ For more information about color, see the <i>Function Reference</i> .
FillPattern	FillPattern (enumerated)	Specifies the hatch pattern used to fill the control.  FDiagonal! is lines going from the lower left to the upper right BDiagonal! is lines going from the upper left to the lower right.
Height	Integer	Specifies the height of the control, in PowerBuilder units.
LineColor	Long	Specifies the numeric value of the line color: -2 to 16,777,215.  ↪ For more information about color, see the <i>Function Reference</i> .
LineStyle	LineStyle (enumerated)	Specifies the pattern of the line used to draw the control.

Attribute	Data type	Description
LineThickness	Integer	Specifies the thickness of the line used to draw the control, in PowerBuilder units. If LineThickness is greater than one pixel (about four PowerBuilder units), the LineStyle is forced to Continuous!
Tag	String	Specifies the tag value assigned to the control.
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.

## Functions

Function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Hide	Integer	Makes the control invisible.
Move	Integer	Moves the control to a specified location.
Resize	Integer	Changes the size of the control.
Show	Integer	Makes the control visible.
TypeOf	Object	Returns the type of the control.



## RoundRectangle

A RoundRectangle is a filled or outlined rectangular drawing object with rounded corners that you typically use for design purposes (for example, you can put a CommandButton or a picture in a RoundRectangle). When you use a RoundRectangle to group controls, the grouping does not affect the behavior of the controls in the RoundRectangle.

### Events

A RoundRectangle cannot be selected and has no events.

## Attributes

Attribute	Data type	Description
CornerHeight	Integer	Specifies the radius of the vertical part of the corners of the control, in PowerBuilder units.
CornerWidth	Integer	Specifies the radius of the horizontal part of the corners of the control, in PowerBuilder units.
FillColor	Long	Specifies the numeric value of the color used to fill the control: -2 to 16,777,215.   For more information about color, see the <i>Function Reference</i> .
FillPattern	FillPattern (enumerated)	Specifies the hatch pattern used to fill the control.  FDiagonal! is lines going from the lower left to the upper right. BDiagonal! is lines going from the upper left to the lower right.
Height	Integer	Specifies the height of the control, in PowerBuilder units.
LineColor	Long	Specifies the numeric value of the line color: -2 to 16,777,215.   For more information about color, see the <i>Function Reference</i> .

Attribute	Data type	Description
LineStyle	LineStyle (enumerated)	Specifies the style of the line used to draw the control.
LineThickness	Integer	Specifies the thickness of the line used to draw the control, in PowerBuilder units. If LineThickness is greater than one pixel (about four PowerBuilder units), the LineStyle is Continuous!
Tag	String	Specifies the tag value assigned to the control.
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.


## Functions

Function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Hide	Integer	Makes the control invisible.
Move	Integer	Moves the control to a specified location.
Resize	Integer	Changes the size of the control.
Show	Integer	Makes the control visible.
TypeOf	Object	Returns the type of the control.


# SingleLineEdit

A `SingleLineEdit` is a box in which the user can enter a single line of text. You typically use a `SingleLineEdit` as an input field.

## Attributes

Attribute	Data type	Description
Accelerator	Integer	The ASCII value of the key you want to assign as the accelerator key for a control.
AutoHScroll	Boolean	Specifies whether the control automatically scrolls horizontally when data is entered or deleted: TRUE FALSE
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215.   For more information about color, see the <i>Function Reference</i> .
Border	Boolean	Specifies whether the control has a border: TRUE FALSE
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control.
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order in the window: TRUE FALSE
DisplayOnly	Boolean	Specifies whether the text in the control is display-only and cannot be changed by the user: TRUE FALSE

Attribute	Data type	Description
DragAuto	Boolean	Specifies whether PowerBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values: TRUE - When the control is clicked, the control is automatically in Drag Mode. FALSE - When the control is clicked, the control is not automatically in Drag Mode. You have to manually put the control into Drag Mode by using the Drag function.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.  When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected): TRUE FALSE
FaceName	String	Specifies the name of the typeface in which the text of the control displays (for example, ARIAL or COURIER).
FontCharSet	FontCharSet (enumerated)	Specifies the font character set used for the text in the control.
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control.
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control.
Height	Integer	Specifies the height of the control, in PowerBuilder units.

Attribute	Data type	Description
HideSelection	Boolean	Specifies whether selected text stays selected (highlighted) even when the control does not have focus: TRUE (text does not stay highlighted) FALSE (text stays highlighted)
Italic	Boolean	Specifies whether the text in the control is italic: TRUE FALSE
Limit	Integer	Specifies the maximum number of characters (0 to 32,767) that can be entered in the control (0 means unlimited).
Password	Boolean	Specifies whether the control is a password field (whether asterisks appear when the user types characters): TRUE FALSE
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
TabOrder	Integer	Specifies the tab value of the control within the window (0 means the user cannot tab to the control).
Tag	String	Specifies the tag value assigned to the control.
Text	String	Specifies the text that displays in the control.
TextCase	TextCase (enumerated)	Specifies the case in which text entered in the control displays.
TextColor	Long	Specifies the numeric value of the color used for text: -2 to 16,777,215.   For more information about color, see the <i>Function Reference</i> .
TextSize	Integer	Specifies the size of the text in the control, in points. For backward compatibility, the size is stored as a negative number; for example, 10-point text size is stored as -10.

<b>Attribute</b>	<b>Data type</b>	<b>Description</b>
Underline	Boolean	Specifies whether the text in the control is underlined: TRUE FALSE
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE
Weight	Integer	Specifies the stroke weight of the text in the control; for example, 400 for normal or 700 for bold.
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.

## Events

<b>Event</b>	<b>Occurs</b>
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When a dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
LoseFocus	When the control loses focus (becomes inactive).
Modified	When the control text has been changed and the user presses ENTER or the TAB key or changes focus to another control.



Event	Occurs
Other	When a Windows message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

## Functions

Function	Data type returned	Description
CanUndo	Boolean	Returns TRUE if the Undo function can be used to undo the last edit in the control and returns FALSE if it cannot.
ClassName	String	Returns the name assigned to the control.
Clear	Integer	Clears the selected text (if any) from the control (but does not place it in the clipboard).
Copy	Integer	Copies (but does not delete) the selected text (if any) from the control to the clipboard.
Cut	Integer	Cuts (deletes) the selected text (if any) from the control and places it in the clipboard.
Drag	Integer	Starts or ends the dragging of a control.
Hide	Integer	Makes the control invisible.
Move	Integer	Moves the control to a specified location.
Paste	Integer	Inserts the contents of the clipboard (if any) at the insertion point in the control. The Paste function replaces the selected text (if any).
PointerX	Integer	Returns the distance the pointer is from the left edge of the control.
PointerY	Integer	Returns the distance the pointer is from the top of the control.
Position	Integer	Returns the position of the insertion point in the control.

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
PostEvent	Booleanr	Adds an event to the end of the message queue for control.
Print	Integer	Prints the control.
ReplaceText	Integer	Replaces the currently selected text (if any) with the specified string. If no text is selected, inserts the text at the current insertion point.
Resize	Integer	Changes the size of the control.
SelectedLength	Integer	Returns the length of the selected text (if any) in the control.
SelectedStart	Integer	Returns the starting position of the selected text (if any) in the control.
SelectedText	String	Returns a string with the selected text (if any) from the control.
SelectText	Integer	Selects the text in the control specified by the starting position and length.
SetFocus	Integer	Sets focus to the control.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its attributes.
Show	Integer	Makes the control visible.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.
Undo	Integer	Cancel the previous editing function performed in the control.


# StaticText

StaticText is display text that the user can select but cannot modify with the keyboard. You can explicitly modify the StaticText in a script.

## Attributes

Attribute	Data type	Description
Alignment	Alignment (enumerated)	Specifies the text alignment in the control.
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215.  ✍ For more information about color, see the <i>Function Reference</i> .
Border	Boolean	Specifies whether the control has a border: TRUE FALSE
BorderColor	Long	Numerical value of the border color: -2 to 16,777,215.  ✍ For more information about colors, see the <i>Function Reference</i> .
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control.
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order of the window: TRUE FALSE

Attribute	Data type	Description
DragAuto	Boolean	Specifies whether PowerBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values: TRUE - When the control is clicked, the control is automatically in Drag Mode. FALSE - When the control is clicked, the control is not automatically in Drag Mode. You have to manually put the control into Drag Mode by using the Drag function.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.  When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected): TRUE FALSE
FaceName	String	Specifies the name of the typeface in which the text of the control will display (for example, ARIAL or COURIER).
FillPattern	FillPattern (enumerated)	Specifies the hatch pattern used to fill the control.  FDiagonal! is lines going from the lower left to the upper right. BDiagonal! is lines going from the upper left to the lower right.
FocusRectangle	Boolean	Specifies whether a dotted rectangle (focus rectangle) frames the control when it has focus: TRUE FALSE

Attribute	Data type	Description
FontCharSet	FontCharSet (enumerated)	Specifies the font character set used for the text in the control.
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control.
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control.
Height	Integer	Specifies the height of the rectangular box that contains the control, in PowerBuilder units.
Italic	Boolean	Specifies whether the text in the control is italic: TRUE FALSE
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
TabOrder	Integer	Specifies the tab value of the control within the window (0 is the default and means the user cannot tab to the control).
Tag	String	Specifies the tag value assigned to the control.
Text	String	Specifies the text that displays in the control.
TextColor	Long	Specifies the numeric value of the text color in the control: -2 to 16,777,215.   For more information about color, see the <i>Function Reference</i> .
TextSize	Integer	Specifies the size of the text in the control, in points. For backward compatibility, the size is stored as a negative number; for example, 10-point text size is stored as -10.
Underline	Boolean	Specifies whether the text in the control is underlined: TRUE FALSE
Visible	Boolean	Specifies whether the control is visible: TRUE FALSE

Attribute	Data type	Description
Weight	Integer	Specifies the stroke weight of the text in the control; for example, 400 for normal or 700 for bold.
Width	Integer	Specifies the width of the rectangular box that contains the control, in pixels.
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.

## Events

Event	Occurs
Clicked	When the control is clicked (selected).
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DoubleClicked	When the control is double-clicked (selected and activated).
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When a dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

## Functions


Function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
Hide	Integer	Makes the control invisible.
Move	Integer	Moves the control to a specified location.
PointerX	Integer	Returns the distance the pointer is from the left edge of the control.
PointerY	Integer	Returns the distance the pointer is from the top of the control.
PostEvent	Boolean	Adds an event to the end of the message queue for the control.
Print	Integer	Prints the control.
Resize	Integer	Changes the size of the rectangular box that contains the control.
SetFocus	Integer	Sets the focus to the control.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its attributes.
Show	Integer	Makes the control visible.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

## User object

User objects are reusable objects that you can build to supplement the standard PowerBuilder objects. User objects can display information, request information from a user, and respond to mouse or keyboard actions.

When you place a visible user object in a window, you are actually placing a user object *control* in the window. The control holds an instance of the user object you select for the window. You build user objects in the User Object painter.

## Attributes

Attribute	Data type	Description and values
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215.  For more information about color, see the <i>Function Reference</i> .
Border	Boolean	Specifies whether the control has a border: TRUE FALSE
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control.
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order of the window: TRUE FALSE
ClassName	String	(External user objects only) Returns the name assigned to the control



Attribute	Data type	Description and values
ColumnsPerPage	Integer	Specifies the number of columns on a page. The default is 0 (10 columns per page). PowerBuilder multiplies UnitsPerColumn by ColumnsPerPage to determine the number of PowerBuilder units to scroll the control horizontally when the user clicks in the scroll bar.
Control[ ]	WindowObject	Specifies the control's objects. You cannot change the contents of this array in a script.
DragAuto	Boolean	Specifies whether PowerBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values: TRUE - When the control is clicked, the control is automatically in Drag Mode. FALSE - When the control is not clicked, the control is not automatically in Drag Mode. You have to manually put the control into Drag Mode by using the Drag function.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.  When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected): TRUE FALSE
Height	Integer	Specifies the height of the control, in PowerBuilder units.

Attribute	Data type	Description and values
HScrollBar	Boolean	Specifies whether a horizontal scroll bar displays: TRUE FALSE
LibraryName	String	(External user objects only) The name of the dynamic-link library (DLL) that contains an external user object class.
LinesPerPage	Integer	Specifies the number of lines on a page. The default is 0 (10 lines per page).  PowerBuilder multiplies UnitsPerLine by LinesPerPage to determine the number of PowerBuilder units to scroll the control vertically when the user clicks in the scroll bar.
ObjectType	UserObjects (enumerated)	Specifies the type of user object.
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
Style	Long	Specifies any additional style bits that you want to use to control how the control displays (external user object only).
TabOrder	Integer	Specifies tab value of the control within the user object (0 means the user cannot tab to the control).
Tag	String	Specifies the tag value assigned to the control.
Text	String	Specifies the text that displays in the control.

Attribute	Data type	Description and values
UnitsPerColumn	Integer	<p>Specifies the number of PowerBuilder units you want to scroll right or left when the user clicks the left or right arrow in the horizontal scroll bar in a control. The default is 0 (1/100 of the width of the window). When UnitsPerColumn is 0, PowerBuilder controls horizontal scrolling automatically.</p> <p>PowerBuilder multiplies UnitsPerColumn by ColumnsPerPage to determine the number of PowerBuilder units to scroll the user object horizontally when the user clicks in the scroll bar.</p>
UnitsPerLine	Integer	<p>Specifies the number of PowerBuilder units you want to scroll up or down when the user clicks the up or down arrow in the vertical scroll bar in a control. The default is 0 (1/100 of the user object height). When UnitsPerLine is 0, PowerBuilder controls vertical scrolling automatically.</p> <p>PowerBuilder multiplies UnitsPerPage by UnitsPerLine to determine the number of PowerBuilder units to scroll the control vertically when the user clicks in the scroll bar.</p>
Visible	Boolean	<p>Specifies whether the control is visible:</p> <p>TRUE FALSE</p>
VScrollBar	Boolean	<p>Specifies whether a vertical scroll bar displays:</p> <p>TRUE FALSE</p>
Width	Integer	<p>Specifies the width of the control, in PowerBuilder units.</p>

Attribute	Data type	Description and values
X	Integer	Specifies the X position (distance from the left edge of screen) of the control, in PowerBuilder units.
Y	Integer	Specifies the Y position (distance from the top of screen) of the control, in PowerBuilder units.

## Events

Event	Occurs
Clicked	(Does not apply to custom visual user objects) When the control is clicked (selected or unselected).
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When a dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	(Does not apply to custom visual user objects) Just before the control receives focus (before it is selected and becomes active).
LoseFocus	(Does not apply to custom visual user objects) When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

## Functions

Function	Data type returned	Description
AddItem	Integer	(VBX user objects supporting AddItem method only) Adds item to list.
ClassName	String	Returns the name assigned to the control
DeleteItem	Integer	(VBX user objects supporting RemoveItem method only) Deletes item from list.
Drag	Integer	Starts or ends the dragging of the control.
EventParmDouble	Integer	Stores the specified numeric event parameter in a double variable.
EventParmString	Integer	Stores the specified string event parameter in a string variable.
Hide	Integer	Makes the control invisible.
InsertItem	Integer	(VBX user objects supporting AddItem method only) Inserts item in list.
Move	Integer	Places the control in a new location specified by the X and Y arguments.
PointerX	Integer	Returns the distance from the left edge of the screen to the pointer, in PowerBuilder units.
PointerY	Integer	Returns the distance from the top of the screen to the pointer, in PowerBuilder units.
PostEvent	Boolean	Adds an event to the end of the message queue for the control.
Print	Integer	Prints control.
Resize	Integer	Changes the size of the control based on the width and height.
SetFocus	Integer	Sets the focus to the control.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.
SetRedraw	Integer	Turns on or off automatic redrawing of the control after every change.

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
Show	Integer	Makes the control visible.
TriggerEvent	Integer	Sends an event to the control and executes the script associated with the event.
TypeOf	Object	Returns the type of the control.

## VScrollBar

A VScrollBar is a vertical bar with arrows at either end and a scroll box. You typically use a VScrollBar to indicate a relative or changing amount (for example, to indicate the percentage of a task that has been completed).

### Note

The VScrollBar control is not the vertical scroll bar that displays to allow the user to scroll through information in a control or window.

## Attributes

Attribute	Data type	Description
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order of the window: TRUE FALSE
DragAuto	Boolean	Specifies whether PowerBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values: TRUE - When the control is clicked, the control is automatically in Drag Mode. FALSE - When the control is clicked, the control is not automatically in Drag Mode. You have to manually put the control into Drag Mode by using the Drag function.

Attribute	Data type	Description
DragIcon	String	<p>Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control.</p> <p>When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.</p>
Height	Integer	Specifies the height of the control, in PowerBuilder units.
MaxPosition	Integer	Specifies the value of the Position attribute when the scroll box is at the bottom of the scroll bar.
MinPosition	Integer	Specifies the value of the Position attribute when the scroll box is at the top of the scroll bar.
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
Position	Integer	Specifies the value between MinPosition and MaxPosition that indicates the position of the scroll box.
StdWidth	Boolean	<p>Specifies whether the standard scroll bar width is used for the VScrollBar:</p> <p>TRUE FALSE</p>
TabOrder	Integer	Specifies the tab value of the control within the window (0 means the user cannot tab to the control).
Tag	String	Specifies the tag value assigned to the control.
Visible	Boolean	<p>Specifies whether the control is visible:</p> <p>TRUE FALSE</p>
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.



Attribute	Data type	Description
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.

## Events

Event	Occurs
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When a dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
LineDown	When the down arrow of the control is clicked.
LineUp	When the up arrow of the control is clicked.
LoseFocus	When the control loses focus (becomes inactive).
Moved	When the scroll box is moved (use the Position attribute to determine the new location).
Other	When a Windows message occurs that is not a PowerBuilder event.
PageDown	When the open space below the scroll box is clicked.
PageUp	When the open space above the scroll box is clicked.
RButtonDown	When the right mouse button is pressed on the control.

## Functions

Function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
Hide	Integer	Makes the control invisible.

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
Move	Integer	Moves the control to a specified location.
PointerX	Integer	Returns the distance the pointer is from the left edge of the control.
PointerY	Integer	Returns the distance the pointer is from the top of the control.
PostEvent	Boolean	Adds an event to the end of the message queue for the control.
Print	Integer	Prints the control.
Resize	Integer	Changes the size of the control.
SetFocus	Integer	Sets the focus to the specified control.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its attributes.
Show	Integer	Makes the control visible.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

## CHAPTER 3

# Attributes, Events, and Functions for Application Objects

### About this chapter

An application is a collection of PowerBuilder windows that perform related activities, such as order entry or accounting activities. The *application object* is the entry point into the windows and controls that perform these activities.

When you or a user runs an application, an Open event occurs in the application object. The Open event triggers the script that initiates all the activity in the application.

This chapter lists the attributes, events, and functions in the definition of application objects.

### Contents

<b>Topic</b>	<b>Page</b>
Attributes	152
Events	153
Functions	154

# Application object

## Attributes

Attribute	Data type	Description
AppName	String	Specifies the name of the application object.
DDETimeOut	Integer	Specifies the number of seconds PowerBuilder acting as the DDE client waits before giving up when trying to communicate with a server via DDE (the default is 10 seconds).
DisplayName	String	User-readable name for your application. This name is displayed, for example, in OLE dialog boxes that show the application's name. If you don't specify a value, the value of AppName is used for DisplayName.
DWMMessageTitle	String	Specifies the title of the DataWindow message boxes that display during execution.
MicroHelpDefault	String	Specifies the default text of the MicroHelp object (the MicroHelp text that displays when you initiate a PowerBuilder session). The default is Ready.
ToolbarFrameTitle	String	Specifies the text that displays as the title for the FrameBar when it is floating.
ToolbarPopupMenuText	String	Specifies the text that displays on the popup menu for toolbars.
ToolbarSheetTitle		Specifies the text that displays as the title for the SheetBar when it is floating.

Attribute	Data type	Description
ToolbarText	Boolean	Specifies whether the text associated with the items in the toolbar displays: TRUE FALSE
ToolbarTips	Boolean	Specifies whether PowerTips display when text is not displayed on the buttons: TRUE FALSE
ToolbarUserControl	Boolean	Specifies whether users can use the toolbar popup menu to hide or show the toolbars, move toolbars, or show text: : TRUE FALSE

## Events

Event	Occurs
Close	When the user closes the application.
Idle	When the Idle function has been called in an application object script and the specified number of seconds have elapsed with no mouse or keyboard activity.
Open	When the user runs the application.
SystemError	When a serious execution time error occurs (such as trying to open a nonexistent window). If there is no script for this event, PowerBuilder displays a message box with the PowerBuilder error number and error message text.  <i>ℳ</i> For information about error messages, see the <i>User's Guide</i> .

## Functions

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
ClassName	String	Returns the class of the application object.
PostEvent	Boolean	Adds an event to the end of the message queue for the application object.
SetLibraryList	Integer	Sets the PBD library list in an executable.
TriggerEvent	Integer	Triggers a specified event in the application object and executes the script for the event.
TypeOf	Object	Returns the type of the application object.

## CHAPTER 4

# Attributes, Events, and Functions for MenuItems

### About this chapter

This chapter lists attributes, events, and functions in the MenuItem definition.

Menus are lists of MenuItems (usually commands or options) that a user can select in the currently active window. A MenuItem can display in either a menu bar or a dropdown or cascading menu.

When the user clicks a MenuItem, a Clicked event is triggered. If there is a dropdown or cascading menu under the clicked item, the script for the Clicked event for the item is executed, and then the menu displays. If there is no menu under the item, the script for the Clicked event for the item is executed.

### Contents

<b>Topic</b>	<b>Page</b>
Attributes	156
Events	158
Functions	158

# MenuItem

## Attributes

Attribute	Data type	Description
Checked	Boolean	Specifies whether the MenuItem is selected (a checkmark displays in the MenuItem): TRUE FALSE
Enabled	Boolean	Specifies whether the MenuItem is enabled (can be clicked): TRUE FALSE
Item[ ]	Menu	Specifies the items under a MenuItem.
MergeOption	MenuItemMergeOption (Enumerated)	Specifies how the menu item is affected when an OLE 2.0 object is activated.
MicroHelp	String	Specifies the text of the MicroHelp for the MenuItem.
Shortcut	Integer	Specifies the integer representation of the shortcut key for the MenuItem.
ParentWindow	Window	Specifies the window that owns the MenuItem.
ShiftToRight	Boolean	Specifies whether the MenuItem shifts down or to the right when other MenuItems are added in a descendant menu: TRUE FALSE



Attribute	Data type	Description
Tag	String	Specifies the tag value assigned to the MenuItem.
Text	String	Specifies the text in the MenuItem.
ToolBarItemDown	Boolean	Specifies whether the toolbar item remains down when it is clicked until you unclick it in a script.
ToolBarItemDownName	String	Specifies the name of the toolbar icon associated with the MenuItem when it is down.
ToolBarItemName	String	Specifies the name of a stock toolbar picture that you want to represent an item in the toolbar or a string containing the name of a bitmap file.
ToolBarItemOrder	Integer	Specifies the order of the item in the toolbar.
ToolBarItemSpace	Integer	Specifies the amount of empty space before the item in the toolbar.
ToolBarItemText	String	Specifies the text that displays in the toolbar item when the display text option is on for toolbars.
ToolBarItemVisible	Boolean	Specifies whether the toolbar item displays: TRUE FALSE
Visible	Boolean	Specifies whether the MenuItem is visible: TRUE FALSE

## Events

Event	Occurs
Clicked	When the MenuItem is clicked (selected or unselected).
Selected	When the user moves to the MenuItem using the arrow keys or the mouse.

## Functions

Function	Data type returned	Description
Check	Integer	Displays a checkmark next to the MenuItem and sets the Checked attribute.
ClassName	String	Returns the class of the MenuItem.
Disable	Integer	Disables (and grays) the MenuItem so that it cannot be selected and unsets the Enabled attribute.
Enable	Integer	Enables the MenuItem so that it can be selected and displays it normally (not grayed) and sets the Enabled attribute.
Hide	Integer	Makes the MenuItem invisible.
PopupMenu	Integer	Displays the MenuItem at the specified location.
PostEvent	Integer	Adds an event to the end of the message queue for the MenuItem.
Show	Integer	Makes the MenuItem visible.
TriggerEvent	Integer	Triggers a specified event in the MenuItem and executes the script for the event.
TypeOf	Object	Returns the type of the control.
Uncheck	Integer	Removes the checkmark next to the MenuItem and sets the Checked attribute to FALSE.

## CHAPTER 5

# Attributes and Functions for MDI Client

### About this chapter

This chapter describes the MDI client attributes window. PowerBuilder creates MDI\_1 when you select frame or frame with MicroHelp as the window style and uses it to identify the client area of the frame window internally. After you save the frame, MDI\_1 displays in the list of objects in the frame window.

MDI client is the area in which open sheets display in a standard MDI frame. In a standard MDI frame window, PowerBuilder sizes MDI\_1 so that it fills the space inside the frame. For example, if the frame has a menu bar and MicroHelp, MDI\_1 fills the space between the sides of the frame and the space below the menu bar and above the MicroHelp.

In a custom MDI frame window, you determine the size of the client area. For example, when a frame has buttons below the menu bar in the frame, you size the client area so it begins below the buttons.

#### Events

MDI Client cannot be selected and has no events.

### Contents

Topic	Page
Attributes	160
Functions	161

# MDI Client

## Attributes

Attribute	Data type	Description
BackColor	Long	Specifies the numerical value of the background color: -2 to 16,777,215.  <i>ℳ</i> For more information about color, see the <i>Function Reference</i> .
BringToTop	Boolean	Specifies whether PowerBuilder moves MDI_1 to the top of the front-to-back order: TRUE FALSE
Height	Integer	Specifies the height of MDI_1, in PowerBuilder units.
MicroHelpHeight	Integer	Specifies the height of the MicroHelp in MDI_1.  If the style of MDI frame window is MDI Frame (no MicroHelp), MicroHelpHeight is 0; otherwise, it is the height of the MicroHelp.
Tag	String	Specifies the tag value assigned to MDI_1.
Visible	Boolean	Specifies whether MDI_1 is visible: TRUE FALSE
Width	Integer	Specifies the width of MDI_1, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from the left edge of the MDI frame window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the MDI frame window), in PowerBuilder units.

## Functions

Function	Data type returned	Description
ClassName	String	Returns the name assigned to MDI_1.
Hide	Integer	Makes MDI_1 invisible.
Move	Integer	Moves MDI_1 to a specified location.
Resize	Integer	Changes the size of MDI_1.
SetRedraw	Integer	Controls automatic redrawing of MDI_1 after each change in its attributes. To reduce flicker when the user closes all the sheets, set SetRedraw to FALSE.
Show	Integer	Makes MDI_1 visible.
TypeOf	Object	Returns the type of a control or MDI_1.



## CHAPTER 6

# Attributes, Events, and Functions for System Objects

About this chapter      This chapter lists attributes, events, and functions for system objects in PowerBuilder.

Contents	Topic	Page
	Error	164
	Message	166
	OLEObject	168
	OLEStorage	170
	OLEStream	172
	Pipeline	174
	Transaction	176

## Error

The Error object is used to record execution-time errors. You can access the Error object in a script (typically for the SystemError event) to learn which error occurred and where it occurred.

You can also customize your own version of the Error object by defining a class user object inherited from the built-in Error object. The user object has two events: Constructor and Destructor.

☞ For more information about creating a custom Error object, see the chapter on user objects in the *User's Guide*.

☞ For more information about using the Error object in an application, see *Building Applications*.

## Attributes

Attribute	Data type	Description
Line	Integer	Identifies the line in the script at which the error occurred.
Number	Integer	Identifies the PowerBuilder error.
Object	String	Contains the name of the object in which the error occurred. If the error occurred in a window or menu, Object will be the same as WindowMenu.
ObjectEvent	String	Contains the event in which the error occurred.
Text	String	Contains the text of the error message.
WindowMenu	String	Contains the name of the window or menu in which the error occurred.



## Events

<b>Event</b>	<b>Occurs</b>
Constructor	When the user object is created.
Destructor	When the user object is destroyed.

## Functions

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
ClassName	String	Returns the name assigned to the user object.
PostEvent	Boolean	Adds an event to the end of the message queue of the user object.
TriggerEvent	Integer	Sends an event to the user object and executes the script associated with the event.
TypeOf	Object	Returns the type of the user object.

## Message

The Message object is used to process events that are not PowerBuilder-defined events, to communicate parameters between windows when you open and close them, and if optional parameters are used in TriggerEvent or PostEvent.

You can also customize your own version of the Message object by defining a class user object inherited from the built-in Message object. The user object has two events: Constructor and Destructor.

*ℳ* For more information about creating a custom Message object, see the chapter on user objects in the *User's Guide*.

## Attributes

The first four attributes of the Message object correspond to the first four attributes of the Microsoft Windows message structure:

Attribute	Data type	Description
Handle	Integer	The handle of the window or control.
Number	Integer	The number that identifies the event (this number comes from Windows).
WordParm	UnsignedInt	The word parameter for the event (this parameter comes from Windows). The parameter's value and meaning are determined by the event.
LongParm	Long	The long parameter for the event (this number comes from Windows). The parameter's value and meaning are determined by the event.
DoubleParm	Double	A numeric or numeric variable.
StringParm	String	A string or string variable.
PowerObjectParm	PowerObject	Any PowerBuilder object type including structures.

Attribute	Data type	Description
Processed	Boolean	A boolean value set in the script for the user-defined event or the Other event:  TRUE—The script processed the event (do not call the default window Proc (DefWindowProc) after the event has been processed).  FALSE—(Default) Call DefWindowProc after the event has been processed.
ReturnValue	Long	The value you want returned to Windows when Message.Processed is TRUE. When Message.Processed is FALSE, this attribute is ignored.

## Events

Event	Occurs
Constructor	When the user object is created.
Destructor	When the user object is destroyed.

## Functions

Function	Data type returned	Description
ClassName	String	Returns the name assigned to the user object.
PostEvent	Boolean	Adds an event to the end of the message queue of the user object.
TriggerEvent	Integer	Sends an event to the user object and executes the script associated with the event.
TypeOf	Object	Returns the type of the user object.

## OLEObject

The OLEObject object acts as a proxy for a remote OLE object.

You can also customize your own version of the OLEObject object by defining a class user object inherited from the built-in OLEObject object. The user object has two events: Constructor and Destructor.

✎ For more information about creating a custom OLEObject object, see the chapter on user objects in the *User's Guide*.

✎ For more information about using the OLEObject object in an application, see *Building Applications*.

### **OLEObject is a dynamic object**

In order to support OLE 2.0, OLEObject is a dynamic object. The PowerBuilder compiler will accept attribute names and function names and parameter lists that are not already defined for the object. If the attributes or functions do not exist during execution, you will get an execution-time error.

## Events

<b>Event</b>	<b>Occurs</b>
Constructor	When the user object is created.
Destructor	When the user object is destroyed.

## Functions

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
ClassName	String	Returns the name assigned to the user object.
ConnectToNewObject	Integer	Creates a new instance of the class and connects to it.

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
ConnectToObject	Integer	Opens a specified file and connects to the corresponding server application.
DisconnectObject	Integer	Releases all objects previously connected
PostEvent	Boolean	Adds an event to the end of the message queue of the user object.
TriggerEvent	Integer	Sends an event to the user object and executes the script associated with the event.
TypeOf	Object	Returns the type of the user object.

# OLEStorage

The OLEStorage object acts as a proxy for an open OLE storage.

You can also customize your own version of the OLEStorage object by defining a class user object inherited from the built-in OLEStorage object. The user object has two events: Constructor and Destructor.

☞ For more information about creating a custom OLEStorage object, see the chapter on user objects in the *User's Guide*.

☞ For more information about using the OLEStorage object in an application, see *Building Applications*.

## Events

Event	Occurs
Constructor	When the user object is created.
Destructor	When the user object is destroyed.

## Functions

Function	Data type returned	Description
ClassName	String	Returns the name assigned to the user object.
Clear	Integer	Releases any storage previously opened.
Close	Integer	Saves the storage and any controls and streams open on the storage, commits the changes, then releases the storage. (Same as calling Save, then Clear.)
MemberDelete	Integer	Deletes the specified member.
MemberExists	Integer	Specifies whether specified member exists.

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
MemberRename	Integer	Renames specified member.
Open	Integer	Opens the storage from a file or opens a substorage within the specified storage object.
PostEvent	Boolean	Adds an event to the end of the message queue of the user object.
Save	Integer	Saves the storage and any controls open on that storage and commits the changes.
SaveAs	Integer	Copies the storage and any controls open on that new storage to a new file or substorage, commits the changes, then releases original storage.
TriggerEvent	Integer	Sends an event to the user object and executes the script associated with the event.
TypeOf	Object	Returns the type of the user object.

## OLEStream

The OLEStream object acts as a proxy for an OLE stream.

You can also customize your own version of the OLEStream object by defining a class user object inherited from the built-in OLEStream object. The user object has two events: Constructor and Destructor.

🌀 For more information about creating a custom OLEStream object, see the chapter on user objects in the *User's Guide*.

🌀 For more information about using the OLEStream object in an application, see *Building Applications*.

## Attributes

Attribute	Data type	Description
Storage	OMStorage	The storage containing the stream (read-only).

## Events

Event	Occurs
Constructor	When the user object is created.
Destructor	When the user object is destroyed.

## Functions

Function	Data type returned	Description
ClassName	String	Returns the name assigned to the user object.
Close	Integer	Releases any stream previously opened.



<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
Length	Integer	Obtains the length of the open stream.
Open	Integer	Opens specified stream from the storage.
PostEvent	Boolean	Adds an event to the end of the message queue of the user object.
Read	Integer	Reads from the stream.
Seek	Integer	Moves within the stream.
TriggerEvent	Integer	Sends an event to the user object and executes the script associated with the event.
TypeOf	Object	Returns the type of the user object.
Write	Long	Writes to the stream.

## Pipeline

A pipeline system object is used to manage a data pipeline during execution. You use a pipeline system object by defining a class user object inherited from the built-in pipeline system object in the User Object painter.

☞ For more information about piping data, see *Building Applications*.

## Attributes

Attribute	Data type	Description
RowsInError	Long	The number of rows the pipeline found in error (for example, rows containing a duplicate key).
RowsRead	Long	The number of rows read by the pipeline.
RowsWritten	Long	The number of rows written by the pipeline.
DataObject	String	The name of the pipeline object (the object created in the Data Pipeline painter).
Syntax	String	The syntax used to create the pipeline object (the object created in the Data Pipeline painter).

## Events

Event	Occurs
Constructor	When the user object is created.
Destructor	When the user object is destroyed.
PipeEnd	When Start or Repair is completed.
PipeMeter	After each block of rows is read or written. The Commit factor specified for the pipeline determines the size of each block.
PipeStart	When a Start or Repair is started.

## Functions

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
Cancel	Integer	Stops execution of a pipeline.
ClassName	String	Returns the name assigned to the user object.
PostEvent	Boolean	Adds an event to the end of the message queue of the user object.
Repair	Integer	Updates the target database with corrections that have been made in the pipeline user object's Error DataWindow.
Start	Integer	Executes a pipeline.
TriggerEvent	Integer	Sends an event to the user object and executes the script associated with the event.
TypeOf	Object	Returns the type of the user object.

## Transaction

The Transaction object specifies the parameters that PowerBuilder uses to connect to a database.

You can also customize your own version of the Transaction object by defining a class user object inherited from the built-in Transaction object. The user object has two events: Constructor and Destructor.

☞ For more information about creating a custom Transaction object, see the chapter on user objects in the *User's Guide*.

☞ For more information about using the Transaction object in an application, see *Building Applications*.

## Attributes

Attribute	Data type	Description
AutoCommit	Boolean	The automatic commit indicator (SQL Server only):  TRUE Commit automatically after every database activity  FALSE Do not commit automatically after every database activity
SQLCode	Long	The success or failure code of the most recent operation:  0 Success  100 Not Found  -1 Error (use SQLDBCode or SQLErrText to obtain the details)
SQLDBCode	Long	The database vendor's error code.
SQLNRows	Long	The number of rows affected (the database vendor supplies this number, so the meaning may not be the same in every DBMS).

Attribute	Data type	Description
Database	String	The name of the database with which you are connecting.
DBMS	String	PowerBuilder vendor identifier.
DBParm	String	DBMS-specific parameters.
DBPass	String	The password that will be used to connect to the database.
Lock	String	The isolation level.
LogID	String	The name or ID of the user who will log on to the server.
LogPass	String	The password that will be used to log on to the server.
ServerName	String	The name of the server on which the database resides.
SQLErrMsgText	String	The database vendor's error message.
SQLReturnData	String	DBMS-specific information.
UserID	String	The name or ID of the user who will connect to the database.

## Events

Event	Occurs
Constructor	When the user object is created.
Destructor	When the user object is destroyed.

## Functions

Function	Data type returned	Description
ClassName	String	Returns the name assigned to the user object.
DBHandle	Long	Returns the handle for your DBMS.

<b>Function</b>	<b>Data type returned</b>	<b>Description</b>
PostEvent	Boolean	Adds an event to the end of the message queue of the user object.
SyntaxFromSQL	String	Generates DataWindow source code based on a SQL SELECT statement.
TriggerEvent	Integer	Sends an event to the user object and executes the script associated with the event.
TypeOf	Object	Returns the type of the user object.

## CHAPTER 7

# Attributes for Environment Object

About this chapter

This chapter lists the attributes for the built-in Environment object.

Contents

<b>Topic</b>	<b>Page</b>
Attributes	180

# Environment

The Environment object is used to hold information about the computing platform the PowerBuilder application is running on. You populate the Environment object using the GetEnvironment function.

☞ For more information about GetEnvironment, see the *Function Reference*.

## Attributes

Attribute	Data type	Description
CPUType	CPUTypes	The CPU.
OSFixesRevision	Integer	The maintenance version of the operating system.
OSMajorRevision	Integer	The major version of the operating system.
OSMinorRevision	Integer	The point release of the operating system.
PBFixesRevision	Integer	The maintenance version of PowerBuilder.
PBMajorRevision	Integer	The major version of PowerBuilder.
PBMinorRevision	Integer	The point release of PowerBuilder.
NumberOfColors	Long	Number of colors on the screen.
ScreenHeight	Long	Height of the screen in pixels.
ScreenWidth	Long	Width of the screen in pixels.
OSType	OSTypes	Operating system or environment.
PBType	PBTypes	Version of PowerBuilder product (for example, Enterprise or Desktop).



**Example**

For example, if you are running PowerBuilder Enterprise 4.0 under Windows 3.1 on an Intel 486 computer with Super VGA, here is what your Environment object would be:

<b>Attribute</b>	<b>Value</b>
CPUType	i486!
OSFixesRevision	0
OSMajorRevision	3
OSMinorRevision	10 (that is, Windows 3.10)
PBFixesRevision	0
PBMajorRevision	4
PBMinorRevision	0
NumberOfColors	16
ScreenHeight	800
ScreenWidth	600
OSType	Windows!
PBType	Enterprise!